Data Aggregation Integrity Based on Homomorphic Primitives in Sensor Networks

Zhijun Li and Guang Gong

Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario, Canada
leezj@engmail.uwaterloo.ca, ggong@calliope.uwaterloo.ca

Abstract. Designing message integrity schemes for data aggregation is an imperative problem for securing wireless sensor networks. In this paper, we propose three secure aggregation schemes that provide provably secure message integrity with different trade-offs between computation cost, communication payload, and security assumptions. The first one is a homomorphic MAC, which is a purely symmetric approach, and is the most computation- and communication-efficient, but requires all data-collecting nodes to share one global key with the base station. The other two make use of (public key based) homomorphic hashing, combined with aggregate MAC and identity-based aggregate signature (IBAS) respectively. The scheme with aggregate MAC allows the base station to share a distinct key with every node, while the scheme with a paring-based IBAS enables all intermediate nodes beside the base station to verify the authenticity of aggregated messages.

1 Introduction

From the very beginning of wireless sensor networks (WSNs) development [13,2,30,14,3], it has been widely accepted that in-network data aggregation plays a critical role in the practicability and appealing of WSNs. In a typical sensor network, hundreds and thousands of low-cost sensor nodes scatter in a targeted area, collect environmental information, and collaboratively transmit data back to a base station. In many cases, users of sensor network applications are only interested in aggregated results after in-network processing, rather than detailed readings from individual nodes. On the other hand, data aggregation during message transmission is a natural way of preserving sensor nodes precious energy. Due to infeasibility of recharging nodes batteries in most circumstances, energy becomes the most valuable resource for sensor nodes. Among all nodes operations, data transmission consumes the most energy [2,3]. Moreover, in the absence of data aggregation, sensor nodes near the base station will suffer from heavy message transmission overhead, and then die of power exhaustion much sooner than other nodes, breaking down the whole network's functionality. Subsequently, data aggregation attracted a great deal of attention and many a data aggregation scheme has been proposed in recent years. Interested readers may refer to [30,14] for systematic surveys on this topic.

I. Nikolaidis and K. Wu (Eds.): ADHOC-NOW 2010, LNCS 6288, pp. 149–162, 2010.

When sensor nodes are deployed in a hostile environment, security measurements should be taken into consideration for network protocols. Attacks to wireless sensor networks not only come from outsider adversaries, but also can be conducted by compromised, previously legitimate nodes. Thus applicable secure protocols should prevent malicious inside nodes from damaging the whole network's functionality, or at least constrain their impacts to a reasonable level. Unfortunately, data aggregation, which requires intermediate nodes to process and change messages, and security objectives, one of which is preventing malicious manipulation, conflict with each other in this regard. As a result, designing secure and practical data aggregation schemes, which are critical to many sensor network applications, imposes an interesting and formidable challenge.

Resembling general security cases in other fields, message integrity might be one of the most important security objectives in sensor networks, and it should addressed by specific protocols. Generally, there are three kinds of message verification approaches for data aggregation: retroactive detection, abnormality-based detection, and cryptographic integrity primitives. Generally speaking, retroactive detection approaches, which involve substantial communication/interaction among the base station and sensor nodes to verify messages integrity, are not satisfactory because their costly performance penalty directly violates the intent of data aggregation. One may argue that the predictable data distribution can be used as a gauge to analyze and detect the abnormality of aggregated results; but the false rates are generally too high to be practical, and thus it is not a dependable solution. Consequently, schemes based on solid cryptographic primitives are usually desirable. Unfortunately, conventional cryptographic integrity primitives, such as message authentication code (MAC) and signature, are not compatible with data aggregation scenarios.

Contributions. Based on new cryptographic homomorphic primitives [1,24,17], we propose three secure aggregation schemes that provide provably secure message integrity. The first one is a homomorphic MAC scheme for data aggregation, which is a revised version of the homomorphic MAC proposal on secure network coding application in [1]. This homomorphic MAC scheme, other than revisions to fit data aggregation scenarios, achieves a little bit performance improvement, as we observe and then remove an unnecessary step in the original scheme. The homomorphic MAC scheme is computation- and communication-efficient, but with one inherent restriction: all data-collecting nodes share one global key with the base station. The assumption that all those nodes are tamper-proof might be too strong to be realistic in many sensor network applications. In order to overcome this drawback, we further propose two secure aggregation schemes based on homomorphic hashing [24,17], at the expense of increasing communication and computation costs. One is to combine homomorphic hashing with aggregate MAC [23], in which every node shares a different key with the base station, while the other is with identity-based aggregated signature [19], which enables intermediate nodes to verify the authenticity of messages. The proposed three protocols present different trade-offs between computation, communication, security and can fit a wide variety of application areas.

Organization. The remainder of the paper is organized as follows. The related work is introduced in Section 2. Then we state the data aggregation network setting along with the security objective, discuss homomorphic primitives and define homomorphic MAC as well as homomorphic hashing in Section 3. Afterward, secure aggregation integrity schemes based on homomorphic MAC and homomorphic hashing are proposed and discussed in Section 4 and Section 5 respectively. Finally, Section 6 concludes the paper.

