# Constrained Function-Based Message Authentication for Sensor Networks

Chia-Mu Yu, *Student Member, IEEE*, Yao-Tung Tsou, Chun-Shien Lu, *Member, IEEE*, and Sy-Yen Kuo, *Fellow, IEEE*

***Abstract*—Sensor networks are vulnerable to false data injection attack and path-based denial of service (PDoS) attack. While conventional authentication schemes are insufficient for solving these security conflicts, an en-route filtering scheme, enabling each forwarding node to check the authenticity of the received message, acts as a defense against these two attacks. To construct an efficient en-route filtering scheme, this paper first presents a Constrained Function-based message Authentication (CFA) scheme, which can be thought of as a hash function directly supporting the en-route filtering functionality. Obviously, the crux of the scheme lies on the design of guaranteeing each sensor to have en-route filtering capability. Together with the redundancy property of sensor networks, which means that an event can be simultaneously observed by multiple sensor nodes, the devised CFA scheme is used to construct a CFA-based en-route filtering (CFAEF) scheme. In addition to the resilience against false data injection and PDoS attacks, CFAEF is inherently resilient against false endorsement-based DoS attack. In contrast to most of the existing methods, which rely on complicated security associations among sensor nodes, our design, which directly exploits an en-route filtering hash function, appears to be novel. We examine the CFA and CFAEF schemes from both the theoretical and numerical aspects to demonstrate their efficiency and effectiveness. Moreover, prototype implementation on TelosB mote demonstrates the practicality of our proposed method.**

***Index Terms*—Authentication, en-route filtering, security, sensor networks.**

## I. Introduction

A WIRELESS sensor network (WSN) is composed of a large number of sensor nodes with limited resources. Since WSNs can be deployed in an unattended or hostile environment, the design of an efficient authentication scheme is of great importance to the data authenticity and integrity in WSNs. In this respect, many authentication schemes have been proposed. The most straightforward way to guarantee data authenticity is to use conventional public-key cryptography-based digital signature techniques. Although the use of public-key cryptography on WSNs has been demonstrated in [30] and [33] to be feasible, the computation overhead is still rather high for resource-constrained devices.

*Authentication Problem:* Sensor networks are vulnerable to *false data injection attack* [46], by which the adversary injects false data, attempting to either deceive the base station (BS, or data sink), and *path-based denial of service* (PDoS) attack [13], by which the adversary sends bogus messages to randomly selected nodes so as to waste the energy of forwarding nodes.[1] Several so-called *en-route filtering schemes* have been proposed to quickly discover and remove the bogus event report injected by the adversary. Here, "en-route filtering" means that not only the destination node but also the intermediate nodes can check the authenticity of the message in order to reduce the number of hops the bogus message travels and, thereby, conserve energy. Hence, it is especially useful in mitigating false data injection attack and PDoS attack [13], because the falsified messages will be filtered out as soon as possible.

*Related Work:* SEF [49] is the first en-route filtering scheme found in the literature that exploits probabilistic key sharing over a partitioned key pool. Due to its design strategy, however, only a few intermediate nodes between the source-destination node pair have the ability to check the validity of forwarding messages, leading to low filtering capability. IHA [54], which verifies the transmitted packets in a deterministic hop-by-hop fashion, has also been proposed to authenticate the event report. It, however, requires complicated key sharing among neighboring nodes and could be vulnerable to node compromises if node compromises are mounted immediately after sensor deployment. Based on the similar idea used in SEF and IHA, several other en-route filtering schemes are proposed. With the sophisticated use of one-way hash chains in clustered sensor networks, DEF [43] has improved filtering power over SEF [49]. Using the proposed multiple-axis technique, GREF [45] is designed to support en-route filtering in the networks with multiple data sinks. LBRS [50], LTE [52], and LEDS [35] take advantage of location information to enhance the resilience to node compromises. CCEF [44], STEF [22], and KAEF [51] are presented to authenticate the transmitted packets only in

[1]The terms "forwarding node" and "intermediate node" are used interchangeably in this paper.

Manuscript received February 22, 2010; revised November 17, 2010; accepted December 29, 2010. Date of publication January 13, 2011; date of current version May 18, 2011. This paper was published, in part, in the *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2009 and the *Proceedings of The ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2009. The work of C.-M. Yu and C.-S. Lu were supported by NSC 97-2221-E-001-008 and NSC 98-2221-E-001-004-MY3. The work of S.-Y. Kuo was supported by NSC 96-2628-E-002-138-MY3. This work was supported by the National Science Council, Taiwan, under Grant NSC 97-2221-E-002-216-MY3 and by the Excellent Research Projects of National Taiwan University under the Center for Quantum Science and Engineering (97R0066-65), and Quantum Computing, Quantum Information/Communication (97R0066-67). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Robert H. Deng.

C.-M. Yu and Y.-T. Tsou are with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, and also with the Institute of Information Science, Academia Sinica, Taipei, Taiwan (e-mail: r91045@csie.ntu.edu.tw; yaodong@iis.sinica.edu.tw).

C.-S. Lu is with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan (e-mail: lcs@iis.sinica.edu.tw).

S.-Y. Kuo is with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: sykuo@cc.ee.ntu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2011.2106120

query-based sensor networks. In addition, exploiting the notion of perturbation polynomial, Zhang *et al.* [55] also proposed an en-route filtering scheme. For the seeming similarity between our proposed scheme and the scheme presented in [55], their difference and the weakness of Zhang *et al.*'s scheme will be described in Section III-B in more detail.

Note that, as to broadcast authentication, $\mu$TESLA and its variants [29], [34] can also serve message authentication well. Nevertheless, broadcast authentication is used to authenticate only the messages sent from the base station while en-route filtering schemes are used for authenticating and filtering a bogus event report that is assumed to not be detected by multiple legitimate sensor nodes in a node-to-node or node-to-BS communication pattern. Thus, the design of broadcast authentication schemes is orthogonal to the content of this paper.

In addition, preauthentication filters [14] also are developed to serve message authentication. Nonetheless, the only similarity between the preauthentication filters and en-route filtering schemes comes from the high level idea that each sensor can verify the authenticity of the received messages. Their difference is mainly due to the fact that the directions of data flow in our paper and [14] are not the same. In particular, in [14], each sensor should authenticate the data that is always sent by the base station. Nevertheless, the issue we address is in the opposite direction to the above; i.e., the data each sensor needs to check could be sent from any sensor and the base station. This makes the big difference between our paper and [14] in terms of the design of algorithms.

In fact, various security issues in WSNs have been considered in the literature. For example, the issue of key establishment is considered in [7], [10], [11], [15], [26], [31], and [46]. From the estimation theory point of view, the sensor data cryptography in WSNs is investigated in [1]. The secure aggregation in WSNs is also studied in [8] and [17]. In this paper, we focus on the design of an en-route filtering scheme that can simultaneously defend against false data injection attack, PDoS attack, and false endorsement-based DoS (FEDoS) attack. For the other security issues, please refer to [16], [21], and [38] for a comprehensive overview.

*Design of En-Route Filtering Schemes:* The redundancy property, which means that an event can be simultaneously observed by multiple sensor nodes, can be used to design the en-route filtering schemes. Specifically, the general design framework is that the source node that senses an event and wants to send an event report to the destination node first collects the neighboring nodes' endorsements of the sensed event. Afterwards, it sends out the event report and endorsements. Each intermediate node and the destination node can check the authenticity of the received report via the verification of the endorsements.

Aiming to enhance the filtering capability and improve the resilience against node compromises, most of the existing en-route filtering schemes rely on complicated security associations (e.g., key sharing), and, therefore, incur some assumptions such as secure bootstrapping time, stable routing, single data sink, the immobility of sensor nodes, etc., making them impractical. We identify the following four problems associated with the existing schemes.

1) The reason the unnecessary assumptions should be made stems from the fact that the message authentication codes (MACs, or keyed hash functions) used do not support en-route filtering functionality, while the authenticity of the forwarding messages needs to be checked by as many intermediate nodes as possible.
2) It has been demonstrated in [9], [12], and [37] that the node is able to send an event report to the other nodes in certain in-network control scenarios. Nonetheless, the existing schemes, which are only effective on the node-to-BS communication pattern, are ineffective in handling false data injection and PDoS attacks in such scenarios.
3) The existing en-route filtering schemes are difficult to apply on mobile sensor networks or networks with multiple sinks. In other words, the applicability of en-route filtering schemes on different network settings should be improved.
4) Last, based on conventional design, all the en-route filtering schemes suffer from a special kind of DoS attack, FEDoS attack [23], which could neutralize the advantages gained from the use of en-route filtering schemes. Although directly integrating the defense against FEDoS attack in [23] with the existing en-route filtering schemes is feasible, the assumption used in [23] has also to be made by the existing en-route filtering schemes and the overhead will be increased. Thus, it is more desirable to have an en-route filtering scheme that is inherently resilient against FEDoS attack.

In this paper, we take a completely different approach to the design of an en-route filtering scheme to avoid the above problems. In particular, instead of establishing security associations, we turn to construct an en-route filtering hash function, Constrained Function-based Authentication (CFA) scheme, and then employ such hash function to generate MACs used to endorse the sensor readings so that each intermediate node can verify the authenticity of forwarding messages. In particular, our proposed CFA possesses the following four characteristics: 1) *Resilience to node compromise* (RNC), which means that the compromised nodes cannot forge the messages sent from the genuine nodes; 2) *immediate authentication* (IA), which can be thought of as a synonym to en-route filtering and can be used to filter out the falsified messages as soon as possible to conserve energy; 3) *independence of network setting* (INS), which means that CFA can be applied to the networks with different network settings; 4) *efficiency* (EFF), which means that CFA has low computational and communication overhead. With these characteristics, a CFA-based en-route filtering (CFAEF) scheme can be constructed in such a way that the source node sends to the destination node a message, together with the corresponding CFA-based endorsements generated by the neighboring nodes. Afterwards, the source node can determine if the neighboring nodes send the false endorsement and each intermediate node has the ability to check the authenticity of forwarding messages. As a whole, as we will show later, the advantages of applying CFA on MAC generation are that the filtering capability can be improved, the resilience against FEDoS attack can be achieved, and the impractical assumptions previously made in the literature are no longer required.

*Our Contributions:* Our contributions are as follows:

1) A CFA scheme for WSNs is proposed. CFA can be thought of as a hash function directly supporting en-route filtering functionality, and can act as a building block for other security mechanisms.
2) A CFAEF scheme that can simultaneously defend against false data injection, PDoS, and FEDoS attacks is proposed. Particularly, compared with the existing methods, which either have low filtering capability or necessitate some unrealistic assumptions, our CFAEF scheme can be applied to arbitrary networks without further assumptions.
3) The efficiency of CFA and CFAEF schemes is studied in both theoretical and numerical aspects. Furthermore, prototype on TelosB mote demonstrate the practicality of our proposed method.

## II. System Model

*Network Model:* We assume a WSN composed of $N$ resource-limited sensor nodes with IDs, $\mathcal{I} \subset \mathbb{N}$. The unique ID for each node can be either arbitrarily assigned in the sensor platform, such as telosB, or fixed in a specific sensing hardware when manufactured, like the MAC address on current network interface cards (NICs). Although one or multiple base stations (or data sinks) are involved in data collection in a WSN, the efficiency of our proposed schemes does not rely on their trustworthiness and authenticity. In addition, arbitrary network topology is allowed in our method. Some or all of the sensor nodes can have mobility. The network planner, prior to sensor deployment, also cannot gain any deployment knowledge pertaining to sensors' locations.

*Security Model:* The objectives of the adversary are to deceive the BS into accepting the falsified event report and to deplete sensor nodes' energy by launching PDoS attack and FEDoS attack. In this paper, sensor nodes are assumed to not be equipped with tamper-resistant hardware. Thus, all the information is exposed and can be utilized by the adversary as long as a node is captured. We also assume that the attacks such as node compromises can be mounted by the adversary immediately after sensor deployment, i.e., the proposed schemes cannot rely on the secure bootstrapping time used in [35] and [53]. In particular, four specific attacks are considered in this paper:

1) *Eavesdropping attack*—The adversary continuously eavesdrops on the communication of the whole network, attempting to enhance its capability to send the false report without being detected. If needed, the adversary may receive a message from one node and then resend a modified message to another node.
2) *Node capture attack*—The adversary eavesdrops on the communication of the whole network and compromises certain sensor nodes. Taking advantage of the overheard information and the information gained from the compromised sensor nodes, the adversary attempts to recover the coefficients of the polynomial $f(x, y, z, w)$ (described later) used in the en-route filtering.
3) *Reflection attack* [43]—Exploiting the techniques borrowed from error correction codes, the adversary attempts to recover the coefficients of the polynomial $f(x, y, z, w)$ used in the en-route filtering. This attack is similar to the node capture attack and is specific to our proposed method.
4) *Deperutrbation attack* [2]—Exploiting certain algebraic operations, the adversary attempts to recover the coefficients of the polynomial $f(x, y, z, w)$ (described later) used in the en-route filtering. This attack is similar to the node capture attack and is specific to our proposed method.

