Journal of Information Security, 2015, 6, 38-50
Published Online January 2015 in SciRes. http://www.scirp.org/journal/jis
http://dx.doi.org/10.4236/jis.2015.61005

Symmetric-Key Based Homomorphic Primitives for End-to-End Secure Data Aggregation in Wireless Sensor Networks

Keyur Parmar, Devesh C. Jinwala

Department of Computer Engineering, S. V. National Institute of Technology, Surat, India
Email: keyur.mtech@gmail.com, dcjinwala@gmail.com

Received 25 September 2014; revised 24 October 2014; accepted 11 December 2014

Abstract

In wireless sensor networks, secure data aggregation protocols target the two major objectives, namely, security and en route aggregation. Although en route aggregation of reverse multi-cast traffic improves energy efficiency, it becomes a hindrance to end-to-end security. Concealed data aggregation protocols aim to preserve the end-to-end privacy of sensor readings while performing en route aggregation. However, the use of inherently malleable privacy homomorphism makes these protocols vulnerable to active attackers. In this paper, we propose an integrity and privacy preserving end-to-end secure data aggregation protocol. We use symmetric key-based homomorphic primitives to provide end-to-end privacy and end-to-end integrity of reverse multicast traffic. As sensor network has a non-replenishable energy supply, the use of symmetric key based homomorphic primitives improves the energy efficiency and increase the sensor network's lifetime. We comparatively evaluate the performance of the proposed protocol to show its efficacy and efficiency in resource-constrained environments.

Keywords

Wireless Sensor Network, Security, Concealed Data Aggregation, In-Network Processing, Secure Data Aggregation, Homomorphic Encryption, Homomorphic MAC

1. Introduction

Recent advancement in Micro-Electro-Mechanical Systems (MEMS) technology has facilitated the development of tiny and cost-effective sensor devices [1] [2]. These tiny sensor devices collaborate to form a network, referred as wireless sensor network (WSN) [3]. These sensor devices often worked as actuators, where each of

How to cite this paper: Parmar, K. and Jinwala, D.C. (2015) Symmetric-Key Based Homomorphic Primitives for End-to-End Secure Data Aggregation in Wireless Sensor Networks. *Journal of Information Security*, **6**, 38-50.
http://dx.doi.org/10.4236/jis.2015.61005

them can have escalating capabilities to perform sensing, processing, and transmission. However, sensor devices have very limited resources like battery power, processor, energy, bandwidth, etc. Among these resources, energy is the most crucial resource due to its direct impact on the sensor nodes' lifetime. Although communication and computation both consume energy, the radio frequency (RF) operations consume far more energy than the CPU instructions [4]. As shown in [4], transmission of a single bit requires the same amount of energy as the execution of 1000 CPU instructions. Hence, protocols in sensor networks aim to reduce the communication traffic. In-network processing, also known as data aggregation, helps to reduce redundant communication traffic [5].

As sensor nodes are deployed in unattended and hostile environments, security becomes an important design parameter. Moreover, traditional security mechanisms cannot be adapted directly due to the unique challenges of sensor networks. Resource-constrained nature of sensor nodes, lack of physical protection, and hostile deployments are among several other characteristics that make security in WSNs a formidable challenge [6]-[8]. Secure data aggregation protocols aim to meet these two critical objectives together, namely, security and data aggregation. However, traditional end-to-end security cannot be realized in data-centric networks like WSNs, where the data are supposed to be altered at intermediate nodes before reaching the base station. As data are aggregated en route, an intermediate node must require a secret key to decrypt encrypted sensor readings before processing. Hop-by-hop secure data aggregation assumes trustworthy intermediate nodes that possess keys to decrypt the encrypted data and process them before forwarding it to the next hop [9]. Malicious adversaries target such intermediate nodes to obtain a large amount of gathered information. Hence, the need to protect the privacy of sensor readings at intermediate nodes arises.

Girao *et al*. [10] [11] proposed "concealed data aggregation" that achieves end-to-end privacy of sensor readings while performing en route aggregation. They used "privacy homomorphism", originally proposed by Rivest *et al*. [12], to perform encrypted data processing. Privacy homomorphism processes encrypted sensor readings without decrypting them at intermediate nodes. However, the feature that protects privacy of sensor readings becomes a hindrance achieving integrity protection. Traditional security protocols often consider privacy homomorphism as weakness to achieve the highest level of security achieved against adaptive chosen-ciphertext attacks (CCA2). The algorithms that support privacy homomorphism are inherently malleable. Hence, the highest level of security that they can achieve, is against non-adaptive chosen ciphertext attacks (CCA1). As sensor network performs en route aggregation of reverse multicast traffic, the malleability property has a catastrophic effect on the correctness of gathered sensor readings. Any compromised sensor nodes can inject fake data packets to falsify the genuinely aggregated data. In addition, if public-key cryptography is employed then with the help of public parameters, the malicious adversary can inject fake data packets without compromising a single node. Hence, the need to ensure the correctness of gathered sensor readings becomes a formidable challenge. Traditional hop-by-hop authentication protocols cannot provide end-to-end integrity protection in data-centric networks. In addition, the packet size considered by TinyOS [13], an operating system for low power wireless devices, is only 36 bytes, out of which only 29 bytes are reserved for payload information. In traditional networks, message authentication codes usually require 8 bytes to provide a reasonable level of security. However in sensor networks, the use of 8 bytes protecting only 2 or 3 bytes of sensor readings, is highly undesirable. In addition, the data are aggregated en route to reduce the communication traffic, but their MAC tags cannot be aggregated. Hence the need for end-to-end privacy, end-to-end authentication and en route aggregation of reverse multicast traffic in sensor networks arises.

In this paper, we provide end-to-end privacy, end-to-end integrity and en route aggregation of sensor readings using only cost-effective symmetric key based mechanisms. We use a homomorphic encryption algorithm to provide the en route encrypted data processing. In addition, we use a homomorphic MAC algorithm to achieve the end-to-end integrity protection. As the proposed protocol uses a symmetric key based encryption and authentication mechanisms, it achieves the significant energy reduction that increases the sensor networks' lifetime. As per our knowledge, the proposed protocol is the first to achieve the above-mentioned objectives using symmetric key based homomorphic primitives.