2 Related Work

Hu and Evans [20] described a secure hop-by-hop data aggregation scheme, in which every node shares with the base station a different key, from which temporary session MAC keys will be derived, and by adopting hash-chain-based delayed message authentication, such as μTESLA [27], intermediate modes, after the base station reveals session MAC keys, will be able to verify the integrity of messages that they buffered. This scheme suffers from communication penalties, as the introduction of μTESLA for distributing session MAC keys incurs considerable communication cost. More disturbingly, in order to detect one inside malicious node that manipulates other nodes input, intermediate nodes have to obtain and buffer all their grandchildren's messages and corresponding MACs, that is, two-hop messages buffer only being able to detect *one* misbehavior node. Although Jadia and Muthuria [21] extended the Hu-Evans scheme by all two nodes in the two-hop communication range sharing pairwise keys and then the scheme eliminates the usage of μTESLA, the fact that both schemes are only capable of preventing a single inside malicious node at an appreciable communication cost makes them impractical.

Yang *et al.* [32] presented a secure hop-by-hop data aggregation protocol for sensor networks named SDAP, using the principles of divide-and-conquer and commit-and-attest, which is a typical example of retroactive detection approach. In SDAP, a probabilistic grouping technique is utilized to dynamically partition the nodes in a tree topology into subtrees. A commitment-based hop-by-hop aggregation is conducted in each subtree to generate a group aggregate, and accordingly the base station identifies the suspicious subtrees based on the set of group aggregates. Finally, each subtree under suspect participates in an attestation procedure to prove the correctness of its group aggregate. Those complicated algorithms cause significant transmission overhead, and may cancel off all communication benefits from data aggregation.

Przydatek, Song, and Perrig [29] proposed secure information aggregation (SIA) to identify forged aggregation values from malicious nodes. In the SIA scheme, a special node named aggregator computes an aggregation result over raw data together with a commitment to the data based on a Merkle-hash tree and sends them back to a remote user, which later challenges the aggregator to verify the aggregation. Later Chan, Perrig, and Song built on the aggregate-commit-prove framework in [29] but extended their single aggregator model to a fully distributed setting. Frikken and Dougherty [16] further improved the

Chan-Perrig-Song scheme. Moreover, Chan and Perrig [10] derived several security primitives form this kind of algorithms.

3 Preliminary

3.1 Network Setting and Security Objective

We consider a sensor network that consists of n sensor nodes which are highly sensitive of energy consumption, and a base station that is only concerned about the statistical results, mainly mean and variance. Thus a data aggregation mechanism is implemented in the sensor network.

Since loose time synchronization among sensor nodes is indispensable for efficient message aggregation and the sensor network is under attacks, it is assumed that there is a secure time synchronization scheme [28] available in the sensor network. We do not explore a specific secure time synchronization selection because it is independent and relatively irrelevant. At a designated time, the sensor network outputs a *report*, which is an overall aggregated result for a task and is *uniquely* identified by a report identifier rid. The report identifier may be the task description combined with the reporting time. It is clear that all sensor nodes should have an agreement on the report identifier specification and know how to correctly generate rid. Otherwise, nodes cannot distinguish messages of different kinds and data aggregation cannot be properly performed.

There are three kinds of roles in the sensor network: a *contributor* that collects environmental readings and generate a *raw message*, an *aggregator* that aggregates all messages that it received plus possibly its own raw message and then produces an *aggregated message*, an *verifier* that verifies the authentication of messages it received. A node may play some of or all the three roles, while the base station is definitely a verifier.

The data are aggregated though the network, and the base station eventually retrieves an aggregated result, i.e. the report. In order to produce the mean of a measurement, it suffices for the base station to retrieve the sum of the samples and the number of contributors. If the variance is desired, the contributors should also provide the squares of their readings and the aggregators accordingly merge the squares. By the mean, the number of contributors and the sum of the squares, one can readily calculate the variance as a basic statistical equation. In other words, we only need to consider an additive aggregation. For the sake of simplicity, we assume that the sensor network is organized as a tree structure rooted on the base station, though our proposed schemes fit into any kind of data additive aggregation architecture.

In addition, to support advanced aggregation requirements, the concept of *weight* is introduced. Specifically, we allow that the measurements of different nodes have different weights for their contributions to the final report. In most cases, node weights are uniform; when different weights are required, we assume that aggregators and the base station are aware of the weights of messages contributors, either via an established agreement, or from explicit indications attached to messages.

The primary objective of our proposals is to provide the message integrity for data aggregation in a cryptographic manner, thus an authentication segment that facilitates verification shall be appended to a message. Generally speaking, it is impossible for a verifier to validate the integrity of an aggregated message without the knowledge of its contributors. This is because if contributors use different keys, the verifier certainly needs knows who those contributors are before using those keys in the verification stage; if a global key is employed and a verifier cannot retrieve contributors of messages, an adversary may easily construct a malicious message to pass the integrity verification by aggregating a single message from one contributor many times, say b times, which is indistinguishable with an aggregated message resulting from b legitimate contributors. In other words, data origin authentication is an inherent requirement for data aggregation integrity.