If required, any pair of sensor nodes can establish their shared key[2] in a noninteractive fashion [46]. Although sensor networks are known to be vulnerable to many attacks such as wormhole attack, selective forwarding attack, etc., we refer to the existing rich literature [4], [19], [25], [42] for these issues and the defense against these attacks is beyond the scope of this paper.

TABLE I
Notation Table

| | Description |
|---|---|
| $N$ | the number of sensor nodes |
| $\mathcal{I}$ | the set of node IDs |
| $K_{u,v}$ $(= K_{v,u})$ | the key shared between nodes $u$ and $v$ |
| $f(x, y, z, w)$ | the legitimate secret polynomial |
| $d$ | the degree of each variable in $f(x, y, z, w)$ |
| $\mathfrak{F}$ | the constrained function set |
| $n_{u,\mathfrak{a}}(y, z, w)$ | the additional perturbation polynomial applied on $f(u, y, z, w)$ |
| $n_{u,\mathfrak{v}}(x, z, w)$ | the additional perturbation polynomial applied on $f(x, u, z, w)$ |
| $auth_u(y, z, w)$ | the authentication polynomial stored in node $u$ |
| $verf_u(x, z, w)$ | the verification polynomial stored in node $u$ |
| $\mathfrak{N}_\mathfrak{a}$ | the authentication perturbation set |
| $\mathfrak{N}_\mathfrak{v}$ | the verification perturbation set |
| $MAC_u(v, m)$ | the message authentication code constructed by node $u$ with the purpose of guaranteeing the authenticity of message $m$ to be received by $v$ |
| $\mathfrak{F}'$ | the weak constrained function set |
| $[\alpha]$ | an instance of the coefficients, $\alpha_{i,j,k,m}$'s, of the polynomial $\sum_{i,j,k,m} \alpha_{i,j,k,m} x^i y^j z^k w^m$ |
| $t$ | the number of endorsements needed to be collected by the source node |

## III. Constrained Function-Based Authentication (CFA) Scheme

Since the proposed CFA scheme is constructed by making use of the pairwise key generated by the CARPY+ scheme [46] for secure communication, we first briefly review CARPY+ in Section III-A to make this paper self-contained. Then, the proposed CFA scheme will be presented in the remaining subsections. In this paper, nodes $u$, $v$, and $\varepsilon$ are denoted as the source node, destination node, and intermediate node, respectively. In addition, $K_{u,v} = K_{v,u}$ is used to represent the pairwise key between the nodes $u$ and $v$. The notations frequently used in the paper are summarized in Table I.

[2]Here, the key establishment scheme in [46], instead of the ones in [7], [10], [11], [15], [26], and [31], is chosen to be used in our proposed method because the latter are *interactive*, which means that two nodes are required to communicate with each other once they would like to establish their common key. In fact, when the mobile sink with localization capability is available, the other key establishment schemes can also be used because, with the aid of the mobile sink, each node can only share keys with its upstream and downstream nodes, as in [35] and [50]. For ease of explanation, we omit the details in this case.

### A. Review of the CARPY+ Scheme [46]

Let $N$, $\lambda$, and $\mathbb{F}_q = \{0, \ldots, q-1\}$, where $q$ is a prime number, be the number of sensor nodes, a security parameter independent of $N$, and a finite field, respectively. Let $A = (D \cdot G)^T$, where $D \in \mathbb{F}_q^{(\lambda+1)\times(\lambda+1)}$ is a symmetric matrix, $G \in \mathbb{F}_q^{(\lambda+1)\times N}$ is a matrix, and $(D \cdot G)^T$ is the transpose of $(D \cdot G)$. Let $K = A \cdot G$. It can be shown that $K$ must be symmetric because $A \cdot G = (D \cdot G)^T \cdot G = G^T \cdot D \cdot G = (A \cdot G)^T$. Before sensor deployment, proper constrained random perturbation vectors are selected and applied on each row vector of $A$ to construct a matrix $W$. In addition, $G$ is selected as a Vandermonde matrix generated by a seed. The $j$th row vector of $W$, $W_{j,-}$, is stored into the node $j$. After sensor deployment, node $u$ can have the shared key with node $v$ by calculating the inner product of the row vector $W_{u,-}$ and the $v$th column vector $G_{-,v}$, then extracting the common part as the shared key. Note that in the CARPY+ scheme, $G$ and $s$ can be publicly known while $A$ should be kept secret. Therefore, CARPY+ can establish a pairwise key between each pair of sensor nodes without needing any communication. This property is an essential part in constructing the proposed CFA scheme, because establishing a key via communications incurs the authentication problem, leading to a circular dependency.

### B. Basic Idea

In the proposed CFA scheme, the network planner, before sensor deployment, selects a secret polynomial $f(x, y, z, w)$ from the set $\mathfrak{F}$ (to be defined in (1) later), whose coefficients should be kept as secret, thereby constituting the security basis of CFA. For simplicity, we assume that the degree of each variable in $f(x, y, z, w)$ is the same, which is $d$, although they can be distinct in our scheme. For each node $u$, the network planner constructs two polynomials, $f_{u,1}(y, z, w) = f(u, y, z, w)$ and $f_{u,2}(x, z, w) = f(x, u, z, w)$. Since directly storing these two polynomials enables the adversary to obtain the coefficients of $f(x, y, z, w)$ by capturing a few nodes, the authentication polynomial $\mathrm{auth}_u(y, z, w)$ and verification polynomial $\mathrm{verf}_u(x, z, w)$ should be, respectively, constructed from the polynomials $f_{u,1}(y, z, w)$ and $f_{u,2}(x, z, w)$ by adding independent perturbation polynomials. Afterwards, the authentication and verification polynomials, instead of $f_{u,1}(y, z, w)$ and $f_{u,2}(x, z, w)$, are stored in node $u$. For source node $u$, the MAC attached to the message $m$ is calculated according to its own authentication polynomial. Let *verification number* be the result calculated from the verification polynomial $\mathrm{verf}_u(x, z, w)$ by substituting the claimed source node ID, the shared pairwise key, and the hashed message into $x$, $z$, and $w$, respectively. The received node considers the received message authentic and intact if and only if the *verification difference*, which is the difference between the received MAC and its calculated verification number, is within a certain predetermined range.

Although our CFA scheme is similar to Zhang *et al.*'s scheme [55], the design strategies used in these two schemes are different, except the fact that both rely on polynomial evaluation. In Zhang *et al.*'s scheme, due to the improper use of perturbation, the nodes' IDs should be artificially assigned, resulting in the limitation of hardware dependence. In addition, as an arbitrary secret polynomial can be used in [55], immediate authentication can be achieved only if the message authentication code forms a polynomial. The worst is that, because the use of perturbation is only conducted on a few coefficients in the secret polynomial in Zhang *et al.*'s scheme, a method very recently proposed in [2] can be utilized to estimate almost all the coefficients in the secret polynomial and forge the MAC.

On the contrary, since the secret polynomial $f(x, y, z, w)$ in CFA is selected such that certain properties are satisfied, the message authentication code can be reduced from a polynomial size to a single number, resulting in less communication overhead (packet overhead). On the other hand, whereas the pairwise key has been considered useless in providing either immediate authentication or resilience to node compromises in previous methods, in this paper we find that the pairwise key is helpful in enhancing the security while retaining the property of immediate authentication. In addition, motivated by the idea of defending against *protocol attacks* in the watermarking community [6], [24], [32], which incorporates the cover signal into the construction of content-dependent watermarks, our proposed method possessing the similar characteristic is resilient against the attack in [2]. Hence, all these characteristics substantially differentiate CFA from [55].

In the following two subsections, the off-line step and on-line step, respectively, will be described.

### C. Off-Line Step of CFA Scheme

Before deploying sensor nodes, the network planner picks a parameter $q$ from which a finite field $\mathbb{F}_q$ is built. All of the operations throughout the paper are performed over $\mathbb{F}_q$ unless specifically mentioned. Let $\mathcal{I}$ be the set of node IDs. Let $\ell$ be the least number of bits sufficient to represent $q$. Assume that node IDs, pairwise key, and hash value can be represented in $\mathbb{F}_q$. In addition, a security parameter $r < \ell$ is also selected. Then, the secret polynomials $f(x, y, z, w)$'s, used as the basis for constructing both authentication and verification polynomials, are defined in *constrained function set*, $\mathfrak{F}$, where

$$\begin{aligned}\mathfrak{F} = \big\{ f(x,y,z,w) \,|\, & |f(x,y,z,w) - f(x,y',z',w)| \le 2^{r-1}, \\ & |f(x,y,z,w) - f(x',y',z',w)| \ge 3\cdot 2^{r-1} - 1, \\ & |f(x,y,z,w) - f(x',y',z',w')| \ge 3\cdot 2^{r-1} - 1, \\ & x,y \in \mathcal{I}, x' \ne x, y' \ne y, z' \ne z, w' \ne w, r < \ell \big\}. \quad (1)\end{aligned}$$

The authentication polynomial, $\mathrm{auth}_u(y, z, w) = f(u, y, z, w) + n_{u,\mathfrak{a}}(y, z, w)$, and verification polynomial, $\mathrm{verf}_u(x, z, w) = f(x, u, z, w) + n_{u,\mathfrak{v}}(x, z, w)$, are stored in each node $u$, where polynomials $n_{u,\mathfrak{a}}(y, z, w)$ and $n_{u,\mathfrak{v}}(x, z, w)$, used for perturbation, are randomly selected from the *authentication perturbation set*, $\mathfrak{N}_\mathfrak{a} = \{n(y, z, w) | 0 \le n(y, z, w) \le 2^{r-2} - 1, y \in \mathcal{I}, 0 \le z, w \le q-1\}$, and the *verification perturbation set*, $\mathfrak{N}_\mathfrak{v} = \{n(x, z, w) | 0 \le n(x, z, w) \le 2^{r-1} - 1, x \in \mathcal{I}, 0 \le z, w \le q-1\}$, respectively. Though the sets $\mathfrak{F}$, $\mathfrak{N}_\mathfrak{a}$, and $\mathfrak{N}_\mathfrak{v}$ appear to be artificial, they guarantee the efficiency and feasibility of immediate authentication of CFA. In addition, constructing $\mathrm{auth}_u(y, z, w)$ and $\mathrm{verf}_u(x, z, w)$

**Algorithm:** CFA-Off-line-Step($q$, $r$)

1. Randomly picks a secret polynomial $f(x, y, z, w) \in \mathfrak{F}$
2. **for** each node $u$
3. Randomly picks $n_{u,\mathfrak{a}}(y, z, w) \in \mathfrak{N}_{\mathfrak{a}}$ and $n_{u,\mathfrak{v}}(x, z, w) \in \mathfrak{N}_{\mathfrak{v}}$
4. Store $auth_u(y, z, w) := f(u, y, z, w) + n_{u,\mathfrak{a}}(y, z, w)$
5. Store $verf_u(x, z, w) := f(x, u, z, w) + n_{u,\mathfrak{v}}(x, z, w)$

Fig. 1. Off-line step of CFA.

from $\mathfrak{F}$, $\mathfrak{N}_{\mathfrak{a}}$, and $\mathfrak{N}_{\mathfrak{v}}$ may be time- and energy-consuming. It, however, could be acceptable because such construction is performed only by the network planner, instead of sensor nodes. If the time required for constructing $\mathrm{auth}_u(y, z, w)$ and $\mathrm{verf}_u(x, z, w)$ is still an issue that cannot be ignored, an efficient method for constructing the polynomials in a restricted version of $\mathfrak{F}$ will be later discussed in Section III-E. The off-line procedure of CFA is described in Fig. 1.