The rest of the paper is organized as follows. Section 2 discusses the relevant literature. In Section 3, we briefly discuss the preliminaries required by the proposed protocol. We present the proposed protocol in Section 4. The overhead analysis is presented in Section 5, followed by the security analysis in Section 6. Section 7 concludes the paper by emphasizing our contributions.

2. Related Work

Hu *et al*. [14] and Przydatek *et al*. [15] explored the ways to protect the aggregated data in WSNs. Their solutions ensure the protection of sensor readings against outsider adversaries. Although numerous authors claim to provide such hop-by-hop secure data aggregation [9] [16], they all assume that intermediate nodes are trustworthy. As sensor nodes are deployed in hostile environments, such assumptions may not suit the need of a large number of applications. Therefore, Girao *et al*. [10] [11] proposed a concealed data aggregation protocol that do not consider trustworthy intermediate nodes. They used Domingo Ferrer's symmetric key based encryption algorithm [17] to perform encrypted data processing at intermediate nodes. In 2005, Castelluccia *et al*. [18] [19] proposed a symmetric key based homomorphic cryptosystem based on one-time pad. In their cryptosystem, each node is equipped with a unique secret key shared with the base station. Hence, a single compromised node cannot make the whole network vulnerable, as in the case of other symmetric key based techniques [17] [20]. Asymmetric key based homomorphic cryptosystems [21] [22] including those based on the elliptic curves [22] [23], are expensive and require more resources compared to their symmetric counterparts [17] [18] [20].

Although concealed data aggregation protects privacy of sensor readings at intermediate nodes, the use of privacy homomorphism [12] makes them inherently malleable [24]. Privacy homomorphism is often being considered as an undesirable property [25]. The algorithms that support privacy homomorphism cannot be secure against adaptive chosen-ciphertext attack (CCA 2) [26]. Encrypted data processing allows intermediate nodes to aggregate the encrypted sensor readings using publicly available information. Hence, encrypted data processing allows not only genuine aggregator nodes, but it also allows malicious adversaries to process the encrypted data without the need for any secret information. Therefore, the need for an authentication mechanism that ensure the integrity of aggregated data becomes imperative.

Although hop-by-hop integrity verification can be achieved through existing authentication mechanisms, the same mechanisms cannot be used to provide end-to-end integrity verification [27]. The en route aggregation of sensor readings and encrypted data processing make end-to-end integrity verification a formidable challenge. Agrawal *et al*. [28] proposed a homomorphic MAC that provides integrity verification in data-centric networks. Homomorphic MAC aggregates message authentication codes (MACs) to reduce the communication traffic. In addition, homomorphic MAC verifies the integrity of aggregated data. Although asymmetric key based homomorphic primitives like asymmetric key based homomorphic encryption [24] and homomorphic digital signature [29] [30] exist in literature, we consider only symmetric key based homomorphic primitives due to their relatively fewer resource requirements.

3. Preliminaries

In this section, we briefly discuss privacy homomorphism [12]. We describe Castelluccia *et al*.'s [18] [19] symmetric key based cryptosystem that supports homomorphic encryption. In addition, we discuss a symmetric key based homomorphic MAC algorithm [28] used to perform MAC aggregation such that the resultant MAC verifies the aggregated and encrypted data.

3.1. Privacy Homomorphism

Rivest *et al*. [12] presented a way to perform the computation over encryption data. Formally, a privacy homomorphism is defined as a mapping function f between a set of plaintexts $\mathcal{P}$ and a set of ciphertexts $\mathcal{C}$ such that the group operation is preserved.

$$f\left(m_1 \oplus m_2\right) = f\left(m_1\right) \otimes f\left(m_2\right), \quad \forall m_1, m_2 \in \mathcal{P} \quad \text{and} \quad \forall f\left(m_1\right), f\left(m_2\right) \in \mathcal{C} \tag{1}$$

Here, the operators used to perform the group operations, need not be different each time. With privacy homomorphism, encryption of aggregated data or aggregation of encrypted data both yield the same result. Cryptosystems can support additive and multiplicative privacy homomorphisms [24]. However, applications in sensor networks require an additive privacy homomorphism only [31]. An additive privacy homomorphism can support functions like minimum, maximum, average, variance, movement detection, etc. In the next section, we briefly discuss homomorphic primitives used for the encryption and message authentication (integrity).

3.2. Homomorphic Encryption

Cryptosystems that support the additive privacy homomorphism can perform an addition operation over the encrypted data without decrypted them. They are basically categorized as either a symmetric-key based homomorphic encryption [17] [18] or an asymmetric-key based homomorphic encryption [23] [32] [33]. In this section, we discuss Castelluccia *et al.*'s [18] [19] symmetric key based encryption algorithm (CMT Cryptosystem). Castelluccia *et al*. replaced X-OR operation typically found in stream cipher based cryptosystems with modular addition operation. In their cryptosystem, each sensor node is equipped with a unique secret key shared with the base station. Hence, a compromised node cannot decrypt other sensor nodes' encrypted data, as in the case of other symmetric key based cryptosystems [11] [20] [31]. Moreover, intermediate nodes can aggregate encrypted data without decrypting them and without having any secret information about the keys used during the encryption.

CMT Cryptosystem

Key Generation $\mathcal{K}$:

1. Randomly choose $K \in \{0,1\}^{\lambda}$ as a decryption key of the base station.
2. For each node $i \in [1,n]$, compute an encryption key $k_i = f_K(i)$, Here, f is a pseudo-random function (PRF).

Encryption $\mathcal{E}$:

1. Given an encryption key k_i and a nonce r. A nonce can be forwarded by the base station in a query or it can be automatically generated based on a predefined criteria.
2. Represent a plaintext m_i, as an integer in the range $[0, M-1]$, where M is the modulus.
3. Compute the ciphertext, $c_i = \mathcal{E}_{k_i}(m_i) = m_i + h(f_{k_i}(r)) \bmod M$.
4. Set $\mathrm{ID}_i = i$.
5. Forward, (ID_i, c_i) as an ID-ciphertext pair.