The simplest way of indicating data origin is to attach the list of contributors to a message. To avoid the communication cost in this approach, we may utilize a mechanism that allows a verifier to implicitly obtain the contributor list, such as derivation from the network topology. This is pretty realistic for the base station as the ultimate verifier. In a case that a verifier is capable of identify all potential contributors, of which only a small fraction do not really participate in a message contribution, a list of exclusive nodes rather than the contributors may be appended to the message. Due to the space limitation, we do not elaborate the techniques of efficiently providing the contributor list for verifiers. Henceforth, we simply assume that an aggregator knows the appropriate weights to aggregate messages, and a verifier of a message can obtain its contributors and corresponding weights. When we discuss a scheme's communication cost, we do not consider the payload from contributor lists and weights, because, as we argued, there might be mechanisms to avoid it, or it is inevitable for message authentication.

As a typical application scenario of this network setting, a sensor network is employed to routinely detect environmental information, such as temperature, humidity, radiation. Every node senses data in a hourly interval, and submits the results on a daily basis. For example, at two o'clock every day, starting from all leaf nodes, messages are transmitted and aggregated over a spanning tree.

3.2 Homomorphic Primitives

Homomorphic property in cryptographic operations may be very useful in a variety of applications, and thus stimulates research on homomorphic primitives, namely homomorphic encryption, homomorphic MAC, homomorphic hashing, and homomorphic signature. Homomorphic encryption [15], in which a user without a decryption key can perform algebraic operations on ciphertext to achieve designated operation results on the corresponding plaintext, has been studied for decades, and recently, an outstanding result, fully homomorphic encryption [18], was proposed, which allows arbitrary operations on ciphertext (and so on plaintext). Even though the only two fully homomorphic encryption schemes [18,12] by now have not provided competitive performance for most applications, they

do reveal a perspective on a powerful, widely demanded technique and we expect that practical schemes will eventually emerge. Those homomorphic encryption schemes shall provide a solid foundation for data confidentiality of aggregated messages. As for homomorphic signature, current schemes [22,8] are mainly aimed at one-sender many-recipients secure multi-cast scenarios, with costly computation overhead (compare to symmetric primitives), thus they may not be suitable for secure data aggregation integrity of WSNs. In contrast, homomorphic MAC and homomorphic hashing can be effectively used to construct message integrity schemes of supporting additive aggregation with weights. Formally, homomorphic MAC and homomorphic hashing are defined as follows.

Definition 1 (Homomorphic MAC [1]). *A homomorphic MAC should satisfy the following properties:*

1. Homomorphism. *Given two (message, tag) pairs* $(\boldsymbol{m}_1, t_1)$ *and* $(\boldsymbol{m}_2, t_2)$*, anyone can create a valid tag* t_a *for an aggregated message* $\boldsymbol{m}_a = w_1\boldsymbol{m}_1 + w_2\boldsymbol{m}_2$ *for any scales* w_1, w_2 *as weights. Typically,* $t_a = w_1 t_1 + w_2 t_2$*.*
2. Security against Chosen Message Attack. *Even under a chosen message attack, in which an adversary is allowed to query tags of polynomial number of messages, it is still infeasible for the adversary to create a valid tag for a message other than an linear combination of some previously queried messages.*

A homomorphic MAC consists of three probabilistic, polynomial-time algorithms (Sign*,* $\mathsf{Aggregate}$*,* Verify*)*

- $t_u = \mathsf{Sign}(k, \mathsf{rid}, \boldsymbol{m}_u, \mathsf{id}_u)$*: node* u *with ID* id_u*, as a contributor of a raw message* $\boldsymbol{m}_u$ *regarding report* rid*, computes a tag* t_u *for* $\boldsymbol{m}_u$ *using* k *as the key.*
- $t = \mathsf{Aggregate}((\boldsymbol{m}_1, t_1, w_1), \ldots, (\boldsymbol{m}_j, t_j, w_j))$*: an aggregator implements the homomorphic property for message-tag pairs in the absence of key* k*, that is, generates a tag* t *for the aggregated message* $\boldsymbol{m} = \sum_{i=1}^{j} w_i \boldsymbol{m}_i$
- $\mathsf{Verify}(k, \mathsf{rid}, \boldsymbol{m}, t)$*: a verifier verifies the integrity of message* $\boldsymbol{m}$ *regarding report* rid *by key* k *and tag* t*.*

The homomorphic MAC scheme is first defined and proposed in [1], intended to provide secure network coding. The definition above is equivalent to that in [1], with emphasis on the data aggregation.