### D. On-Line Step of CFA Scheme

After sensor deployment, the sensor node may work as a source node, intermediate node, or destination node depending on whether the message is to be sent or verified. In the following, we describe the operations one should perform when the node acts as different roles. It should be noted that the pairwise key $K_{u,v} = K_{v,u}$, used here, is constructed by applying the CARPY+ scheme [46] on nodes $u$ and $v$, respectively.

*Source Node (Message Transmission):* When node $u$ wants to send a message $m$ to node $v$, it calculates the message authentication code:

$$\mathrm{MAC}_u(v, m) = \mathrm{auth}_u\left(v, K_{u,v}, h(m)\right) + n_{u,s}$$

where $n_{u,s}$ is randomly picked from the set $\{0, \ldots, 2^{r-2}\}$. Then, the packet $\mathcal{M} = \langle u, v, m, \mathrm{MAC}_u(v, m)\rangle$ is sent to $v$ possibly through a multihop path. Note that the message authentication code $\mathrm{MAC}_u(v, m)$ is only a number here.

*Destination Node (Message Verification):* After receiving the packet $\mathcal{M} = \langle u, v, m, \mathrm{MAC}_u(v, m)\rangle$, the destination node $v$ first calculates the verification number:

$$\mathrm{verf}_v\left(u, K_{v,u}, h(m)\right)$$

according to its own verification polynomial $\mathrm{verf}_v(x, z, w)$ and then calculates the corresponding *verification difference*, $\mathrm{VD}_{v,u}$:

$$\mathrm{VD}_{v,u} = \left|\mathrm{verf}_v\left(u, K_{v,u}, h(m)\right) - \mathrm{MAC}_u(v, m)\right|.$$

If $\mathrm{VD}_{v,u}$ is within the range $\{0, \ldots, 2^{r-1} - 1\}$, where $r$ is a security parameter mentioned in Section III-C, then the authenticity and integrity of the packet $\mathcal{M}$ is successfully verified. Otherwise, the packet $\mathcal{M}$ is dropped. The principle behind this step is as follows:

$$\begin{aligned}
&\mathrm{verf}_v\left(u, K_{v,u}, h(m)\right) - \mathrm{MAC}_u(v, m)\\
&\quad = \left(f\left(u, v, K_{v,u}, h(m)\right) + n_{v,\mathfrak{v}}\left(u, K_{v,u}, h(m)\right)\right)\\
&\qquad - \left(f\left(u, v, K_{u,v}, h(m)\right) + n_{u,\mathfrak{a}}\left(v, K_{u,v}, h(m)\right) + n_{u,s}\right)\\
&\quad = \left(f\left(u, v, K_{v,u}, h(m)\right) - f\left(u, v, K_{u,v}, h(m)\right)\right)\\
&\qquad + \left(n_{v,\mathfrak{v}}\left(u, K_{v,u}, h(m)\right) - \left(n_{u,\mathfrak{a}}\left(v, K_{u,v}, h(m)\right) + n_{u,s}\right)\right)\\
&\quad = n_{v,\mathfrak{v}}\left(u, K_{v,u}, h(m)\right)\\
&\qquad - \left(n_{u,\mathfrak{a}}\left(v, K_{u,v}, h(m)\right) + n_{u,s}\right). \qquad (2)
\end{aligned}$$

From the rules of constructing authentication and verification polynomials, we know that $n_{\varepsilon,\mathfrak{v}}(u, K_{\varepsilon,u}, h(m)) \in \{0, \ldots, 2^{r-1} - 1\}$, $n_{u,\mathfrak{a}}(v, K_{u,v}, h(m)) \in \{0, \ldots, 2^{r-2} - 1\}$, and $n_{u,s} \in \{0, \ldots, 2^{r-2}\}$. Thus, when $\mathcal{M}$ is genuine, the verification difference $\mathrm{VD}_{v,u} = |\mathrm{verf}_v(u, K_{v,u}, h(m)) - \mathrm{MAC}_u(v, m)|$ must be within $\{0, \ldots, 2^{r-1} - 1\}$. In other words, when the $\mathrm{MAC}_{u'}(v, m')$, for some $u' \neq u$ and $m' \neq m$, is randomly generated by the adversary attempting to claim that $\mathrm{MAC}_{u'}(v, m')$ is sent from $u'$ or constructed from $m'$, the probability that such a falsified MAC successfully passes the verification is $2^r - 1/q$, and, therefore, the probability of detecting such a falsified MAC is $1 - (2^r - 1/q)$.

*Intermediate Node (Message Verification):* After receiving the packet $\mathcal{M} = \langle u, v, m, \mathrm{MAC}_u(v, m)\rangle$, the intermediate node $\varepsilon$ first calculates $\mathrm{verf}_\varepsilon(u, K_{\varepsilon,u}, h(m))$ according to its own verification polynomial $\mathrm{verf}_\varepsilon(x, z, w)$ and then calculates the verification difference $\mathrm{VD}_{\varepsilon,u} = |\mathrm{verf}_\varepsilon(u, K_{\varepsilon,u}, h(m)) - \mathrm{MAC}_u(v, m)|$. If $\mathrm{VD}_{\varepsilon,u}$ is within the range $\{0, \ldots, 2^r - 1\}$, then the authenticity of the packet $\mathcal{M}$ is successfully verified, and the packet $\mathcal{M}$ will be forwarded by node $\varepsilon$. Otherwise, the packet $\mathcal{M}$ is dropped. The principle behind this step is as follows. When a genuine packet $\mathcal{M}$ is received, we can obtain:

$$\begin{aligned}
&\mathrm{verf}_\varepsilon\left(u, K_{\varepsilon,u}, h(m)\right) - \mathrm{MAC}_u(v, m)\\
&\quad = \left(f\left(u, \varepsilon, K_{\varepsilon,u}, h(m)\right) + n_{\varepsilon,\mathfrak{v}}\left(u, K_{\varepsilon,u}, h(m)\right)\right)\\
&\qquad - \left(f\left(u, v, K_{u,v}, h(m)\right) + n_{u,\mathfrak{a}}\left(v, K_{u,v}, h(m)\right) + n_{u,s}\right)\\
&\quad = \left(f\left(u, \varepsilon, K_{\varepsilon,u}, h(m)\right) - f\left(u, v, K_{u,v}, h(m)\right)\right)\\
&\qquad + \left(n_{\varepsilon,\mathfrak{v}}\left(u, K_{\varepsilon,u}, h(m)\right)\right.\\
&\qquad - \left(n_{u,\mathfrak{a}}\left(v, K_{u,v}, h(m)\right) + n_{u,s}\right). \qquad (3)
\end{aligned}$$

By the construction of $\mathfrak{F}$, we know:

$$\left|f\left(u, \varepsilon, K_{\varepsilon,u}, h(m)\right) - f\left(u, v, K_{u,v}, h(m)\right)\right| \leq 2^{r-1}. \qquad (4)$$

In addition, from the rules of constructing authentication and verification polynomials, we know that $n_{\varepsilon,\mathfrak{v}}(u, K_{\varepsilon,u}, h(m)) \in \{0, \ldots, 2^{r-1} - 1\}$, $n_{u,\mathfrak{a}}(v, K_{u,v}, h(m)) \in \{0, \ldots, 2^{r-2} - 1\}$, $n_{u,s} \in \{0, \ldots, 2^{r-2}\}$, and therefore $(n_{\varepsilon,\mathfrak{v}}(u, K_{\varepsilon,u}, h(m)) - (n_{u,\mathfrak{a}}(v, K_{u,v}, h(m)) + n_{u,s}) \in \{-2^{r-1} + 1, \ldots, 2^{r-1} - 1\}$. Hence, the verification difference $\mathrm{VD}_{\varepsilon,u}$ must be within $\{0, \ldots, 2^r - 1\}$.

On the other hand, consider the case where node $u$ has been compromised by the adversary. The adversary now wants to deceive $v$ that a message $m$ sent by $u$ is sent by $u' \neq u$. Consider for example the modified packet

$$\mathcal{M}' = \langle u', v, m, \mathrm{MAC}_u(v, m) \rangle \tag{5}$$

where $u'$ means a node ID the adversary pretends to be. Note that though the adversary can compromise multiple nodes, here we only consider the adversary who exploits the information obtained from a single captured node $u$, and focus on the use of the constructed set $\mathfrak{F}$. The verification procedure at the intermediate node $\varepsilon$ is as follows:

$$\begin{aligned}
&\mathrm{verf}_\varepsilon\left(u', K_{\varepsilon,u'}, h(m)\right) - \mathrm{MAC}_u(v, m) \\
&\quad = \left(f\left(u', \varepsilon, K_{\varepsilon,u'}, h(m)\right) + n_{\varepsilon,\mathfrak{v}}\left(u', K_{\varepsilon,u'}, h(m)\right)\right) \\
&\qquad - \left(f\left(u, v, K_{u,v}, h(m)\right) + n_{u,\mathfrak{a}}\left(v, K_{u,v}, h(m)\right) + n_{u,s}\right) \\
&\quad = \left(f\left(u', \varepsilon, K_{\varepsilon,u'}, h(m)\right) - f\left(u, v, K_{u,v}, h(m)\right)\right) \\
&\qquad + \left(n_{\varepsilon,\mathfrak{v}}\left(u', K_{\varepsilon,u'}, h(m)\right)\right. \\
&\qquad - \left.\left(n_{u,\mathfrak{a}}\left(v, K_{u,v}, h(m)\right) + n_{u,s}\right).\right.
\end{aligned} \tag{6}$$

By the construction of $\mathfrak{F}$, we know

$$\left|f\left(u', \varepsilon, K_{\varepsilon,u'}, h(m)\right) - f\left(u, v, K_{u,v}, h(m)\right)\right| \geq 3 \cdot 2^{r-1} - 1. \tag{7}$$

In addition, from the construction of authentication and verification polynomials, we know that $n_{\varepsilon,\mathfrak{v}}(u', K_{\varepsilon,u'}, h(m)) \in \{0, \ldots, 2^{r-1} - 1\}$, $n_{u,\mathfrak{a}}(v, K_{u',v}, h(m)) \in \{0, \ldots, 2^{r-2} - 1\}$, and $n_{u,s} \in \{0, \ldots, 2^{r-2}\}$. Therefore, the verification difference $\mathrm{VD}_{\varepsilon,u}$ must be not within $\{0, \ldots, 2^{r-1}\}$ and the packet $\mathcal{M}'$ will be dropped. In other words, when the adversary follows the procedures in CFA, once the source node ID of a message is modified, such malicious manipulation will be *deterministically* detected by the intermediate nodes. Definitely, the adversary can choose to not follow the procedures in CFA. If the falsified MAC in $\mathcal{M}'$ is randomly generated by the adversary, the probability that such a falsified MAC successfully passes the verification executed by the intermediate node is $2^{r+1} - 1/q$, and, therefore, the probability of detecting such a falsified MAC is $1 - (2^{r+1} - 1/q)$. The on-line procedure of CFA is described in Fig. 2.

**Algorithm:** CFA-On-line-Step

**Scenario:** node $u$ sends a message $m$ to node $v$

**Source node** $u$:

1. Calculate $K_{u,v}$ and $h(m)$
2. Compute $MAC_u(v, m) := auth_u(v, K_{u,v}, h(m)) + n_{u,s}$, where $n_{u,s}$ is randomly picked from $\{0, \ldots, 2^{r-2} - 1\}$
3. Send the packet $\mathcal{M} := \langle u, v, m, MAC_u(v, m) \rangle$

**Intermediate node** $\varepsilon$ **(on receiving** $\mathcal{M}$**):**

1. Calculate $K_{u,\varepsilon}$ and $h(m)$
2. Calculate $VD_{\varepsilon,u} := |verf_\varepsilon(u, K_{\varepsilon,u}, h(m)) - MAC_u(v, m)|$
3. **if** $VD_{\varepsilon,u} \in \{0, \ldots, 2^r - 1\}$ **then** forwarding $\mathcal{M}$ **else** drop $\mathcal{M}$

**Destination node** $v$ **(on receiving** $\mathcal{M}$**):**

1. Calculate $K_{u,v}$ and $h(m)$
2. Calculate $VD_{v,u} := |verf_v(u, K_{v,u}, h(m)) - MAC_u(v, m)|$
3. **if** $VD_{v,u} \in \{0, \ldots, 2^{r-1} - 1\}$ **then** accept $\mathcal{M}$ **else** drop $\mathcal{M}$

Fig. 2. On-line step of CFA.