Decryption $\mathcal{D}$:

1. Given a ciphertext $c = \sum_{i \in \mathrm{ID}} c_i$, nodes' identity information ID, and a nonce r, generate $k_i = f_K(i)$, $\forall i \in \mathrm{ID}$.
2. Decrypt the ciphertext, $\mathcal{D}_K(c) = c - \sum_{i \in \mathrm{ID}} h(f_{k_i}(r)) \bmod M = \sum_{i \in \mathrm{ID}} m_i \bmod M$

Ciphertexts Aggregation $\mathcal{A}$:

1. Given $c_i = \mathcal{E}_{k_i}(m_i)$ and $c_j = \mathcal{E}_{k_j}(m_j)$
2. Compute an aggregated ciphertext, $c = c_i + c_j \bmod M = \mathcal{E}_k(m_i + m_j) + h(f_k(r))$ where $k = f_K(i+j)$.
3. Set $\mathrm{ID} = i \cup j$.

3.3. Homomorphic Message Authentication Code

In data-centric networks, data are supposed to be altered en route. Hence, traditional authentication mechanisms cannot be used to provide end-to-end integrity support. Moreover, privacy homomorphism makes sensor readings more vulnerable to active attackers. Any intermediate node can aggregate ciphertexts without performing decryption or without having any secret information. Although there exist homomorphic digital signatures [29] [30] that support integrity verification in data-centric networks, they remain infeasible due to the excessive computation and communication cost required to perform per packet integrity verification. Homomorphic MAC is a symmetric counterpart of a homomorphic digital signature [29] [30] used for integrity verification.

Agrawal *et al*. [28] proposed a homomorphic MAC to verify the integrity of encrypted and aggregated data in data-centric networks. Homomorphic MAC can be formally defined using three probabilistic and polynomial-time algorithms, namely, Sign, Verify and Combine. Given a vector space V spanned by the vectors $v_1, v_2, \cdots, v_m \in \mathbb{F}_q^{n+m}$. The Sign algorithm computes a MAC tag for one basis vector at a time. The Combine algorithm is used to perform the aggregation of MAC tags while the Verify algorithm verifies the vector-MAC tag pairs. Here, we briefly discuss a homomorphic MAC algorithm. However, its detailed description with relevant security proofs can be found in [28].

Homomorphic MAC

• Given a pseudo random generator $G: \mathcal{K}_G \to \mathbb{F}_q^{n+m}$ and a pseudo random function $F: \mathcal{K}_F \times (\mathcal{I} \times [m]) \to \mathbb{F}_q$.

Let $k_1 \in \mathcal{K}_G$ and $k_2 \in \mathcal{K}$ are the keys used for the MAC construction.

Sign: Given i^{th} basis vector $v \in \mathbb{F}_q^{n+m}$ and a key pair $k = (k_1, k_2)$ do:

1. $u \leftarrow G(k_1) \in \mathbb{F}_q^{n+m}$
2. $b \leftarrow F(k_2, (id, i)) \in \mathbb{F}_q$
3. $\mathcal{T} \leftarrow (u \cdot v) + b \in \mathbb{F}_q$

Here, $\mathcal{T} \in \mathbb{F}_q$ is a MAC tag.

Combine: Given (v_1, t_1, α_1), $\cdots$, (v_m, t_m, α_m), compute,

$$\mathcal{T} \leftarrow \sum_{j=1}^{m} \alpha_j \mathcal{T}_j \in \mathbb{F}_q$$

Verify: Given a secret key $k = (k_1, k_2)$ and $y = (y_1, \cdots, y_{n+m}) \in \mathbb{F}_q^{n+m}$, verify a tag $\mathcal{T}$ as follows.

- $u \leftarrow G(k_1) \in \mathbb{F}_q^{n+m}$ and $a \leftarrow (u \cdot y) \in \mathbb{F}_q$
- $b \leftarrow \sum_{i=1}^{m} \left[y_{n+i} \cdot F(k_2, (id, i)) \right] \in \mathbb{F}_q$
- If $a + b = \mathcal{T}$ then output 1; otherwise output 0

4. The Proposed Protocol

In this section, we present our proposed protocol to provide integrity assured concealed data aggregation for reverse multicast traffic in wireless sensor networks. The proposed protocol is the first that achieves the formidable objectives like en route aggregation, privacy at intermediate nodes and integrity assurance of aggregated, as well as raw sensor readings, using the symmetric-key based homomorphic primitives.

We use a symmetric key based encryption algorithm to provide the privacy of sensor readings at intermediate nodes. The encryption algorithm helps to achieve an end-to-end privacy of reverse multicast traffic in wireless sensor networks. In addition, we used a symmetric key based message authentication code (MAC) that verifies the integrity of received packets when there exist malicious outsider adversaries. Finally, we use a homomorphic MAC to provide an end-to-end integrity verification ensuring the protection against malicious intermediate adversaries. The proposed protocol not only reduces the bandwidth consumption through aggregation, it reduces the bandwidth consumption when there exist malicious adversaries. For the clarity, we list the notations used by the proposed protocol in **Table 1**.

4.1. Integrity Assured Concealed Data Aggregation Protocol

• At Leaf Nodes:

- Each node (leaf node and intermediate node) i shares a unique symmetric-key k_i with the base station. In addition, each node i shares a pairwise symmetric key k_{i-j} with a neighboring node j.
- The base station shares a unique symmetric-key pair $k' = \{k_1, k_2\}$ with the leaf nodes only.
- Node i encrypts its sensor reading m_i using a symmetric key k_i shared with the base station.

Table 1. Table of notations.