Definition 2 (Homomorphic Hashing [24,17]). *A homomorphic hash function* H *is a hash function satisfying:*

1. Homomorphism. *For any two messages* $\boldsymbol{m}_1$*,* $\boldsymbol{m}_2$ *and scalars* w_1*,* w_2*, it holds that* $H(w_1\boldsymbol{m}_1 + w_2\boldsymbol{m}_2) = H(\boldsymbol{m}_1)^{w_1} H(\boldsymbol{m}_2)^{w_2}$*.* [1]

[1] Intuitively, the homomorphic equation should be $H(w_1\boldsymbol{m}_1 + w_2\boldsymbol{m}_2) = w_1 H(\boldsymbol{m}_1) + w_2 H(\boldsymbol{m}_2)$. In fact, that just uses a different notation on group operation and essentially they are equivalent.

2. Collision Resistance. *There is no probabilistic polynomial-time (PPT) adversary capable of forging* $(\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3, w_1, w_2)$ *satisfying both* $\boldsymbol{m}_3 \neq w_1\boldsymbol{m}_1 + w_2\boldsymbol{m}_2$ *and* $H(\boldsymbol{m}_3) = H(\boldsymbol{m}_1)^{w_1} H(\boldsymbol{m}_2)^{w_2}$.

The homomorphic hashing can be used in many applications, such as secure network coding [17], secure peer-to-peer content distribution using erasure codes [24].

4 Secure Aggregation with Homomorphic MAC

Our first proposal is a specific homomorphic MAC scheme that fully complies with Definition 1. Basically, the scheme is a revised version of the homomorphic MAC for network coding proposed by Agrawal and Boneh (AB Scheme) [1].

4.1 Scheme Description

To formally present our schemes, message $\boldsymbol{m}$ is formed as d segments of l bits. Let $q = 2^l$, then the message space is $\mathbb{F}_q^d$. In other words, message $\boldsymbol{m}$ can be represented as a vector of d segments: $(m_1, m_2, \cdots, m_d)$, where $m_i \in \mathbb{F}_q, i = 1, 2, \cdots, d$. As the additive operation is over finite field $\mathbb{F}_q$, q should be greater than the bound of the desired data sum. We stress that this is also an inherent requirement in the data aggregation.

To generate and verify tags, all contributors and verifiers share one global MAC key that consists of (k_1, k_2). Naturally, those nodes should be tamper-proof to protect the protocol security. Let $\mathcal{K}_1$ and $\mathcal{K}_2$ denote the key spaces of k_1 and k_2 respectively, $\mathcal{I}$ denote the space of node identities, and $\mathcal{R}$ denote the space of report identifiers. Two pseudo random functions are required: $R_1 : \mathcal{K}_1 \rightarrow \mathbb{F}_q^d$ and $R_2 : (\mathcal{K}_2 \times \mathcal{R} \times \mathcal{I}) \rightarrow \mathbb{F}_q$.

The three algorithms (Sign, Aggregate, Verify) are given as follows.

- $\mathsf{Sign}(k, \mathsf{rid}, \boldsymbol{m}_u, \mathsf{id}_u)$, by node u as a contributor
  1. $\boldsymbol{a} = R_1(k_1) \in \mathbb{F}_q^d$.
  2. $b_u = R_2(k_2, \mathsf{rid}, \mathsf{id}_u) \in \mathbb{F}_q$.
  3. $t_u = \boldsymbol{a} \circ \boldsymbol{m}_u + b_u \in \mathbb{F}_q$, where $\circ$ stands for the inner product of two vectors $\boldsymbol{a}$ and $\boldsymbol{m}_i$ over finite field $\mathbb{F}_q$, that is, $\boldsymbol{a} \circ \boldsymbol{m}_u$ is equal to $a_1 m_{u,1} + a_2 m_{u,2} + \cdots + a_d m_{u,d} \mod q$.
- $\mathsf{Aggregate}((\boldsymbol{m}_1, t_1, w_1), \ldots, (\boldsymbol{m}_j, t_j, w_j))$, by an aggregator
  1. $\boldsymbol{m} = \sum_{i=1}^{j} w_i \boldsymbol{m}_i \in \mathbb{F}_q^d$, in which the additive operation is over $\mathbb{F}_q$.
  2. $t = \sum_{i=1}^{j} w_i t_i \in \mathbb{F}_q$.
- $\mathsf{Verify}(k, \mathsf{rid}, \boldsymbol{m}, t)$, by a verifier with the knowledge of contributor identities and weights
  1. $\boldsymbol{a} = R_1(k_1) \in \mathbb{F}_q^d$.
  2. $b = \sum_{i=1}^{j} [w_i \cdot R_2(k_2, \mathsf{rid}, \mathsf{id}_i)] \in \mathbb{F}_q$.
  3. if $\boldsymbol{a} \circ \boldsymbol{m} + b = t$ outputs "ACCEPT"; otherwise outputs "REJECT".

4.2 Discussion and Comparison

By the same reductionist proof of Theorem 2 in [1], this scheme is probably secure against chosen message attack based on the pseudo-randomness of R_1 and R_2. Since the tag size is l-bit, in order to achieve 80-bit security level, l should not be less than 80.