### E. Implementation Issues

The effectiveness and efficiency of the proposed CFA scheme rely on the use of $\mathrm{auth}_u(y, z, w)$ and $\mathrm{verf}_u(x, z, w)$, which satisfy the constrained function set $\mathfrak{F}$, the authentication perturbation set $\mathfrak{N}_\mathfrak{a}$, and the verification perturbation set $\mathfrak{N}_\mathfrak{v}$. As the construction of $\mathfrak{N}_\mathfrak{a}$ and $\mathfrak{N}_\mathfrak{v}$ is relatively easy, in this section, we focus on the construction of $\mathrm{auth}_u(y, z, w)$ and $\mathrm{verf}_u(x, z, w)$, with particular emphasis on the construction of $f(x, y, z, w)$.

A straightforward method for deriving proper $f(x, y, z, w)$ is to construct the whole set $\mathfrak{F}$ and then randomly pick one from $\mathfrak{F}$. When the coefficients of the polynomials in $\mathfrak{F}$ are constrained with $\mathbb{F}_q$, there are $q^{(d+1)^4}$ possible four-variate $d$-degree polynomials. Thus, $O(q^{2\cdot(d+1)^4})$ tests are required because each of the $q^{(d+1)^4}$ four-variate $d$-degree polynomials needs to check whether it satisfies the constraints $|f(x, y, z, w) - f(x', y, z', w')| \geq 3 \cdot 2^{r-1} - 1$, $|f(x, y, z, w) - f(x', y', z', w)| \geq 3 \cdot 2^{r-1} - 1$, and $|f(x, y, z, w) - f(x, y', z', w)| \leq 2^{r-1}$ in $\mathfrak{F}$, by examining the other $q^{(d+1)^4} - 1$ possibilities of different input variables. The above construction of $\mathfrak{F}$ will be accomplished before sensor deployment by the network planner that is usually assumed to be resource-abundant. Despite its feasibility, such an exhaustive search-like method is not sufficiently efficient. In the following, we develop an efficient algorithm trading the deterministic security for the construction efficiency on the basis of the observation that, in some cases, a variant of $\mathfrak{F}$ is sufficient for our use and the search for a variant of $\mathfrak{F}$ can accelerate the construction of $f(x, y, z, w)$. Hence, we emphasize on how to efficiently construct a variant $\mathfrak{F}'$ of the original constrained function set $\mathfrak{F}$.

Let $\mathfrak{F}'$ be the *weak constrained function set* as follows:

$$\begin{aligned}
\mathfrak{F}' = \{ f(x, y, z, w) \,|\, &|f(x, y, z, w) - f(x, y', z', w)| \leq 2^{r-1}, \\
&x, y \in \mathcal{I}, x' \neq x, y' \neq y, z' \neq z, w' \neq w, r < \ell \}.
\end{aligned} \tag{8}$$

Obviously, $\mathfrak{F}$ is a subset of $\mathfrak{F}'$ since some constraints in $\mathfrak{F}$ are discarded. As to the construction of $f(x, y, z, w)$ in $\mathfrak{F}'$, our idea is to construct a random subset of $\mathfrak{F}'$ that is as large as possible.

**Algorithm:** $\mathfrak{F}'$-Construction ($[\alpha]$ `is the final output of this algorithm`)

1. randomly select $[\alpha]$
2. **while** $[\alpha]$ cannot satisfy Eq. (13)
3. $\quad [\alpha] := [\alpha/2]$
4. randomly construct $\Omega := \{(i,j,k,m) | 0 \le i,j,k,m \le d\}$ with $|\Omega| \ge 0$
5. **for** each element $(i,j,k,m)$ in $\Omega$
6. $\quad$ find the maximum $\varphi$ such that $\langle [\alpha], (i,j,k,m), \varphi \rangle$ is satisfied with Eq. (13)
7. $\quad$ set $[\alpha] := \langle [\alpha], (i,j,k,m), \pi \rangle$, where $\pi$ is randomly selected from $[0, \varphi]$
8. $[\alpha] := [\phi]$, where $[\phi]$ is randomly selected from $\{[\alpha'] | [\alpha'] \preceq [\alpha]\}$

Fig. 3. $\mathfrak{F}'$-construction algorithm.

After that, the polynomials used in CFA are sampled from the constructed subset of $\mathfrak{F}'$. Assume that $x \in [x_{\min}, x_{\max}], y \in [y_{\min}, y_{\max}], z \in [z_{\min}, z_{\max}]$, and $w \in [w_{\min}, w_{\max}]$. The property useful in constructing a polynomial $f(x,y,z,w)$ satisfying the constraints in $\mathfrak{F}'$ will be shown as follows.

Assume that $f(x,y,z,w) = \sum_{i,j,k,m=0}^{d} \alpha_{i,j,k,m} x^i y^j z^k w^m$, $d \in \mathbb{Z}_+$, $\alpha_{i,j,k,m} \in \mathbb{F}_q$. The polynomial $f(x,y,z,w)$ can be rewritten as:

$$\sum_{i,m=0}^{d} \alpha_{i,0,0,m} x^i w^m + \sum_{i,m=0,j,k=1}^{d} \alpha_{i,j,k,m} x^i y^j z^k w^m. \quad (9)$$

With the representation in (9), the term $f(x,y,z,w) - f(x,y',z',w)$ can be written as:

$$\begin{aligned} &\sum_{i,m=0}^{d} \alpha_{i,0,0,m} x^i w^m + \sum_{i,m=0,j,k=1}^{d} \alpha_{i,j,k,m} x^i y^j z^k w^m \\ &\quad - \left( \sum_{i,m=0}^{d} \alpha_{i,0,0,m} x^i w^m \right. \\ &\qquad \left. + \sum_{i,m=0,j,k=1}^{d} \alpha_{i,j,k,m} x^i (y')^j (z')^k w^m \right) \\ &= \sum_{i,m=0,j,k=1}^{d} \alpha_{i,j,k,m} x^i w^m \left( y^j z^k - (y')^j (z')^k \right). \quad (10) \end{aligned}$$

By taking the constraint $|f(x,y,z,w) - f(x,y',z',w)| \le 2^{r-1}$ in $\mathfrak{F}'$ into consideration, we have

$$\begin{aligned} -2^{r-1} \le &\sum_{i,m=0,j,k=1}^{d} \left( \alpha_{i,j,k,m} x^i w^m \left( y^j z^k - (y')^j (z')^k \right) \right) \\ &\le 2^{r-1}. \end{aligned} \quad (11)$$

With $-2^{r-1}$ and $2^{r-1}$ being the lower bound and upper bound of $f(x,y,z,w) - f(x,y',z',w)$, respectively, (11) can be rewritten as: We can examine if a given set of $\alpha_{i,j,k,m}$'s, $\forall i,j,k,m$, constitutes a polynomial $f(x,y,z,w)$ of $\mathfrak{F}'$ by exploiting the definitions in (8) and considering the extremes in (12)

$$\begin{cases} \max \left\{ \sum_{i,m=0,j,k=1}^{d} \alpha_{i,j,k,m} x^i w^m \right. \\ \qquad \left. \times \left( y^j z^k - (y')^j (z')^k \right) \right\} \le 2^{r-1} \\ \min \left\{ \sum_{i,m=0,j,k=1}^{d} \alpha_{i,j,k,m} x^i w^m \right. \\ \qquad \left. \times \left( y^j z^k - (y')^j (z')^k \right) \right\} \ge -2^{r-1} \end{cases} \quad (12)$$

shown in (13)

$$\begin{cases} \sum_{i,m=0,j,k=1}^{d} \alpha_{i,j,k,m} x_{\max}^i w_{\max}^m \times \\ \qquad \left( y_{\max}^j z_{\max}^k - y_{\min}^j z_{\min}^k \right) \le 2^{r-1} \\ \sum_{i,m=0,j,k=1}^{d} \alpha_{i,j,k,m} x_{\max}^i w_{\max}^m \times \\ \qquad \left( y_{\min}^j z_{\min}^k - y_{\max}^j z_{\max}^k \right) \ge -2^{r-1}. \end{cases} \quad (13)$$

Define $f'(x,y,z,w) = \sum_{i,j,k,m=0}^{d} \alpha'_{i,j,k,m} x^i y^j z^k w^m$, which is only different from $f(x,y,z,w)$ in the part of coefficients. From (13), we can observe that the possible range of $|f'(x,y,z,w) - f'(x,y',z',w)|$ will be contained in $|f(x,y,z,w) - f(x,y',z',w)|$, i.e., $\max\{f'(x,y,z,w) - f'(x,y',z',w)\} \le \max\{f(x,y,z,w) - f(x,y',z',w)\}$ and $\min\{f'(x,y,z,w) - f'(x,y',z',w)\} \ge \min\{f(x,y,z,w) - f(x,y',z',w)\}$, if $\alpha_{i,j,k,m} - \alpha'_{i,j,k,m} \ge 0$, $\forall i,\ j,\ k,\ m$. With this *monotone* property, our algorithm for randomly sampling a polynomial from a random subset of $\mathfrak{F}'$, whose pseudocode is shown in Fig. 3, can be described as follows.

As $\alpha_{i,j,k,m}$ in $f(x,y,z,w)$ denotes the coefficient of $x^i y^j z^k w^m$ for specified $i,j,k,m$, we use $[\alpha]$ to denote an instance of $\alpha_{i,j,k,m}$'s, $\forall i,j,k,m$. At the beginning of $\mathfrak{F}'$-Construction algorithm shown in Fig. 3, we randomly choose $[\alpha]$ and determine if the chosen $[\alpha]$ satisfies (13). If $[\alpha]$ fails to satisfy (13), $[\alpha] := [\alpha/2]$ is checked recursively until (13) is satisfied (Lines 1 $\sim$ 3). Here, $[\alpha/2]$ consists of $\lfloor (\alpha_{i,j,k,m}/2) \rfloor$'s, where each $\alpha_{i,j,k,m}$ is an element in $[\alpha]$. Note that the loop (Lines 2 $\sim$ 3) is guaranteed to terminate within at most $\log q$ steps because at least the setting of $\alpha_{i,j,k,m} = 0$, $\forall i,j,k,m$,

is satisfiable. With the monotone property, we can also guarantee that any polynomial sampling from $\{[\alpha']|[\alpha'] \preceq [\alpha]\}$ is one of the polynomials in $\mathfrak{F}'$. Here, $[\alpha'] \preceq [\alpha]$ means that the possible range of $|f'(x,y,z,w) - f'(x,y',z',w)|$ will be contained in $|f(x,y,z,w) - f(x,y',z',w)|$. Thus, after the execution of the loop containing Lines 2 $\sim$ 3, we can sample a polynomial $f(x,y,z,w) \in \mathfrak{F}'$ from the sample space $\{[\alpha']|[\alpha'] \preceq [\alpha]\}$. Nevertheless, we can, in fact, further extend the sample space by tuning selected $\alpha_{i,j,k,m}$'s (Lines 5 $\sim$ 7). For example, suppose $|\Omega|\alpha_{i,j,k,m}$'s are chosen to be tuned. In particular, defining $\langle[\alpha],(i,j,k,m),\varphi\rangle$ as $[\alpha]$ whose $\alpha_{i,j,k,m}$ is selected to be replaced by $\varphi$, we can extend the range of $|f'(x,y,z,w) - f'(x,y',z',w)|$ by maximizing the selected $\alpha_{i,j,k,m}$ so that the size of $\{[\alpha']|[\alpha'] \preceq [\alpha]\}$ will be increased. Together with $[\alpha]$ obtained after the loop in Lines 5 $\sim$ 7, Line 8 behaves like sampling a polynomial from a subset of $\mathfrak{F}'$, which could be randomly different due to the random construction of $\Omega$ (Line 4). Note that a search of maximum $\varphi$ (Line 6) can be accomplished by conducting binary search on the positive integers greater than $\alpha_{i,j,k,m}$. Since we should conduct binary search once for each element in $\Omega$, the running time of $\mathfrak{F}'$-Construction algorithm is $O(|\Omega|\log q)$. Indeed, from the theoretical point of view, it might obtain only a useless constant polynomial after the execution of $\mathfrak{F}'$-Construction algorithm, therefore, require executing the algorithm multiple times. Nevertheless, in practice, when a sufficiently large security parameter $r$ [as defined in (1) and (8)] is selected, executing the algorithm once is sufficient for sampling a nontrivial polynomial from $\mathfrak{F}'$. In fact, our implementation of the $\mathfrak{F}'$-Construction algorithm on MATLAB finishes the job of sampling a nonconstant polynomial from $\mathfrak{F}'$ within minutes. It should be noted that $\mathfrak{F}'$-Construction algorithm is not a uniform sampling over $\mathfrak{F}'$. As we mentioned earlier, what we do is to construct and then sample from a random subset of $\mathfrak{F}'$. Nevertheless, due to the use of $\Omega$ with the purpose of tuning randomly selected $\alpha_{i,j,k,m}$'s, we can still guarantee that there is a nonzero probability of each polynomial in $\mathfrak{F}'$ being sampled, resulting the sufficient security against directly guessing all the coefficients $\alpha_{i,j,k,m}$'s.