Symbol	Description
i	Sensor node ID.
m_i	Sensor reading of a node i.
k_i	A symmetric key between a node i and the base station.
k_{i-j}	A symmetric key between a node i and a node j.
$k' = \{k_1, k_2\}$	Symmetric keys shared between the base station and leaf nodes.
c_i	Ciphertext generated by a node i.
r	A nonce transmitted by the base station in a query.
h	A provably secure pseudo random function.
M	A modulus, predefined by the base station.
$\mathcal{T}_i'$	Homomorphic MAC tag generated by a node i using a key pair k_i'.
$\mathcal{T}_i$	MAC tag generated by a node i using a key k_{i-j} shared with its neighboring node j.
x	A number that represents the child nodes of an intermediate node.
HDR	Header information that uniquely identifies the node.

$$c_i = \mathcal{E}_{k_i}(m_i) = m_i + h\left(f_{k_i}(r)\right) \bmod M$$

Here, h represent a provably secure pseudo-random function (PRF) and r is a nonce included in a query forwarded by the base station.

- Node i generates a homomorphic MAC tag $\mathcal{T}_i' = H - \text{MAC}_{k'}(m_i)$ using a key pair $k' = \{k_1, k_2\}$ shared with only the base station.
- Node i generates a MAC tag, $\mathcal{T}_i = \text{MAC}_{k_{i-j}}(c_i \| \mathcal{T}_i')$ using a pairwise symmetric key k_{i-j} shared with its neighboring node j.
- Node i forwards a packet, $(\text{HDR}, c_i, \mathcal{T}_i, \mathcal{T}_i')$, towards the base station. Here, HDR contains the identity information for a node i.

• At Intermediate Nodes:

- Intermediate node j receives the packet, $(\text{HDR}, c_i, \mathcal{T}_i, \mathcal{T}_i')$, forwarded by a node i. In the same way, it receives packets forwarded by its other child nodes.
- Intermediate node j generates a MAC tag and compares it with the received MAC tag and accepts the packet if they are same. If they are different, it drops the packet.

$$\mathcal{T}_i \stackrel{?}{=} \text{MAC}_{k_{i-j}}(c_i \| \mathcal{T}_i')$$

- Intermediate node j aggregates a ciphertexts forwarded by its child nodes.

$$c_j = \sum_{i=1}^{x} c_i \quad \text{Here, } i \in [1 \cdots x] \text{ represents the child nodes of node } j$$

- Intermediate node j combines the homomorphic MAC tags $\mathcal{T}'$ coming from its child nodes.

$$\mathcal{T}_j' = \sum_{i=1}^{x} \mathcal{T}_i'$$

- Intermediate node j generates a MAC tag, $\mathcal{T}_j = \text{MAC}_{k_{i-j}}(c_j)$ using a pairwise symmetric key k_{j-l} shared with its neighboring node l.
- Intermediate node j concatenates the header information HDR.
- Node j forwards a packet, $(\text{HDR}, c_j, \mathcal{T}_j, \mathcal{T}_j')$, towards the base station.

• At the Base Station:

- The base station verifies the received packet, $(\text{HDR}, c_l, \mathcal{T}_l, \mathcal{T}_l')$ using a pairwise symmetric key $k_{l\text{-BS}}$ shared with its neighboring node l.

$$\mathcal{T}_l \overset{?}{=} \text{MAC}_{k_{l\text{-BS}}}(c_l \| \mathcal{T}_l')$$

The verification ensures that the ciphertext and its corresponding homomorphic MAC tag are not modified by the outsider adversaries.

- The base station decrypts the aggregated ciphertext, $m = D_k(c_l)$.
- The base station generates a H-MAC tag using the key pair $k' = \{k_1, k_2\}$.

$$\mathcal{T}' = H - \text{MAC}_{k'}(m)$$

- The base station compares the newly generated homomorphic MAC tag with the received homomorphic MAC tag. If they are same, it accepts the aggregated sensor reading m.

4.2. Example

As shown in **Figure 1**, there are two types of nodes in the network: 1) Leaf nodes and 2) Aggregator nodes. Each leaf node encrypts the sensor reading with a symmetric-key shared with the base station. In addition, each leaf node generates a homomorphic MAC using a symmetric-key shared with the base station. Finally, it generates a MAC over the ciphertext-Homomorphic MAC pair using a key shared with its neighboring node. The aggregator nodes verify the MACs and if they are correct, they aggregate the ciphertext as well as homomorphic MACs. Moreover, aggregator nodes generate a MAC over the aggregated ciphertext and H-MAC pair using a pair-wise secret key shared with its neighboring node. At the base station the MACs and homomorphic MACs are verified and if they are correct, the aggregated sensor reading is accepted.

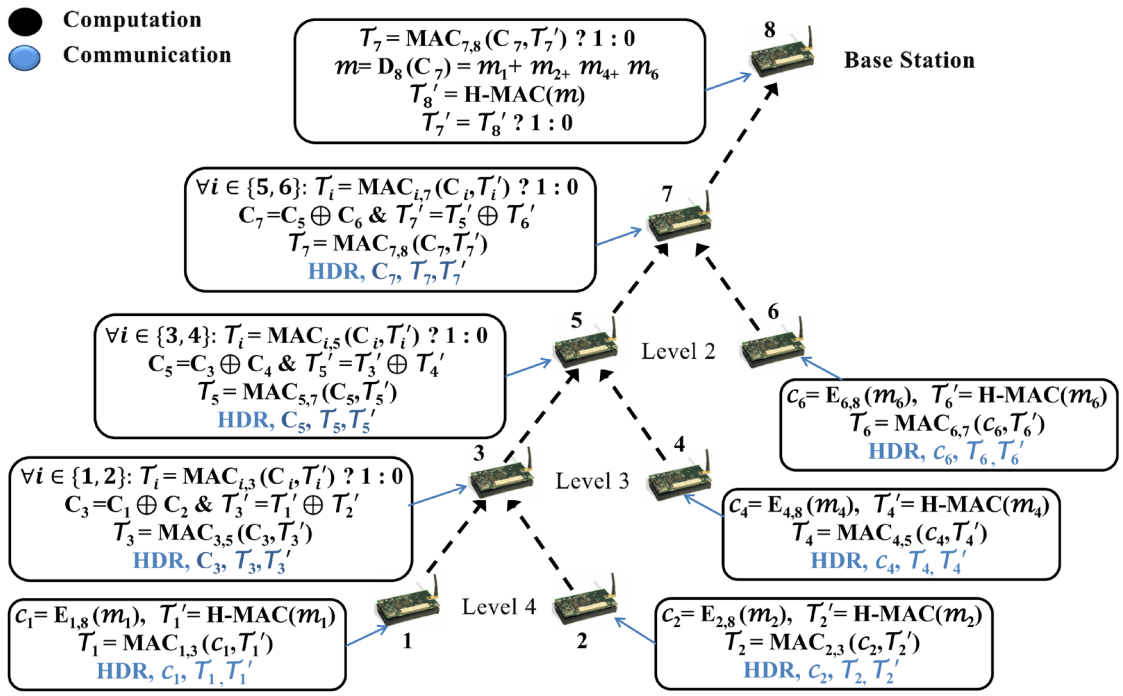

Figure 1. Integrity assured concealed data aggregation.