To support secure network coding, the space $\mathcal{I}$ in the AB scheme [1] is $\mathbb{F}_q^c$, albeit id_i is a vector base identifier, rather than a node id, and c is the number of vector base. Since every message in the network coding should include a vector in $\mathbb{F}_q^c$ to indicate the combination coefficients of c vector bases, which are analogue to weights in the data aggregation, usually $q = 2^8$ is recommended (as in the AB Scheme) to save communication cost while maintaining high success decoding probability for random network coding. Such a small q, however, undermines the security level, as the tag size would be 8-bit and an adversary can fake a message's tag at least with probability 1/256. Fortunately, the data aggregation does not suffer that limitation—the weights are not randomly chosen by aggregators. Therefore, we can safely use $q \geq 2^{80}$.

In addition, the AB homomorphic MAC scheme specifies $R_1 : \mathcal{K}_1 \rightarrow \mathbb{F}_q^{d+c}$, $\boldsymbol{a} = R_1(k_1) \in \mathbb{F}_q^{d+c}$, and then $t_u = \boldsymbol{a} \circ (\boldsymbol{m}_u || \mathsf{id}_u) + b$. We observe that the occurrence of id_u in $(\boldsymbol{a} \circ (\boldsymbol{m}_i || \mathsf{id}_u))$ is unnecessary and then it is removed in our revision because id_u has been used in the computation of $b = R_2(k_2, \mathsf{rid}, \mathsf{id}_u)$. This modification slightly improves the computation performance and can apply to both network coding and data aggregation scenarios.

We notice that our proposed scheme has a similar structure to the data aggregation MAC scheme proposed by Castelluccia *et al.* (CCMT scheme) [9]. In their scheme, the space of message $\boldsymbol{m}$ is limited to $\mathbb{F}_q$, which means that the tag is as long as the maximal length of messages. This approach violates a principle on MAC that a MAC scheme should support arbitrary length of message and output short, fixed length of tags. Admittedly, the length of messages in our scheme has to be determined beforehand, but it is a basic requirement for data aggregation. In addition, the CCMT scheme does not supply a reductionist security proof; Theory 2 in [9] pertaining to the scheme security is more like an argument than a proof. Nonetheless, the CCMT scheme provides a necessary integrity scheme for data aggregation, and our homomorphic MAC scheme can be thought as the combination of the CCMT scheme and the AB scheme.

The security of the proposed MAC scheme relies on the pseudo-randomness of R_1 and R_2. In principle, all provably secure pseudo-random generators are public-key based[2], involving heavy computation. As a widely employed method, we may use AES [11] to implement R_1 and R_2. In this way, the proposed scheme is very computationally efficient, and the key lengths of k_1 and k_2 are 128-bit. On the other hand, a 80-bit tag would suffice to allow a verifier to check the authenticity of an aggregated message, which presents the optimal communication overhead. One inherent drawback in homomorphic MACs is that one

[2] A public-key based approach does not necessarily indicate that it involves public/private keys; instead, it implies that the approach employs typical public-key cryptosystem operations, such as exponentiation over a big group.

single MAC key is shared by all contributors and verifiers. If sensor nodes are not tamper-proof and one of them is compromised by an adversary, the whole system security is breached.

5 Two Schemes Based on Homomorphic Hashing

In order to overcome the drawback of one global MAC key in the previous scheme, we propose two schemes based on homomorphic hashing.

5.1 Construction of Homomorphic Hashing

The first step is to find a homomorphic hashing function suitable for sensor networks. At present, there are only two homomorphic hashing functions: one is based on the hardness of discrete logarithm [24], and the other is based on the intractability of integer factorization [17].

Discrete Logarithm [24]. Let $\mathbb{G}$ be a cyclic group of prime order p in which the discrete logarithm problem is hard, and the public parameters contain a description of $\mathbb{G}$ and d random generators $g_1, g_2. \cdots, g_d \in \mathbb{G}$. Then a homomorphic hashing on message $\boldsymbol{m} = (m_1, m_2, \cdots, m_d) \in \mathbb{Z}_p^d$ can be constructed by

$$H(\boldsymbol{m}) \stackrel{\text{def}}{=} \prod_{i=1}^{d} g_i^{m_i}. \tag{1}$$

It is easy to verify that the homomorphic property is satisfied in this construction, and the collision resistance is guaranteed by the hardness of the discrete logarithm problem in $\mathbb{G}$.

Integer Factorization [17]. Let N be the product of two safe primes[3] so that the group $\mathbb{Q}_N$ of quadratic residues modulo n is cyclic, and let $g_1, g_2, \cdots, g_d$ be generators of $\mathbb{Q}_N$. Then a homomorphic hashing on message $\boldsymbol{m} = (m_1, m_2, \cdots, m_d) \in \mathbb{Z}_N^d$ can be constructed by

$$H_N(\boldsymbol{m}) \stackrel{\text{def}}{=} \prod_{i=1}^{d} g_i^{m_i} \mod N. \tag{2}$$

Finding a collision is computationally equivalent to factoring N, which is intractable.