Since the feasibility of our CFA scheme relies on the polynomial $f(x,y,z,w)$ sampled from $\mathfrak{F}'$, it is important to guarantee that the size $|\mathfrak{F}'|$ of $\mathfrak{F}'$ is greater than zero. In addition, the polynomials in $\mathfrak{F}'$ can be thought of as the "composite" key in CFA. Thus, $|\mathfrak{F}'|$ should be as large as possible to defined against the adversary randomly guessing the coefficients of $f(x,y,z,w)$. After certain calculation, we can know that $|\mathfrak{F}'|$ can be lower bounded by

$$\sum_{i=1}^{(d+1)^4-(d+1)^2} \sum_{\rho=i+1}^{\lfloor (r-1)/\bar{\ell} \rfloor} \frac{1}{4(\rho-i)\sqrt{3}} e^{\pi\sqrt{\frac{2(\rho-i)}{3}}}. \tag{14}$$

The derivation details are described in Appendix A.

When $f(x,y,z,w)$ is selected from the weak constrained function set $\mathfrak{F}'$, the filtering capability will be slightly reduced. Its impact on the security of CFA using $f(x,y,z,w) \in \mathfrak{F}'$ is discussed in the following. Even if $f(x,y,z,w)$ is selected from $\mathfrak{F}'$, the destination and the intermediate nodes, when receiving the genuine message, can still correctly accept and forward the received message, respectively. The validation procedures are the same as those in (2) and (3), and therefore, are omitted here. The destination node and intermediate nodes, however, only *probabilistically* drop falsified messages in all cases of CFA using $f(x,y,z,w) \in \mathfrak{F}'$, instead of *deterministically* dropping the modified messages in certain cases[3] of CFA using $f(x,y,z,w) \in \mathfrak{F}$. The principle behind this change is as follows:

$$\begin{aligned}
&\mathrm{verf}_{\varepsilon}(u', K_{\varepsilon,u'}, h(m)) - \mathrm{MAC}_u(v,m) \\
&\quad = (f(u', \varepsilon, K_{\varepsilon,u'}, h(m)) + n_{\varepsilon,\mathfrak{v}}(u', K_{\varepsilon,u'}, h(m))) \\
&\qquad - (f(u, v, K_{u,v}, h(m)) + n_{u,\mathfrak{a}}(v, K_{u,v}, h(m)) + n_{u,s}) \\
&\quad = (f(u', \varepsilon, K_{\varepsilon,u'}, h(m)) - f(u, v, K_{u,v}, h(m))) \\
&\qquad + (n_{\varepsilon,\mathfrak{v}}(u', K_{\varepsilon,u'}, h(m)) \\
&\qquad\quad - (n_{u,\mathfrak{a}}(v, K_{u,v}, h(m)) + n_{u,s})).
\end{aligned} \tag{15}$$

The first term $(f(u', \varepsilon, K_{\varepsilon,u'}, h(m)) - f(u, v, K_{u,v}, h(m)))$ in (15) can be an arbitrary element in $\mathbb{F}_q$, leading also to the arbitrariness of the final result in (15). Therefore, when $\mathfrak{F}'$ is used, for the verification performed by the intermediate nodes, the probability that $|\mathrm{verf}_{\varepsilon}(u', K_{\varepsilon,u'}, h(m)) - \mathrm{MAC}_u(v,m)|$ happens to be within the range $[-2^r+1, 2^r-1]$ is maintained as $2^{r+1} - 1/q$ for the case where the adversary randomly generates the falsified MAC and is increased from 0 to $2^{r+1} - 1/q$ for the case where the adversary follows the procedures in CFA to generate the falsified MAC. With a similar argument, one can also show that, for the destination node, the probability that the falsified message successfully passes the verification on the intermediate node is $2^r - 1/q$, and, therefore, the probability of detecting falsified messages is $1 - (2^r - 1/q)$.

## IV. CFA-Based En-Route Filtering (CFAEF) Scheme

With CFA described in Section III, the design of CFAEF scheme is straightforward. The CFAEF scheme consists of three phases: node initialization phase, report endorsement phase, and en-route filtering phase, which, respectively, will be described as follows.

*Node Initialization Phase:* At first, a global security parameter $t$, which indicates the maximum number of compromised nodes tolerable in the CFAEF scheme, is selected. If the number of compromised nodes exceeds $t$, then the adversary can inject falsified data without being detected. It should be noted that such a limitation is also applied to all en-route filtering schemes unless additional location information is used. In addition, each node $u$ is preloaded with $\mathrm{auth}_u(y,z,w)$ and $\mathrm{verf}_u(x,z,w)$ prepared for the use of CFA. Last, the sensor nodes are deployed on the sensing region.

*Report Endorsement Phase:* After sensor deployment, a node enters this phase when it has an event report to be sent.[4] More specifically, once a node $u$ wants to send an event report $E$ to the destination node $v$, it first broadcasts $E$ in plaintext to its neighboring nodes. If the neighboring node $\mu$ agrees with $E$, then it generates an MAC, $\mathrm{MAC}_{\mu}(v,E)$ via the proposed CFA

[3]When the adversary follows the procedures in CFA, the falsified message can be deterministically detected.

[4]An event could be simultaneously observed by multiple nodes. Here we assume that one of these detecting nodes is responsible for sending the event report, but the election of such node is beyond the scope of this paper.

scheme, and sends an endorsement of $E$, $\mathrm{MAC}_{\mu}(v, E)$, back to $u$. After collecting $t$ MACs from the neighboring nodes,[5] $\mu_1, \ldots, \mu_t$, $u$ first checks whether the value of $|\mathrm{verf}_u(\mu_j, E) - \mathrm{MAC}_{\mu_j}(v, E)|, j = 1, \ldots, t$, is within the predetermined range $[0, 2^r - 1]$. Note that the sensor nodes $\mu_1, \ldots, \mu_t$ are *endorsing nodes* of the event report $E$ and the reporting node $u$. If some of the collected endorsements, $\mathrm{MAC}_{\mu_j}(v, E)$, fail to be verified, $u$ drops all $\mathrm{MAC}_{\mu_j}(v, E)$'s and acquires other endorsements from the neighboring nodes other than $\mu_1 \ldots, \mu_t$. $u$ forwards

$$\langle E, u, v, \mathrm{MAC}_u(v, E), \mu_1, \mathrm{MAC}_{\mu_1}(v, E), \ldots, \mu_t, \mathrm{MAC}_{\mu_t}(v, E)\rangle$$

to $v$ only when all of $t$ collected endorsements are successfully verified.

*En-Route Filtering Phase:* Once receiving the packet

$$\langle E, u, v, \mathrm{MAC}_u(v, E), \mu_1, \mathrm{MAC}_{\mu_1}(v, E), \ldots, \mu_t, \mathrm{MAC}_{\mu_t}(v, E)\rangle$$

the intermediate node $\varepsilon$ first checks whether the attached endorsements are generated by $t + 1$ distinct nodes. The packet is dropped if the verification fails. Afterwards, for each $\nu$ of the $t + 1$ endorsements, node $\varepsilon$ checks whether $\mathrm{VD}_{\varepsilon,\nu} = |\mathrm{verf}_{\varepsilon}(\nu, E) - \mathrm{MAC}_{\nu}(v, E)|$ is within the predetermined range $[0, 2^r - 1]$. Only if node $\varepsilon$ succeeds in verifying all the $t + 1$ endorsements, is the packet forwarded. Otherwise, the packet is dropped. The operation performed by the destination node $v$ is similar to that performed by the intermediate node. The difference is that $v$ checks whether $\mathrm{VD}_{v,\nu} = |\mathrm{verf}_{v}(\nu, E) - \mathrm{MAC}_{\nu}(v, E)|$ is within the predetermined range $[0, 2^{r-1} - 1]$. Only if $v$ succeeds in verifying all the $t + 1$ endorsements, is the event report $E$ accepted. Otherwise, the packet is dropped.

## V. Performance and Security Evaluation

In this section, for CFAEF, in addition to analyzing the overhead (Section V-A), we study its security (Section V-B) and compare the energy saving with the other methods (Section V-C).

### A. Overhead Analysis

As to the storage overhead, two trivariate polynomials need to be stored in each node in CFA, as shown in Fig. 1. Therefore, in CFAEF, the storage overhead $O(d^3)$ is required due to the use of authentication and verification polynomials.

For the endorsing node, the computation overhead comes from the calculation of the message authentication code, which involves trivariate polynomial evaluation and requires $O(d^3)$ arithmetic operations [5], [36]. On the other hand, the computation overhead for the source node, intermediate nodes, and destination node is the same, which is $O(td^3)$, because $t$ MACs should be calculated.

[5]The WSNs in our consideration possess high node density such that $t$-coverage [18], [39], [41] can be achieved.

As to the communication overhead of CFAEF, the source node has to communicate with the neighboring nodes to obtain the endorsements. Moreover, the source node has to send

$$\langle E, u, v, \mathrm{MAC}_u(v, E), \mu_1, \mathrm{MAC}_{\mu_1}(v, E), \ldots, \mu_t, \mathrm{MAC}_{\mu_t}(v, E)\rangle$$

instead of $\langle E, u, v\rangle$, to the destination node. As a result, the additional communication overhead incurred by the use of CFAEF is $O(tH)$, where $H$ is the average number of hops between two arbitrary nodes in a network. The issue of the energy consumption incurred by the use of CFA will be discussed in Section V-C in more details.

### B. Security

First, we study the security of the proposed CFA scheme. In particular, we assume that the adversary attempts to recover the coefficients of $f(x, y, z, w)$ and $\mathfrak{F}'$ is used in CFA. Please note that our security proof, compared to the pure descriptive manner of proving security in other works, suffices to demonstrate the security of our scheme at least in the engineering point of view, because many attacks (eavesdropping attack, node capture attack, reflection attack, and deperturbation attack) are considered and some of them (eavesdropping attack, node capture attack, and reflection attack) are further modeled and analyzed in our paper.

*Resilience Against Eavesdropping Attack:* Consider the adversary that can only modify the transmitted packet and retransmit the modified one in order to deceive the destination node into accepting that the packet originates from the other node or that the message is authentic. The probability of the adversary successfully deceiving the destination node can be analyzed as follows. If the message $m$ with $\mathrm{MAC}_u(v, m)$ sent by the node $u$ is modified to $m' \neq m$ or $u' \neq u$, then we can know that the probability that the intermediate node forwards the message $m'$ is at most $2^{r+1} - 1/q$ and the probability that the destination node accepts the message $m'$ is at most $2^r - 1/q$. This can be explained by the fact that, to deceive the destination node, the best strategy that can be adopted by the adversary is to forge the MAC corresponding to $m'$ and $u'$. Nonetheless, such MAC can only be randomly guessed by the adversary. Therefore, the verification difference would be arbitrary and the probabilities that $\mathrm{VD}_{\varepsilon,u}$ and $\mathrm{VD}_{v,u}$ happen to be within the predetermined ranges are at most $2^{r+1} - 1/q$ and $2^r - 1/q$ for the intermediate node and destination node, respectively.