5. Overhead Analysis

In sensor networks, communication requires far more energy than the computation. As shown in [4], transmitting a single bit over a meter range requires the same amount of energy as to execute 1000 CPU instructions. Hence, one of the major goals of WSNs is to reduce the communication traffic. In this section, we compare the bandwidth consumption of the proposed protocol with the same scenarios as discussed by Castelluccia *et al*. [18] [19] to ease the comparison. In the next section, we present a network model used to measure the energy consumption.

5.1. Network Model

To measure the bandwidth consumption, we consider a network model as described in Castelluccia *et al*. [18] [19]. We consider a base station and multitude of sensor nodes spanned across the multiple levels of hierarchy. For ease of comparison, we consider the same 3-ary tree topology as described in [18] [19], and as shown in **Figure 2**. However, it can be any *n*-ary tree topology. In addition, the assumption of a tree topology can be seamlessly replaced by a cluster-based topology or a hybrid topology. The measured sensor readings can be represented by a $\log_2(n)$ bit integer that can represent *n* different sensor readings. Generally, temperature sensors require only 7 bits to represent 128 different temperature readings. For other sensors, we can increase the number of bits required for representation, or we can group sensor readings to reduce the total number of bits required for representation.

To measure the bandwidth consumption, we used the same packet format as used by the TinyOS [13], an operating system for resource-constrained sensor networks. In TinyOS, the packet size is of 36 bytes where 7 bytes are reserved for packet header and remaining 29 bytes are for data payload. If payload data is more than 29 bytes, it is fragmented into 29 byte blocks, and remaining payload is shifted to other packets. As shown in TinyECC [34], due to the digital signature, the packet size needs to be extended up to 102 bytes. Hence, we use the 4 or 8 byte MAC as suggested by Karlof *et al*. [35] for resource-constrained devices.

5.2. Communication Overhead

We compare the performance of the proposed protocol for End-to-end Encryption and Authentication, with four different scenarios. 1) No Aggregation at intermediate nodes (NA); 2) Concatenation of Payload (CP); 3) Hop-by-hop Encryption (HE); 4) End-to-end Encryption (EE).

As shown in **Figure 3**, the comparison with above mentioned scenarios shows that the bandwidth consumption is high when the data are not aggregated en route. Moreover, no aggregation and packet concatenation based approaches consume much higher bandwidth than the scenarios where en route aggregation are performed. The bandwidth consumption of aggregation based approaches is negligible compared to the NA and CP scenarios.

As shown in **Figure 4**, when we exclusively compare the aggregation based scenarios, the proposed protocol consumes few more bytes compared to hop-by-hop encryption and end-to-end encryption approach. In hop-by-hop encryption, the number of bits transmitted by leaf nodes can be calculated as $\mathrm{HDR} + \log_2(128) = 56 + 7 = 63$.

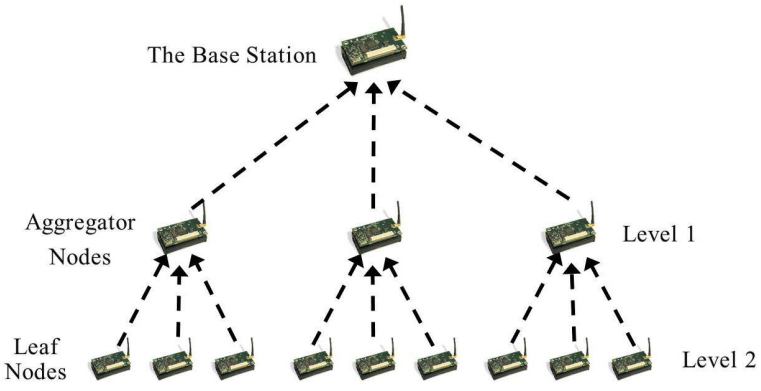

Figure 2. 3-ary tree topology.

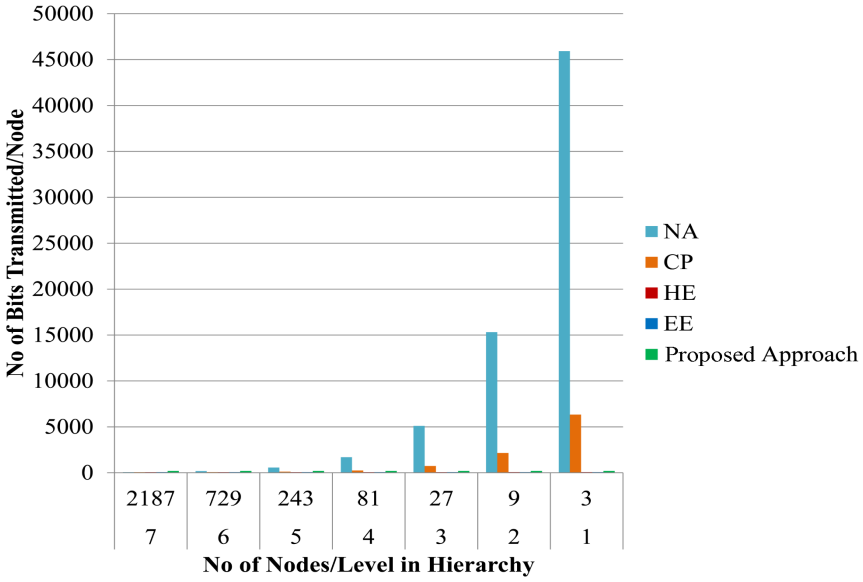

Figure 3. Communication overhead.