Comparison. The homomorphic hashing function (2) can use the form of $H_N(\boldsymbol{m}) = 2^{\boldsymbol{m}} \mod N$ by choosing a proper N such that 2 is a generator of $\mathbb{Q}_N$ and the integer value converted from any message $\boldsymbol{m}$ is less than N. Subsequently, it presents some computational advantage over hashing function (1) by

[3] A prime number p is a safe prime if $(p-1)/2$ is also a prime.

fast exponentiation. However, then its hash value size, which is the same as the size of N, exceeds the message size. This is unacceptable in the data aggregation of sensor networks. Even for the basic form (2), in order to provide 80-bit security, N is at least 1024-bit, while by using elliptic curve cryptography (ECC), the hash value size of function (1) can be approximately as low as 160-bit. Moreover, the practicability of implementing ECC in low-cost sensor nodes has been successfully demonstrated in [25,31]. Therefore, the suitable homomorphic hashing for secure data aggregation integrity in WSNs should be function (1).

5.2 Aggregation Integrity by Homomorphic Hashing

Since we choose the homomorphic hashing function (1), the message space is $\mathbb{F}_p^d$, where p is a prime number and $p \geq 2^{160}$ for 80-bit security. For a raw message $\boldsymbol{m}_i$, node i computes a raw hash value $h_i = H(\boldsymbol{m}_i)$, and uses a mechanism to sign h_i, which will be specified later, in a way that allows verifiers to verify the authenticity of h_i. When a verifier receives an aggregated message $\boldsymbol{m} = \sum_{i=1}^{j}(w_j \boldsymbol{m}_i)$ along with j pairs of (raw hashing value, weight) (h_i, w_i), it first determines whether the hashing values are valid, and then verifies the message's integrity by checking whether

$$\prod_{i=1}^{j} h_i^{w_i} \stackrel{?}{=} H(\boldsymbol{m}).$$

This scheme is proven secure in the standard model via reductionist from the discrete logarithm problem [24,8], when raw hash values are authenticated by a secure mechanism. In the following two subsections, we describe two communication-efficient mechanisms to authenticate h_i.

5.3 Authentication by Aggregate MAC

Aggregate MAC [23] presents the property that multiple MAC tags, computed by different contributors on multiple raw hash values, can be aggregated into a single tag that can be verified by a verifier who shares a distinct key with each contributor. The construction of aggregate MAC has been long known. In fact, an aggregate MAC which is provably secure [23] can be constructed from essentially any standard message authentication code as follows.

For simplicity, we assume that the base station is the sole verifier. Let k_i be the symmetric key shared by node i and the base station, Mac be a standard deterministic MAC, for example: CBC-MAC [5], HMAC [4]. To authenticate a raw hash value h_i, node i generates a tag: $t_i = \mathsf{Mac}_{k_i}(\mathsf{rid}, h_i)$. Any aggregator can aggregate j tags by simply computing the XOR of all the tag values: $t = \bigoplus_{i=1}^{j} t_i$. Then the base station uses the aggregate tag t to verify the authenticity of all raw hash values by checking whether

$$t \stackrel{?}{=} \bigoplus_{i=1}^{j} \mathsf{Mac}_{k_i}(\mathsf{rid}, h_i).$$

5.4 Authentication by Identity-Based Aggregate Signature

Aggregate MACs, like all other symmetric-key MACs, demand verifiers to comprehend contributors keys. In many circumstances, it would be much appreciated that all intermediate nodes can verify the authenticity of raw hash values (and then aggregated messages). In terms of communication cost, the best scheme providing such a property is an identity-based aggregate signature (IBAS), in which different raw hash values produced by many different contributors, whose public keys are their identities, can be authenticated by one single aggregate signature.

As far as we know, there are three IBAS schemes which are provably secure: GR scheme [19], BN scheme [6], and BGOY scheme [7]. The BN scheme [6] requires interactions of all signers, and the BGOY scheme demands a sequential signature aggregation procedure; thus both are not suitable for secure aggregation in WSNs. One presumably too strong assumption in the GR scheme [6] is that all signers must use a same unique string when signing, which, fortunately, is not a problem at all in the WSN secure aggregation application, because an unique rid for every report is known to all nodes.

GR Paring-Based IBAS Scheme [19]. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of some large prime order q that efficiently support a bilinear mapping $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. That is, $\hat{e}(aQ, bR) = \hat{e}(Q, R)^{ab}$ for all $Q, R \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}$. The GR IBAS scheme works as follows.