*Resilience Against Node Capture Attack:* We consider the case where the adversary not only eavesdrops on the transmitted messages but also compromises $n$ nodes to use the security information stored in them, trying to recover the coefficients of $f(x, y, z, w)$. We can know that the adversary cannot break $f(x, y, z, w)$ if only $n \leq d$ nodes are compromised [3]. When the adversary has compromised $n > d$ nodes, the complexity for it to obtain the coefficients of $f(x, y, z, w)$ is $\Omega(q^{d+1})$. This can be explained as follows. Assume that $u_0, \ldots, u_{n-1}$ are $n$ compromised nodes. Let $x_0$, $z_0$, and $w_0$ be arbitrary elements in $\mathbb{F}_q$. We know that if we can arbitrarily construct $f(x_0, y, z_0, w_0)$ for any $x_0$, $y_0$, and $w_0$, then the coefficients of $f(x, y, z, w)$ can

be inferred by solving a system of equations. Thus, our goal is to obtain the coefficients of $f(x_0, y, z_0, w_0)$. Note that the discussion and effect of obtaining the coefficients of, for example, $f(x, y_0, z_0, w_0)$ is the same as that of obtaining the coefficients of $f(x_0, y, z_0, w_0)$. Thus, we omit the former case and focus only on the latter case here. We can know that $f(x_0, y, z_0, w_0)$ can always be written as $\sum_{j=0}^{d} C_j y^j$. Based on the construction of $\mathrm{verf}_u(x, z, w)$, we can derive the following $n$ equations:

$$\sum_{j=0}^{d} C_j (u_i)^j = \mathrm{verf}_{u_i}(x_0, z_0, w_0) - n_{u_i,\mathfrak{b}}(u_i, z_0, w_0), \quad 0 \le i \le n-1. \quad (16)$$

In this system of equations, there are $d+1+n$ unknown variables including $C_j$ $(0 \le j \le d)$ and $n_{u_i,\mathfrak{b}}(u_i, z_0, w_0)$ $(0 \le i \le n-1)$. There are, however, only $n$ equations. Thus, $d+1$ unknown variables should be eliminated or correctly guessed. The polynomials, $\mathrm{auth}_{u_i}(y, z, w)$'s, may be used by the adversary to reduce the number of unknown variables.

*Resilience Against Reflection Attack:* A method that is able to reduce the number of unknown variables is called *reflection attack* in [55] and is employed here. Let $a_i = \mathrm{verf}_{u_i}(u_0, z_0, w_0) - \mathrm{auth}_{u_0}(u_i, z_0, w_0) = n_{u_i,\mathfrak{b}}(u_i, z_0, w_0) - n_{u_0,\mathfrak{a}}(u_i, z_0, w_0)$. The above equation can be rewritten as $n_{u_i,\mathfrak{b}}(u_i, z_0, w_0) = a_i + n_{u_0,\mathfrak{a}}(u_i, z_0, w_0)$. Together with this equation, (16) can be represented as

$$\sum_{j=0}^{d} C_j (u_i)^j = \mathrm{verf}_{u_i}(x_0, z_0, w_0) - a_i - n_{u_0,\mathfrak{a}}(u_i, z_0, w_0), \quad 0 \le i \le n-1. \quad (17)$$

It can be observed that reflection attack does not work in breaking $f(x, y, z, w)$ with higher probability because there are still $d+1+n$ unknown variables in $n$ equations. Thus, $d+1$ unknown variables should be eliminated or correctly guessed. Since each unknown variable can be of at least $r$ bits length, the complexity of recovering the coefficients is $\Omega(2^{r(d+1)})$.

*Resilience Against Deperturbation Attack:* Very recently, an attack was proposed in [2] to break the perturbation polynomial-based authentication schemes by compromising only a few sensor nodes. In general, it relies on the following two observations. First, when the notion of perturbation is applied only on a few coefficients in the secret polynomial, the adversary can recover almost all the coefficients not affected by the perturbation. Second, the coefficients affected by the perturbation are actually independent of the message to be sent itself. In the following, the terms "*affected coefficients*" and "*unaffected coefficients*" are used to denote the coefficients affected by the perturbation and the coefficients *not* affected by the perturbation, respectively. Note that, according to the first observation, unaffected coefficients can be fully recovered by the adversary. With these two observations, the adversary can constitute the MAC corresponding to the forged message $m'$ sent by an arbitrary uncompromised node $u'$ as follows. The polynomial that is constituted by the unaffected coefficients is evaluated according to $u'$ and $m'$. Afterwards, the result in the former derivation is combined with the affected coefficients eavesdropped from the other packets sent by $u'$. Consequently, the final result can be used to authenticate $m'$ sent by $u'$. As the first victim of this attack, the security of Zhang *et al.*'s scheme [55] is broken.

Here, we notice that the attack proposed in [2] is very similar to the *protocol attacks* such as *copy attack* [6], [24], [27], [32] in the watermarking community. In particular, with the observation that the watermark embedded in a cover signal $s$ as a watermarked signal $s^w$ is independent of $s$ itself, the copy attack aims to denoise $s^w$ to get the hidden watermark, which will be copied and embedded into another signal $s'$ so that the watermark can also be extracted/detected from $s'$. This creates the false-positive problem. The usual defense against copy attacks is to make the watermark content-dependent. In other words, the watermark is sufficiently correlated with the cover signal itself.

Based on the above observations, it turns out that our CFA scheme also has the similar content-dependent feature. Specifically, recall from Fig. 1 that $\mathrm{auth}_u(y, z, w) := f(u, y, z, w) + n_{u,\mathfrak{a}}(y, z, w)$ and $\mathrm{verf}_u(x, z, w) := f(x, u, z, w) + n_{u,\mathfrak{b}}(x, z, w)$. It can be observed that the perturbation to be added to the secret polynomial depends on not only the source/destination node ID, but also the message to be sent itself. Hence, on the one hand, the attack proposed in [2] cannot be applied to our CFA scheme because, in the CFA scheme, all the coefficients in the secret polynomial have the possibility to be affected by the introduced perturbation so that quite a few coefficients in the secret polynomial can be easily estimated. On the other hand, the perturbation introduced for different messages are dependent on the message itself. Thus, the adversary can no longer act in such a way that it can reuse the MAC corresponding to the message $m$ in the construction of the MAC corresponding to the message $m'$. As a whole, our CFA scheme can be resilient against the attack proposed in [2].

*Filtering Capability of CFAEF:* After the security of CFA is established, the resilience of CFAEF against false data injection attack, PDoS attack, and FEDoS attack is obvious. For example, CFAEF is resilient to false data injection attack and PDoS attack because, with the MACs generated by CFA, the false data can be detected and dropped by either intermediate nodes or the destination node when the number of compromised nodes does not exceed $t$. In particular, with $t$ endorsements required, the probabilities of detecting the bogus message on each intermediate node and the destination node are $1-(2^{r+1}-1/q)^t$ and $1-(2^r-1/q)^t$, respectively. On the other hand, FEDoS attack is useless for the adversary because if the compromised node sends a false endorsement to the source node, the source node, acting as the intermediate node between the endorsing node and destination, can identify the false endorsement via the CFA verification, and refuse to communicate with the compromised node thereafter.

### C. Energy Savings

In this section, the energy consumption model similar to that used in [35], [43], [49] is used to analyze the energy savings of various schemes. Due to the fact that, the higher the filtering capability, the lower the energy consumed for forwarding falsified messages, the evaluation of energy consumption is somewhat equivalent to the evaluation of the filtering capability. Thus, in the literature, the evaluation of the energy savings heavily relies on counting the number of hops the falsified message traveled.

TABLE II
COMPARISON OF ENERGY CONSUMPTION OF DIFFERENT EN-ROUTE FILTERING SCHEMES

| | Energy Consumption of Computation | Energy Consumption of Communication |
|---|---|---|
| no filtering | $\frac{t}{2}e_h$ | $192H(1+\beta)(e_t+e_r)$ |
| SEF [49] | $\frac{t}{2}e_h$ | $(256+10t)(H+\frac{\beta}{0.01t})(e_t+e_r)$ |
| DEF [43] | $c_1 e_h$ | $(215+104t)(H+\frac{\beta}{0.055t})(e_t+e_r)$ |
| IHA [54] | $3e_h$ | $(192+64t)(H+\beta\frac{t}{2})(e_t+e_r)$ |
| LBRS [50] | $c_2 e_h$ | $(192+64t)(H+\frac{\beta}{0.01t})(e_t+e_r)$ |
| LEDS [35] | $2e_h$ | $c_3\beta(192+64t)(H+\beta\frac{t}{2})(e_t+e_r)$ |
| GREF [45] | $c_6 e_h$ | $(192+64t)(H+\frac{\beta}{\frac{t-c_4}{t}c_5})(e_t+e_r)$ |
| CFAEF (this paper) | $(t+1)(d+1)^3(3\log d+c_7)e_m$ | $(192+16t)(H+\frac{\beta}{1-\frac{2^{r+1}-1}{q}})(e_t+e_r)$ |

Nonetheless, only counting the number of the hops the falsified message traveled will mislead us into thinking that the en-route filtering scheme being considered is both efficient and effective. We argue that, to correctly evaluate the efficiency and effectiveness of an en-route filtering scheme, calculating the corresponding energy consumption works as the best measurement. We later will describe the disadvantage of using the number of hops the falsified message traveled, and the advantage of directly using the energy consumption.

*Energy Calculation:* A general formula, $L_r(H+(\beta/p))$, is shown in [43] for evaluating the number of bits to be transmitted in the report forwarding with the consideration of en-route filtering schemes, where $L_r$, $H$, $\beta$, and $p$ denote the bit-length of the report plus endorsements, the average number of hops between two arbitrary nodes, the ratio of the false report to the legitimate report, and the probability of detecting the false report on each node, respectively. As $e_t$ and $e_r$ denote the energy for transmitting one single bit and the energy for receiving one single bit, respectively, we use the formula

$$E = L_r\left(H+\frac{\beta}{p}\right)(e_t+e_r) \tag{18}$$

to calculate the energy consumption incurred by the communication of en-route filtering schemes. Note that the above formula only works for probabilistic key sharing-based en-route filtering schemes such as SEF [49] and DEF [43]. For the deterministic key sharing-based en-route filtering schemes such as IHA [54] and LEDS [35], the corresponding energy consumption should be considered otherwise. Note also that the energy consumption incurred by the computation is usually not considered. Nevertheless, because our proposed scheme requires more computation, for fairness of comparison, the energy consumption incurred by the computation is included in the energy calculation here. The energy consumption of different schemes is summarized in Table II. It should be noted that the energy consumption values in Table II are the approximate ones because different parameters in different schemes should be considered. Although the formulas of energy consumption are provided in some works [35], [43], [49], it is still cumbersome to list all of the notational details even without explaining their meanings. For simplification, we only provide the approximate energy consumption in different en-route filtering schemes in Table II. The derivation of approximate energy consumption in Table II of different schemes is described in Appendix B.

Let $e_X^{\mathrm{comp}}$ and $e_X^{\mathrm{comm}}$ be the energy consumption incurred by the computation and communication, respectively, of the en-route filtering scheme $X$. For example, $e_{\mathrm{Ord}}^{\mathrm{comp}}$ is the energy consumption incurred by the computation when no en-route filtering scheme is used, and $e_{\mathrm{SEF}}^{\mathrm{comm}}$ is the energy consumption incurred by the communication of SEF. We further define $e_X = e_X^{\mathrm{comp}} + e_X^{\mathrm{comm}}$ as the total energy consumption of $X$. Throughout the energy evaluation, the common parameters, MAC size with 64 bits and the byte-length of the report with 24 bytes, were used for the methods adopted for comparisons. From [43], with the default parameter setting, we know that $e_{\mathrm{Ord}} = 192H(1+\beta)$, $e_{\mathrm{SEF}} \cong (t/2)e_h + (256+10t)(H+(\beta/0.01t))$, and $e_{\mathrm{DEF}} \cong c_1 e_h + (215+104t)(H+(\beta/0.055t))$,[6] where $c_1$ is defined in Appendix B.