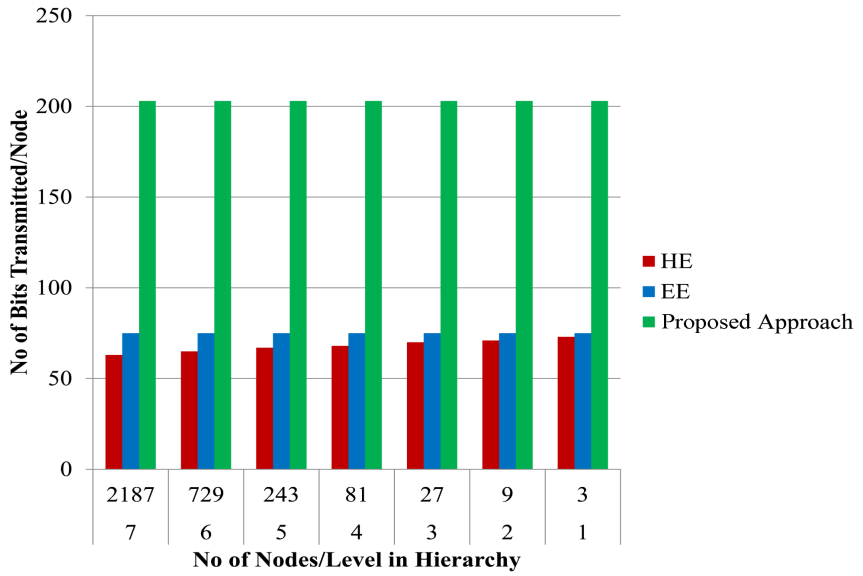

Figure 4. Communication overhead (en route aggregation).

For end-to-end encryption approach [18], the number of bits transmitted by leaf nodes are $\mathrm{HDR}+\log_2(128)+\log_2(2187)$. In our proposed approach, we provide authentication against both insider and outsider adversaries. We use MAC for en route authentication and homomorphic MAC for end-to-end authentication. Karlof *et al.* [35] suggested using a 4 byte MAC for authentication in WSNs. Due to the limited resources, the 4 byte MAC is proven to be secure for sensor networks. However, we consider an 8 byte MAC and an 8 byte Homomorphic MAC to calculate the bandwidth consumption. The reason to consider the 8 byte MAC is due to the fact that we consider a powerful and resource-rich adversary which may not have the same resource limitations like WSNs.

Although the proposed protocol requires few more bytes compared to the HE and EE approaches, it performs significantly better when there exist malicious adversaries. The schemes that do not support authentication are vulnerable to attackers that transmit malicious and fake packets toward the base station. Hence, they consume far more energy than the energy consumed during integrity verification.

6. Security Analysis

In this section, we measure the security strength of the proposed protocol with respect to some well-known cryptographic attacks against secure data aggregation protocols.

6.1. Known-Ciphertext Attack

In a known-ciphertext attack, an adversary tries to deduce a plaintext or a key from a set of known ciphertexts. Deterministic cryptosystems are vulnerable to such attacks where a plaintext is transformed into the same ciphertext. However, probabilistic cryptosystems that generate different ciphertexts for the same plaintext are resilient against this attack. In the proposed protocol, we use CMT cryptosystem as an underlying encryption algorithm. The CMT cryptosystem uses a pseudo-random function to generate a unique key before performing the encryption. Moreover, each key is used only once to perform the encryption. Therefore, ciphertexts generated using different keys will be different for the same plaintext. Hence, any outsider adversary who has access to ciphertexts, cannot deduce any information related to the key or plaintexts by analyzing ciphertexts.

6.2. Known-Plaintext Attack

In a known-plaintext attack, an adversary tries to deduce the key or recover the information about a plaintext form its ciphertext. In this attack, an adversary has access to some plaintext-ciphertext pairs through which it tries to recover the information about other plaintexts from their ciphertexts. In sensor networks, nodes are not deployed in a secure environment. Hence, compromised nodes can be used to generate such plaintext-ciphertext pairs. In the proposed protocol, each node is equipped with a unique key shared with the base station. Hence, a node can only have access to its own secret key. It does not have any information about other nodes' secret keys. Therefore, compromised nodes cannot leak any information about the remaining sensor nodes. In the proposed protocol, although we use a symmetric-key based cryptosystem for encryption, the unique keys embedded into the nodes help to thwart the known-plaintext attack.

6.3. The Sybil Attack

In a Sybil attack, a single malicious node can present itself as a large number of nodes [36] [37]. It can impersonate other nodes through fake identity information. In the proposed protocol, each node is equipped with a unique secret key shared with the base station. In addition, each node shares a pairwise secret key with its neighboring node(s). Hence, no malicious node can successfully impersonate other nodes or create fake identities, without having the keying information of those nodes. In our proposed protocol, the validation of the source node, through dual authentication mechanism, helps to ensure the protection against a Sybil attack. The protocols that do not have any authentication mechanism [11] [31] or the protocols that use a global shared secret key [11] [18] have been affected by the Sybil attack. In the proposed protocol, the pairwise unique secret keys provide the protection against the Sybil attack.

6.4. Node Capture Attack

In the proposed protocol, each node is equipped with two types of keys; 1) a unique secret key that is shared with the base station 2) a pairwise secret key with its neighboring node(s). If an adversary compromises a node, it can have access to the stored information therein. However, as we use a homomorphic encryption for privacy protection, the data once encrypted cannot be decrypted without the base station's secret key. Hence, any captured sensor node can only reveal its own sensor readings in a raw form. A compromised node cannot decrypt the ciphertexts encrypted with other sensor nodes' encryption keys. In addition, en route authentication and end-to-end authentication ensure that the captured node cannot violate the integrity of raw sensor readings or aggregated sensor readings without being detected. Hence, the proposed protocol protects the network against node capture attacks.