- *Setup*: To set up the scheme, a private key generator (PKG)
  1. generates groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q and an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.
  2. chooses an arbitrary generator $P \in \mathbb{G}_1$.
  3. picks a random $s \in \mathbb{Z}/q\mathbb{Z}$ as the master key of PKG and sets $Q = sP$.
  4. chooses three cryptographic hash functions $H_1, H_2 : \{0,1\}^* \to \mathbb{G}_1$ and $H_3 : \{0,1\}^* \to \mathbb{Z}/q\mathbb{Z}$.
- *Private key generation*: Node i receives from the PKG the values of $sP_{i,\alpha}$ as its private key for $\alpha \in \{0,1\}$, where $P_{i,\alpha} = H_1(\mathsf{id}_i, \alpha) \in \mathbb{G}_1$.
- *Signing*: To sign $\boldsymbol{h}_i$, node i
  1. computes $P_{\mathsf{rid}} = H_2(\mathsf{rid}) \in \mathbb{G}_1$.
  2. computes $c_i = H_3(h_i, \mathsf{id}_i, \mathsf{rid}) \in \mathbb{Z}/q\mathbb{Z}$.
  3. generates random $r_i \in \mathbb{Z}/q\mathbb{Z}$.
  4. computes signature (S_i, T_i), where $S_i = r_i P_{\mathsf{rid}} + sP_{i,0} + c_i sP_{i,1}$ and $T_i = r_i P$.
- *Signature Aggregation*: Signatures (S_i, T_i) for $1 \le i \le j$ can be aggregated into (S, T), where $S = \sum_{i=1}^{j} S_i$, and $T = \sum_{i=1}^{j} T_i$.
- *Verification*: Any node can verify the signature by checking whether

$$\hat{e}(S, P) \stackrel{?}{=} \hat{e}(T, P_{\mathsf{rid}})\ \hat{e}(Q, \sum_{i=0}^{j} P_{i,0} + \sum_{i=0}^{j} c_i P_{i,1}).$$

This scheme is proven secure in the random oracle model, on the assumption of hardness of computational Diffie-Hellman problem.

Generally speaking, paring is a highly computation-intense operation and more costly than ordinary public key based operations. Consider the fact that identity-based schemes eliminate the cost of transmitting nodes public keys and most of practical identity-based encryptions are paring-based, the use of the GR paring-based IBAS scheme in the secure WSN data aggregation is justifiable. In addition, TinyPBC [26] which implements and measures paring operations give an affirmative answer to the question of whether paring is feasible in the WSNs, albeit their paring implementation is understandably slow.

5.5 Discussion

To verify the integrity of an aggregated message, a verifier should retrieve the raw hash values of the contributors, which constitutes considerate communication payload and is an instinctive downside for homomorphic-hashing-based approaches. When the message size in a application does not exceed the homomorphic hashing result size (160-bit typically), then the homomorphic hashing is redundant, and directly applying aggregate MAC/signature to raw messages is preferred. If the message size is substantially greater than the hashing value size, which is quite common for WSNs, then using homomorphic hashing would significantly reduce the communication cost, as in the application scenario described in Section 3.1.

For 80-bit security, the signature of the GR scheme is roughly 320-bit, while a typical aggregate MAC tag is 80-bit. The third scheme, which combines homomorphic hashing function (1) with the GR identity-based aggregate signature, provides the most promising security for data aggregation integrity. Since all intermediate nodes are capable of verifying the integrity of (raw or aggregated) messages in that scheme, a node (or an adversary) that tries to inject invalid messages into the sensor network can be easily caught. It is worth to notice that this is achieved at the computational cost of paring operation by intermediate nodes. By contrast, the second scheme (with aggregate MAC) does not require paring operation and is useful in practice. If detecting an invalid (message, tag) pair in the second scheme, the base station can require the corresponding child to submit its aggregation record, and then interacts with grandchildren until reaching leaf nodes. In this way, the base station can determine which nodes should be responsible for faking messages and then expels them from the network.

6 Conclusion

In this paper, we present three secure aggregation schemes that provide provably secure message integrity with different trade-offs between computation cost, communication payload, and security assumptions. The first proposal is a concrete homomorphic MAC scheme for WSN data aggregation integrity, and the other two are combining homomorphic hashing with aggregate MAC and identity-based aggregate signature respectively. We detail on the selections and constructions of those three cryptographic primitives and discuss their practicability on wireless sensor networks.

Acknowledgment

The research is supported by the NSERC Strategic Project Grants.