According to (18), with the calculation[7] similar to [43] and the setting of $\ell = 16$ and $r = 11$, the energy consumption $e_{\mathrm{CFAEF}}^{\mathrm{comm}}$ in CFAEF can also be derived as $e_{\mathrm{CFAEF}}^{\mathrm{comm}} \cong (192+16t)(H+(\beta/1-(2^{r+1}-1/q)))$. In CFAEF, each intermediate node needs to check the legitimacy of $t+1$ MACs. Each check involves a trivariate polynomial evaluation. In the worst case, the intermediate nodes do not resort to sophisticated algorithms in the polynomial evaluation; i.e., the polynomial evaluation is accomplished by calculating the values term by term and finally summing the evaluation result of each term up. As the multivariate squaring method in calculating exponentiation is used, the number of multiplications required is $(d+1)^3(3\log d+c_7)$, where $c_7$ is the joint Hamming weight of the exponents of variables in the term under the consideration. For example, as the term $x^3y^3z^3$ is considered, its joint Hamming weight $c_7$ of the exponents will be six since there are six ones in the binary representation of the exponents. It may be observed that the computation overhead of CFAEF is higher than that of the other schemes. Nonetheless, the communication overhead of CFAEF is substantially lower than that of the other schemes. This can be validated by the subsequent numerical results. Moreover, CFAEF possesses several nice characteristics such as the resilience against FEDoS attack and the applicability on mobile

[6]It seems to have some erroneous calculations in deriving $p$ of DEF because the derivation of $p$ in [43] is independent of $t$. Nevertheless, $p$ should be positively proportional to $t$ in essence. Thus, the formula of $e_{\mathrm{DEF}}$ used here is an approximation based on our observation that $p$ should be somewhat linearly dependent on $t$. Note that 0.055 in $e_{\mathrm{DEF}}$ is obtained by simply dividing 0.275, which is provided in [43] as the $p$ of DEF when $t=5$, by 5.

[7]In [43], the packet length is only calculated based on counting the lengths of the report and MACs excluding the lengths contributed from the source node ID, destination node ID, and endorsing nodes IDs.

TABLE III
CHARACTERISTICS OF DIFFERENT EN-ROUTE FILTERING SCHEMES

| | Resilience against PDoS Attack | Resilience against FEDoS Attack | Require Stable Routing | Require Secure Bootstrapping Time | Require Location Information | Can be applied on Multi-Sink Network | Can be applied on Mobile Network |
|---|---|---|---|---|---|---|---|
| SEF [49] | Yes | No | No | No | No | Yes | Yes |
| IHA [54] | Yes | No | Yes | Yes | No | No | No |
| DEF [43] | Yes | No | No | No | No | No | No |
| LBRS [50] | Yes | No | No | Yes | Yes | No | No |
| LEDS [35] | Yes | No | Yes | Yes | Yes | No | No |
| GREF [45] | Yes | No | No | Yes | Yes | Yes | No |
| CFAEF (this paper) | Yes | Yes | No | No | No | Yes | Yes |

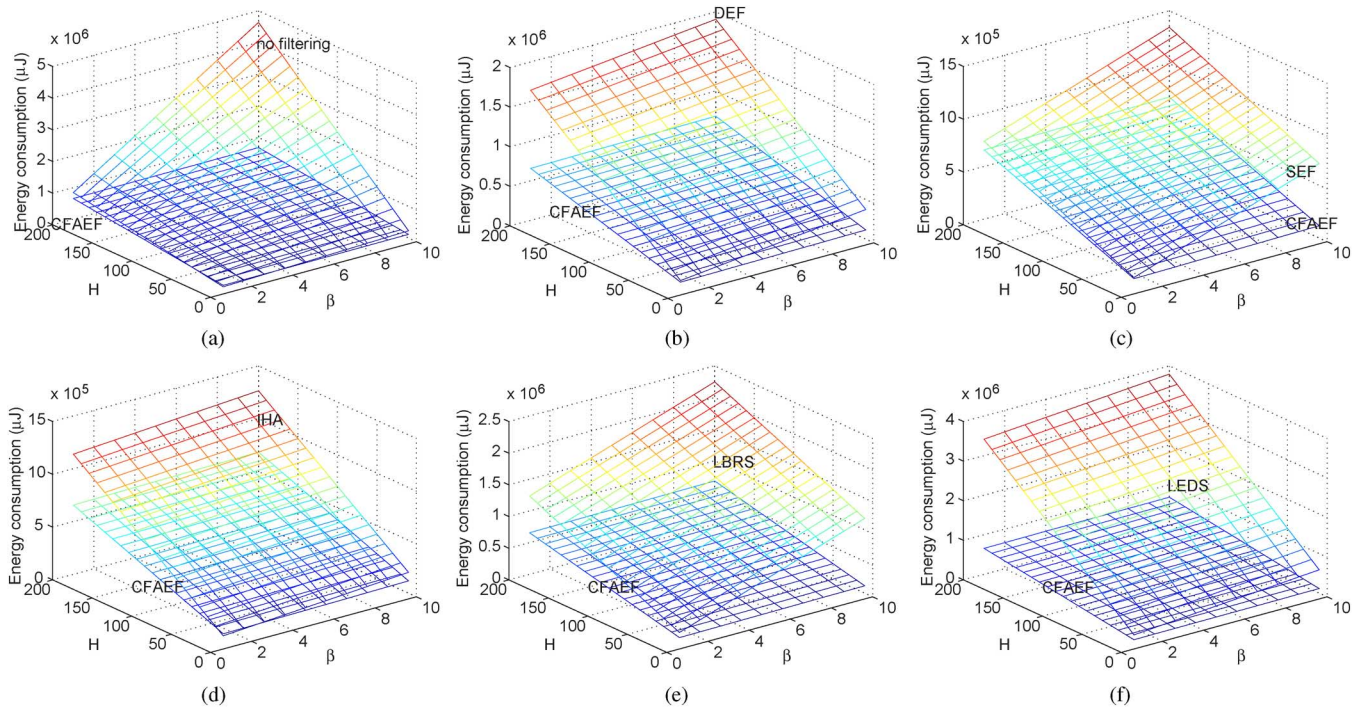



Fig. 4. Numerical results of the energy consumption as a function of $h$ and $\beta$ (best viewed on a color display). (a) CFAEF versus no en-route filtering; (b) CFAEF versus DEF; (c) CFAEF versus SEF; (d) CFAEF versus IHA; (e) CFAEF versus LBRS; (f) CFAEF versus LEDS.

networks, which are listed in Table III. The properties listed in Table III will be detailed later.

*The Metrics:* It has been demonstrated in [43] that the filtering effectiveness of DEF is superior to that of SEF; i.e., the number of hops the falsified message traveled in SEF is greater than that in DEF. Nevertheless, as our numerical results shown in Fig. 4(b) and (c) indicate, we have an interesting observation that $e_{\mathrm{SEF}}$ is generally lower than $e_{\mathrm{DEF}}$. The reason for this is that, although DEF indeed has better filtering effectiveness, it also has greater packet overhead, which reduces the benefit of energy savings especially in the case where no bogus message is injected. The above argument can also be validated by observing that $e_{\mathrm{DEF}}$ grows rapidly (slowly) as $H$ ($\beta$) increases and $e_{\mathrm{SEF}}$ grows rapidly (slowly) as $\beta$ ($H$) increases. This stems from the following three facts. First, the packet overhead of DEF is larger than that of SEF in general. Second, since the filtering effectiveness of DEF is superior to that of SEF, when more bogus data are injected into the network, DEF can detect them as early as possible, resulting in the merely slight increase of $e_{\mathrm{DEF}}$. Nevertheless, the increase of $H$ leads to the rapid growth of $e_{\mathrm{DEF}}$ because of a large amount of energy consumed for transmitting the larger packets in DEF. Third, because of the inferior filtering effectiveness, SEF is relatively vulnerable to false data injection attacks, implying that, when more bogus messages are considered, $e_{\mathrm{SEF}}$ grows quickly. $e_{\mathrm{SEF}}$, however, grows steadily since transmitting small-size packets in SEF consumes less energy. Recall that en-route filtering schemes are utilized to reduce the energy waste even when the adversary can inject the bogus data into the network. Nevertheless, en-route filtering schemes should also be energy-efficient when the adversary does not inject the bogus data. Thus, from the above observations of $e_{\mathrm{SEF}}$ and $e_{\mathrm{DEF}}$, we argue that, solely considering the "filtering effectiveness" is meaningless. For example, the network adopting DEF will deplete the energy soon even if the adversary does not inject the bogus data. Hence, to evaluate the efficiency and effectiveness of an en-route filtering scheme, the best way is to estimate its energy consumption in the presence of the adversary being able to launch false data injection, PDoS, and FEDoS attacks.

*Numerical Results:* In the following, the numerical results on energy consumption of various schemes will be presented.

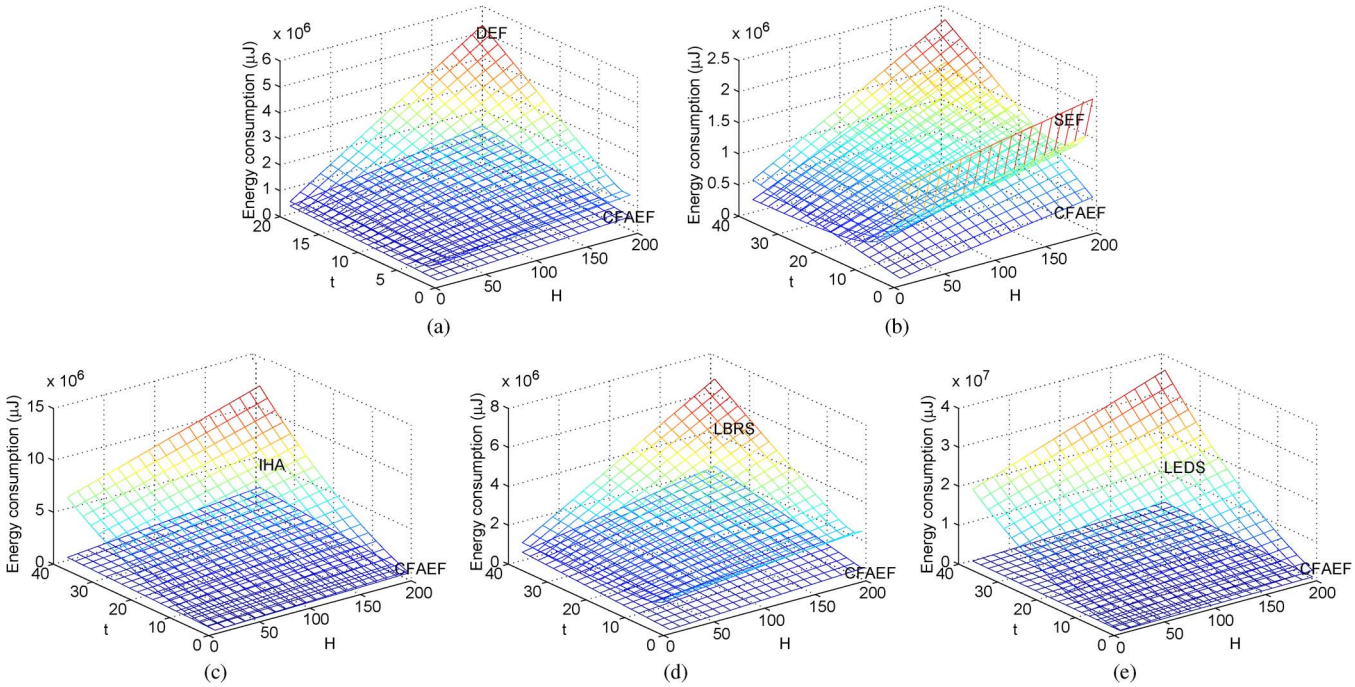



Fig. 5. Numerical results of the energy consumption as a function of $t$ and $H$ (best viewed on a color display). (a) CFAEF versus DEF; (b) CFAEF versus SEF; (c) CFAEF versus IHA; (d) CFAEF versus LBRS; (e) CFAEF versus LEDS.

In our evaluation, the setting of $c_1 = c_2 = 10$ and $c_3 = 3$ was used. When CFAEF is considered, we assume that $c_7 = 20$. The setting of $e_t = 7.4$ $\mu$J/bit, $e_r = 3.575$ $\mu$J/bit, and $e_h = 0.7375$ $\mu$J/bit was used based on the measurement results on MICA2 mote in [40]. To our knowledge, there is no known measurement result on $e_m$. Nevertheless, we believe that $e_m$ should be less than the energy required for AES decryption. Thus, $e_m$ is assumed to be 0.31123 $\mu$J/bit. Since only $e_{\mathrm{CFAEF}}$ involves with $e_m$, such overestimation of $e_m$ can be used to obtain the upper bound of $e_{\mathrm{CFAEF}}$.

The numerical results of the energy consumption as a function of $H$ and $\beta$ are shown in Fig. 4. For all schemes, the larger the $H$ or $\beta$, the more the energy consumed. This is obvious because the increase of $H$ implies the increase of the number of hops the genuine and bogus messages travel, and the increase of $\beta$ implies the increase of the number of packets to be transmitted. It can also be known from Fig. 4 that $e_{\mathrm{CFAEF}}$ is the lowest among all the schemes while the energy consumption of the other schemes increases dramatically.