6.5. Malleability

Any cryptosystem that supports privacy homomorphism is inherently malleable. The malleability property helps aggregator nodes to process encrypted data. However, it also helps compromised intermediate nodes to aggregate

fake data packets without being detected. As our proposed protocol uses a symmetric key based homomorphic cryptosystem, it becomes malleable. The property that ensures the privacy of sensor readings has a negative effect on the data integrity. However, the risk of undesired malleability can be mitigated through authentication mechanisms. We use en route authentication and end-to-end authentication to verify the integrity of sensor readings. Hence, any maliciously aggregated data packets can be detected nearer to their sources. In conclusion, although we use inherently malleable homomorphic encryption for privacy protection, the authentication mechanisms help to mitigate the risk of undesired malleability and to ensure the correctness of sensor readings.

6.6. Denial of Service Attacks

The proposed protocol has a high degree of resistance against denial of service attacks. The use of symmetric-key based authentication mechanisms to verify the integrity of packets, reduces the energy consumption compared to asymmetric-key based digital signatures [38]. In addition, we provide en route authentication as well as end-to-end authentication using two different symmetric-key based authentication mechanisms. The en route authentication mechanism helps to thwart the pollution attacks, in which a single malicious node can flood the network with fake packets. Such attacks not only affect the correctness of gathered information, but they also affect the precious energy and therefore the lifetime of sensor networks. The en route authentication helps to detect maliciously injected packets nearer to their sources. Therefore, the proposed protocol reduces the energy consumption compared to the protocols where packets need to be forwarded up to the base station, in order to be verified.

7. Conclusion

In this paper, we present a way to ensure the integrity, privacy and en route aggregation of converge-cast traffic in wireless sensor networks. The proposed protocol is the first that achieves these objectives using only the symmetric key based mechanisms. We used two different homomorphic primitives namely, homomorphic encryption and homomorphic message authentication code. The proposed protocol protects the sensor readings against insiders as well as outsider adversaries. A comparison of the proposed protocol with existing protocols shows its viability and efficiency on resource constrained devices. Moreover, we analyze the security of the proposed protocol with respect to some well-known attacks on concealed data aggregation. We believe that the proposed protocol helps to improve the resource utilization in resource-constrained environments while achieving the desired security objectives.

Acknowledgements

This research was a part of the project “A Secure Data Aggregation System and An Intrusion Detection System for Wireless Sensor Networks”. It was supported by the Department of Electronics and Information Technology, Ministry of Communications and Information Technology, Government of India.

References

[1] MEMSIC (2014) MICAz Mote Platform. http://www.memsic.com/userfiles/files/Datasheets/WSN/6020-0060-04-B_MICAz.pdf

[2] MEMSIC (2014) TelosB Mote Platform. http://www.memsic.com/userfiles/files/Datasheets/WSN/6020-0094-02_B_TELOSB.pdf

[3] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) Wireless Sensor Networks: A Survey. *Computer Networks*: *The International Journal of Computer and Telecommunications Networking*, **38**, 393-422. http://dx.doi.org/10.1016/S1389-1286(01)00302-4

[4] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. and Pister, K. (2000) System Architecture Directions for Networked Sensors. *ACM SIGPLAN Notices*, **35**, 93-104. http://dx.doi.org/10.1145/356989.356998

[5] Fasolo, E., Rossi, M., Widmer, J. and Zorzi, M. (2007) In-Network Aggregation Techniques for Wireless Sensor Networks: A Survey. *IEEE Wireless Communications*, **14**, 70-87. http://dx.doi.org/10.1109/MWC.2007.358967

[6] Chan, H. and Perrig, A. (2003) Security and Privacy in Sensor Networks. *Computer*, **36**, 103-105. http://dx.doi.org/10.1109/MC.2003.1236475

[7] Perrig, A., Stankovic, J. and Wagner, D. (2004) Security in Wireless Sensor Networks. *Communications of the ACM*,

47, 53-57. http://dx.doi.org/10.1145/990680.990707

[8] Wang, Y., Attebury, G. and Ramamurthy, B. (2006) A Survey of Security Issues in Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, **8**, 2-23. http://dx.doi.org/10.1109/COMST.2006.315852

[9] Ozdemir, S. and Xiao, Y. (2009) Secure Data Aggregation in Wireless Sensor Networks: A Comprehensive Overview. *Computer Networks*: *The International Journal of Computer and Telecommunications Networking*, **53**, 2022-2037. http://dx.doi.org/10.1016/j.comnet.2009.02.023

[10] Girao, J., Schneider, M. and Westhoff, D. (2004) CDA: Concealed Data Aggregation in Wireless Sensor Networks. *Proceedings ACM Workshop on Wireless Security*, *WiSe*'04, Poster Presentation, Philadelphia.

[11] Girao, J., Westho, D. and Schneider, M. (2005) CDA: Concealed Data Aggregation for Reverse Multicast Traffic in Wireless Sensor Networks. *Proceedings of the* 40*th International Conference on Communications*, Seoul, 16-20 May 2005, 3044-3049.

[12] Rivest, R.L., Adleman, L. and Dertouzos, M.L. (1978) On Data Banks and Privacy Homomorphisms. *Foundations of Secure Computation*, **4**, 169-180.

[13] Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E. and Culler, D. (2005) TinyOS: An Operating System for Sensor Networks. In: Weber, W., Rabaey, J.M. and Aarts, E., Eds., *Ambient Intelligence*, Springer Berlin Heidelberg, Berlin, 115-148. http://dx.doi.org/10.1007/3-540-27139-2_7

[14] Hu, L. and Evans, D. (2003) Secure Aggregation for Wireless Networks. *Proceedings of the Symposium on Applications and the Internet Workshops*, Washington DC, 27-31 January 2003, 384-391.

[15] Przydatek, B., Song, D. and Perrig, A. (2003) SIA: Secure Information Aggregation in Sensor Networks. *Proceedings of the* 1*st International Conference on Embedded Networked Sensor Systems*, Los Angeles, 5-7 November 2003, 255-265.