References

1. Agrawal, S., Boneh, D.: Homomorphic MACs: MAC-Based Integrity for Network Coding. In: ACNS 2009. LNCS, vol. 5536, pp. 292–305. Springer, Heidelberg (2009)
2. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine 40(8), 102–114 (2002)
3. Baronti, P., Pillai, P., Chook, V.W.C., Chessa, S., Gotta, A., Hu, Y.F.: Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards. Computer Communications 30(7), 1655–1695 (2007)
4. Bellare, M., Canetti, R., Krawczyk, H.: Keying Hash Functions for Message Authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (1996)
5. Bellare, M., Kilian, J., Rogaway, P.: The security of the cipher block chaining message authentication code. Journal of Computer and System Sciences 61(3), 362–399 (2000)
6. Bellare, M., Neven, G.: Identity-Based Multi-signatures from RSA. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 145–162. Springer, Heidelberg (2006)
7. Boldyreva, A., Gentry, C., O'Neill, A., Yum, D.H.: Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 276–285. ACM, Alexandria (2007)
8. Boneh, D., Freeman, D., Katz, J., Waters, B.: Signing a Linear Subspace: Signature Schemes for Network Coding. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 68–87. Springer, Heidelberg (2009)
9. Castelluccia, C., Chan, A.C.F., Mykletun, E., Tsudik, G.: Efficient and provably secure aggregation of encrypted data in wireless sensor networks. ACM Trans. Sen. Netw. 5(3), 1–36 (2009)
10. Chan, H., Perrig, A.: Efficient security primitives derived from a secure aggregation algorithm. In: Proceedings of the 15th ACM Conference on Computer and Communications Security. ACM, Alexandria (2008)
11. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer, Heidelberg (2002)
12. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully Homomorphic Encryption over the Integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010)
13. Estrin, D., Govindan, R., Heidemann, J., Kumar, S.: Next Century Challenges: Scalable Coordination in Sensor Networks. In: Proceedings of the 5th ACM/IEEE International Conference on Mobile Computing and Networking, pp. 263–270. IEEE Computer Society, Seattle (1999)
14. Fasolo, E., Rossi, M., Widmer, J., Zorzi, M.: In-network aggregation techniques for wireless sensor networks: a survey. IEEE Wireless Communications 14(2), 70–87 (2007)
15. Fontaine, C., Galand, F.: A survey of homomorphic encryption for nonspecialists. EURASIP Journal on Information Security 2007(1), 1–15 (2007)

16. Frikken, K.B., Dougherty IV, J.A.: An efficient integrity-preserving scheme for hierarchical sensor aggregation. In: Proceedings of the first ACM Conference on Wireless Network Security, pp. 68–76. ACM, Alexandria (2008)
17. Gennaro, R., Katz, J., Krawczyk, H., Rabin, T.: Secure Network Coding Over the Integers. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 142–160. Springer, Heidelberg (2010)
18. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, pp. 169–178. ACM, Bethesda (2009)
19. Gentry, C., Ramzan, Z.: Identity-Based Aggregate Signatures. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T.G. (eds.) PKC 2006. LNCS, vol. 3958, pp. 257–273. Springer, Heidelberg (2006)
20. Hu, L., Evans, D.: Secure aggregation for wireless networks. In: Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT 2003 Workshops), pp. 384–391 (2003)
21. Jadia, P., Mathuria, A.: Efficient Secure Aggregation in Sensor Networks. In: Bougé, L., Prasanna, V.K. (eds.) HiPC 2004. LNCS, vol. 3296, pp. 40–49. Springer, Heidelberg (2004)
22. Johnson, R., Molnar, D., Song, D., Wagner, D.: Homomorphic Signature Schemes. In: Preneel, B. (ed.) CT-RSA 2002. LNCS, vol. 2271, pp. 244–245. Springer, Heidelberg (2002)
23. Katz, J., Lindell, A.: Aggregate Message Authentication Codes. In: Malkin, T.G. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 155–169. Springer, Heidelberg (2008)
24. Krohn, M.N., Freedman, M.J., Mazières, D.: On-the-fly verification of rateless erasure codes for efficient content distribution. In: IEEE Symposium on Security and Privacy 2004, pp. 226–240 (2004)
25. Liu, A., Ning, P.: TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. In: International Conference on Information Processing in Sensor Networks (IPSN 2008), pp. 245–256 (2008)
26. Oliveira, L.B., Scott, M., Lopez, J., Dahab, R.: TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks. In: 5th International Conference on Networked Sensing Systems, INSS 2008, pp. 173–180 (2008)
27. Perrig, A., Szewczyk, R., Culler, V.W.D., Tygar, J.D.: SPINS: Security protocols for sensor networks. In: Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM), pp. 189–199. IEEE, Rome (2001)
28. Poovendran, R., Wang, C., Roy, S.: Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks. Springer, Heidelberg (2007)
29. Przydatek, B., Song, D., Perrig, A.: SIA: Secure Information Aggregation in Sensor Networks. In: Proceedings of the First International Conference on Embedded Networked Sensor Systems, Los Angeles, California, USA, pp. 255–265 (2003)
30. Rajagopalan, R., Varshney, P.K.: Data-aggregation techniques in sensor networks: a survey. IEEE Communications Surveys & Tutorials 8(4), 48–63 (2006)
31. Szczechowiak, P., Oliveira, L., Scott, M., Collier, M., Dahab, R.: NanoECC: Testing the Limits of Elliptic Curve Cryptography in Sensor Networks. In: Verdone, R. (ed.) EWSN 2008. LNCS, vol. 4913, pp. 305–320. Springer, Heidelberg (2008)
32. Yang, Y., Wang, X., Zhu, S., Cao, G.: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks. In: Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 356–367 (2006)