[16] Sang, Y., Shen, H., Inoguchi, Y., Tan, Y. and Xiong, N. (2006) Secure Data Aggregation in Wireless Sensor Networks: A Survey. *Proceedings of the* 7*th International Conference on Parallel and Distributed Computing, Applications and Technologies*, Taipei, 4-7 December 2006, 315-320.

[17] Domingo-Ferrer, J. (2002) A Provably Secure Additive and Multiplicative Privacy Homomorphism. *Proceedings of the* 5*th International Conference on Information Security*, Berlin, 30 September-2 October 2002, 471-483.

[18] Castelluccia, C., Mykletun, E. and Tsudik, G. (2005) Efficient Aggregation of Encrypted Data in Wireless Sensor Networks. *Proceedings of the* 2*nd Annual International Conference on Mobile and Ubiquitous Systems*: *Networking and Services*, Washington DC, 17-21 July 2005, 109-117.

[19] Castelluccia, C., Chan, A.C.F., Mykletun, E. and Tsudik, G. (2009) Efficient and Provably Secure Aggregation of Encrypted Data in Wireless Sensor Networks. *ACM Transactions on Sensor Networks* (*TOSN*), **5**, 1-36. http://dx.doi.org/10.1145/1525856.1525858

[20] Peter, S., Piotrowski, K. and Langendoerfer, P. (2007) On Concealed Data Aggregation for Wireless Sensor Networks. *Proceedings of the* 4*th IEEE Consumer Communications Networking Conference*, Las Vegas, 11-13 January 2007, 192-196.

[21] Mykletun, E., Girao, J. and Westho, D. (2006) Public Key Based Cryptoschemes for Data Concealment in Wireless Sensor Networks. *Proceedings of the IEEE International Conference on Communications*, Istanbul, 11-15 June 2006, 2288-2295.

[22] Ugus, O. (2007) Asymmetric Homomorphic Encryption Transformation for Securing Distributed Data Storage in Wireless Sensor Networks. Technische Universität Darmstadt, Darmstadt.

[23] Koblitz, N. (1987) Elliptic Curve Cryptosystems. *Mathematics of Computation*, **48**, 203-209. http://dx.doi.org/10.1090/S0025-5718-1987-0866109-5

[24] Fontaine, C. and Galand, F. (2007) A Survey of Homomorphic Encryption for Nonspecialists. *EURASIP Journal on Information Security*, **2007**, 1-10. http://dx.doi.org/10.1155/2007/13801

[25] Dolev, D., Dwork, C. and Naor, M. (1991) Non-Malleable Cryptography. *Proceedings of the* 23*rd Annual ACM Symposium on Theory of Computing*, New York, 5-8 May 1991, 542-552.

[26] Racko, C. and Simon, D. (1992) Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In: Feigenbaum, J., Ed., *Advances in Cryptology—CRYPTO* '91, Springer Berlin Heidelberg, Berlin, 433-444.

[27] Chan, A.C.F. and Castelluccia, C. (2008) On the (Im)possibility of Aggregate Message Authentication Codes. *Proceedings of the IEEE International Symposium on Information Theory*, Toronto, 6-11 July 2008, 235-239.

[28] Agrawal, S. and Boneh, D. (2009) Homomorphic MACs: MAC-Based Integrity for Network Coding. *Proceedings of the* 7*th International Conference on Applied Cryptography and Network Security*, Paris-Rocquencourt, 2-5 June 2009, 292-305.

[29] Johnson, R., Molnar, D., Song, D.X. and Wagner, D. (2002) Homomorphic Signature Schemes. *Proceedings of the Cryptographers' Track at the RSA Conference*, San Jose, 18-22 February 2002, 244-262.

[30] Boneh, D., Freeman, D., Katz, J. and Waters, B. (2009) Signing a Linear Subspace: Signature Schemes for Network Coding. *Proceedings of the* 12*th International Conference on Practice and Theory in Public Key Cryptography*, Irvine, 18-20 March 2009, 68-87.

[31] Westho, D., Girao, J. and Acharya, M. (2006) Concealed Data Aggregation for Reverse Multicast Traffic in Sensor Networks: Encryption, Key Distribution, and Routing Adaptation. *IEEE Transactions on Mobile Computing*, **5**, 1417-1431. http://dx.doi.org/10.1109/TMC.2006.144

[32] Okamoto, T. and Uchiyama, S. (1998) A New Public-Key Cryptosystem as Secure as Factoring. *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, Espoo, 31 May-4 June 1998, 303-318.

[33] Paillier, P. (2000) Trapdooring Discrete Logarithms on Elliptic Curves over Rings. *Proceedings of the* 6*th International Conference on the Theory and Application of Cryptology and Information Security*, Kyoto, 3-7 December 2000, 573-584.

[34] Liu, A. and Ning, P. (2008) TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. *Proceedings of the* 7*th International Conference on Information Processing in Sensor Networks*, St. Louis, 22-24 April 2008, 245-256.

[35] Karlof, C., Sastry, N. and Wagner, D. (2004) TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. *Proceedings of the* 2*nd International Conference on Embedded Networked Sensor Systems*, Baltimore, 3-5 November 2004, 162-175. http://dx.doi.org/10.1145/1031495.1031515

[36] Karlof, C. and Wagner, D. (2003) Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. *Ad Hoc Networks*, **1**, 293-315. http://dx.doi.org/10.1016/S1570-8705(03)00008-8

[37] Newsome, J., Shi, E., Song, D. and Perrig, A. (2004) The Sybil Attack in Sensor Networks: Analysis & Defenses. *Proceedings of the* 3*rd International Symposium on Information Processing in Sensor Networks*, Berkeley, 26-27 April 2004, 259-268.

[38] Li, Z. and Gong, G. (2010) Data Aggregation Integrity Based on Homomorphic Primitives in Sensor Networks. *Proceedings of the* 9*th International Conference on Ad-Hoc, Mobile and Wireless Networks*, Edmonton, 20-22 August 2010, 149-162.

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or Online Submission Portal.