The numerical results of the energy consumption as a function of $t$ and $H$ are depicted in Fig. 5. Note that $e_{\mathrm{Ord}}$ is independent of $t$, and, therefore, is ignored here. Because of the high filtering capability in DEF, the larger $t$ only marginally improves the higher filtering effectiveness, but the larger $t$ has the substantially larger packet overhead, degrading the energy savings. Nonetheless, since the packet overhead in DEF is significantly larger than that in CFAEF, compared to $e_{\mathrm{CFAEF}}$, $e_{\mathrm{DEF}}$ grows promptly as $t$ increases. Hence, it can be shown in Fig. 5(a) that, as $t$ increases, $e_{\mathrm{CFAEF}}$ grows relatively slowly. On the other hand, although each endorsement in SEF only has poor filtering capability, when multiple endorsements are used, the filtering effectiveness can be enhanced. It turns out that, as $t$ increases, the packet overhead only slightly increases but the filtering effectiveness are greatly enhanced. Thus, $e_{\mathrm{SEF}}$ decreases as $t$ increases.[8] In particular, because of the low packet overhead, when $t \leq 20$, with the increased packet overhead due to the raise of $t$, the filtering capability of SEF has the greater improvement in detecting the bogus message, resulting in the reduction of $e_{\mathrm{SEF}}$. Nevertheless, when $t \geq 20$, since the filtering capability cannot be enhanced anymore in our parameter setting, SEF also encounters the problem that the increase of $t$ only implies the larger $e_{\mathrm{SEF}}$. Thus, as for the comparison of SEF and CFAEF, under the conditions of relative low packet overhead and $t \leq 20$, $e_{\mathrm{CFAEF}}$ is lower than $e_{\mathrm{SEF}}$ mainly because CFAEF can offer greater filtering capability and therefore reduce the energy wasted on transmitting falsified messages. When $t \in [20, 40]$ as shown in Fig. 5(b), $e_{\mathrm{CFAEF}}$ is lower than $e_{\mathrm{SEF}}$ primarily because the packet overhead in CFAEF incurred by the increase of $t$ is still lower than SEF. Note that, in the cases where $t$ becomes larger, $e_{\mathrm{CFAEF}}$ will be larger than $e_{\mathrm{SEF}}$ eventually because the packet overhead of CFAEF is lower than that of SEF only in certain cases. Nonetheless, in the realistic sensor network applications, the sensor node does not have so many neighboring sensor nodes. Hence, it is not necessary to consider such cases. Recall that IHA guarantees that the false report can be detected within at most $t$ hops. As shown in Fig. 5(c), because the packet overhead will be increased and at the same time the upper bound of the number of hops the false reports can travel is increased, $e_{\mathrm{IHA}}$ is positively proportional to $t$. Since LBRS and LEDS can be thought of as the location-based variants of SEF and IHA, respectively, the curves of their energy consumption shown in Fig. 5(d) and (e) are similar to those shown in Fig. 5(b) and (c), respectively. Note that this kind of similarity can also be observed in Figs. 4 and 6.

[8]In fact, as $t \geq 20$, $e_{\mathrm{SEF}}$ turns to increases, because the increase of $t$ does not improve the filtering effectiveness and only increases the packet overhead.

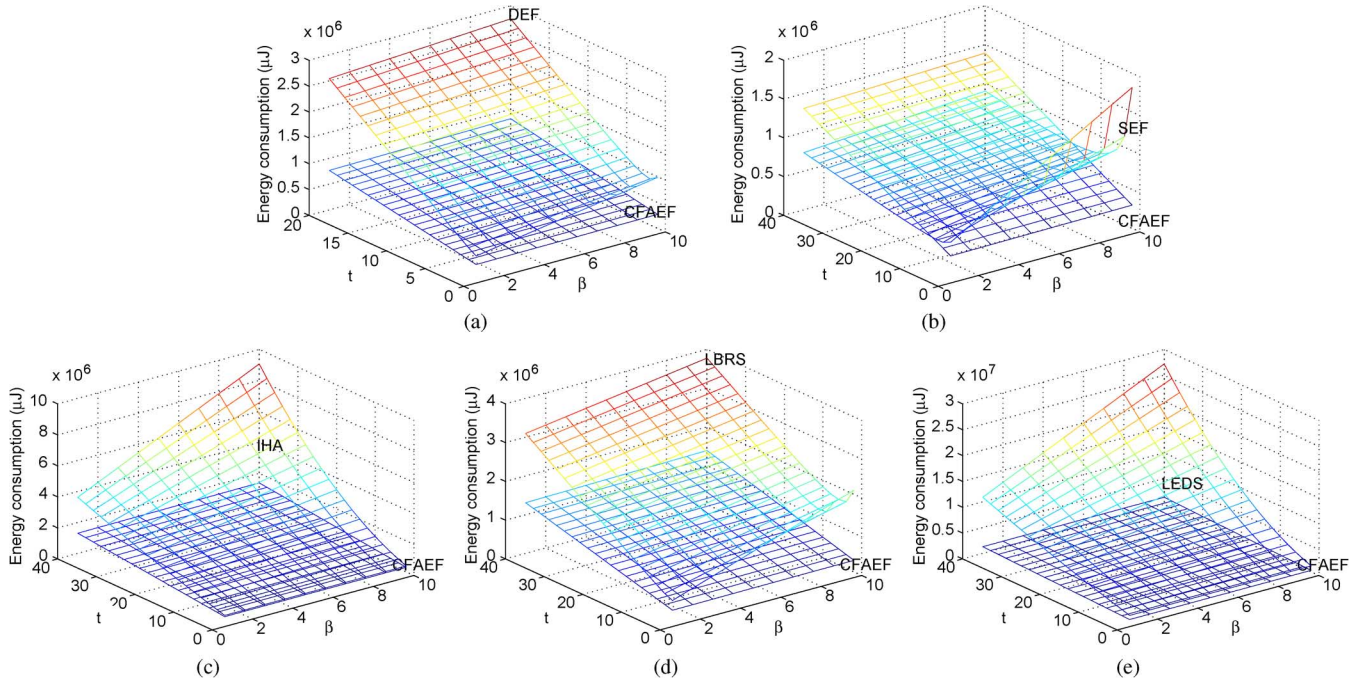



Fig. 6. Numerical results of the energy consumption as a function of $t$ and $\beta$ (best viewed on a color display). (a) CFAEF versus DEF; (b) CFAEF versus SEF; (c) CFAEF versus IHA; (d) CFAEF versus LBRS; (e) CFAEF versus LEDS.

The numerical results of the energy consumption as a function of $t$ and $\beta$ are depicted in Fig. 6. It can be seen from Fig. 6 that $e_{\mathrm{CFAEF}}$ is lower than $e_{\mathrm{DEF}}$ in all cases but $e_{\mathrm{CFAEF}}$ is lower than $e_{\mathrm{SEF}}$ only in certain cases. Specifically, when the setting of $t \geq 20$ was used, $e_{\mathrm{CFAEF}}$ could be greater than $e_{\mathrm{SEF}}$ when different $\beta$'s were used. This can also attribute to the difference of the packet overhead between CFAEF and SEF. Note that the packet overhead of CFAEF and SEF are $(192 + 16t)$ and $(256 + 10t)$, respectively. This means that the packet overhead of CFAEF will be greater than that of SEF when $t$ becomes larger. Moreover, although the filtering effectiveness of each endorsement in CFAEF is superior to that in SEF, because the increase of $t$ only has the marginal improvement of the filtering effectiveness, the overall filtering effectiveness of CFAEF is actually similar to that of SEF when the filtering effectiveness of multiple endorsements is considered. Hence, $e_{\mathrm{CFAEF}}$ will gradually become larger than $e_{\mathrm{SEF}}$ because when the filtering capability are similar, the packet overhead dominates the energy consumption. Note that, it is not necessary to consider the case where large $t$ is required because the sensor node rarely has over 20 neighboring sensor nodes in real applications. Also note that, even a highly dense network is considered, although $e_{\mathrm{SEF}}$ could be lower than $e_{\mathrm{CFAEF}}$, SEF is still vulnerable to PDoS and FEDoS attacks whereas CFAEF is inherently resilient against PDoS and FEDoS attacks.

*Discussion:* As shown above, at the expense of increased computation and storage overhead, the filtering capability of CFAEF is superior to that in the other schemes and, therefore, the energy savings due to the use of CFAEF is better than the others. However, the design of CFAEF possesses additional advantages. For example, when the other schemes are used, even the existence of one single compromised node could neutralize the advantages gained from the use of en-route filtering schemes because this compromised node can periodically and intentionally send the false endorsements so that all of the genuine reports will be considered falsified. Nonetheless, because of the use of CFA in CFAEF, CFAEF is inherently resilient against FEDoS attacks.

In addition, several existing schemes [35], [45], [50], [54] base their security on the existence of a period of secure bootstrapping time. Within this time, the adversary cannot launch attacks. As indicated in [35], such an assumption is impractical. Therefore, this kind of scheme would be useless in the harsh and hostile environments. Nonetheless, the security of CFAEF does not rely on this assumption.

Some schemes [35], [54] require the complicated security association among nodes. Here, the complicated security association means that one particular node must have key-sharing with some of the other specific nodes. When this kind of association is established, the en-route filtering becomes simple because the intermediate nodes can utilize the pre-established key shared with the other nodes to check the legitimacy of the received messages. Nevertheless, this kind of association also draws a lot of drawbacks: 1) Since the association is fixed, one should repair the association if the nodes run out of the energy or malfunction due to unknown reasons. 2) The subsequent messages must pass through the nodes on a fixed path to the BS. Otherwise, since the nodes not on the predefined path could possibly not have the key shared with the nodes on the predefined path, the en-route filtering will not work. This implies the need of very stable routing, which is not feasible in all cases. 3) Unless specific design is involved or the security associations are formed on all of the possible paths from the nodes that will generate the reports to the BS, in general, the en-route filtering schemes with the requirement of complicated security association cannot be applied to the network with multiple sinks and on mobile net-

TABLE IV
POSSIBLE PARAMETER SETTINGS

| $q$ | $\ell$ | $r$ | $d$ | The Probability of Detecting False Report | Storage Overhead (bit) |
|---|---|---|---|---|---|
| $2^{16}-15$ | 16 | 13 | 2 | $2^{-2}$ | 432 |
| $2^{16}-15$ | 16 | 13 | 5 | $2^{-2}$ | 3456 |
| $2^{32}-5$ | 32 | 27 | 5 | $2^{-4}$ | 6912 |
| $2^{32}-5$ | 32 | 27 | 4 | $2^{-4}$ | 4000 |
| $2^{32}-5$ | 32 | 24 | 5 | $2^{-7}$ | 6912 |

works. 4) As the security association is used in conjunction with the location information, some special assumptions should be included. For example, a mobile robot is needed in LEDS [35] while the secure bootstrapping time is required in IHA [54]. Nonetheless, our scheme does not need such an "initial setup phase" to setup the security associations among sensor nodes. In particular, although some security materials should be stored in each sensor before sensor deployment (Offline Step of CFA), the CFA scheme can work immediately after sensor deployment without any configurations or assumptions. In addition, when a message needs to be authenticated, although the sensor needs to generate the key via CARPY+ [46], establishing the shared key in CARPY+ does not rely on the other assumptions and even does not require the communication between sensors. Hence, due to the above reasons, we claim that our CFAEF does not require the establishment of the complicated security associations among sensors.

Actually, each of the existing en-route filtering scheme can be applied to the network with multisink. Nevertheless, as shown in Table III, it is not economical to implement some en-route filtering schemes on the multisink network. Here, "not economical" means that although IHA [54] (or DEF [43] or LBRS [50] or LEDS [35]) can be adapted to be applied to the network with multisink in a straightforward way that the network planner just applies IHA (or DEF or LBRS or LEDS) many times, the overhead will also be linearly increased with the number of data sinks, which is a disaster. Thus, we consider some of the existing schemes incapable of being applied to the network with multisink.

Finally, as the location information is required for each node in some schemes [35], [45], [50], which consume additional energy, each node in our proposed CFAEF scheme does not require us to derive its geographic position and, therefore, will be more energy-efficient.

*Prototype Implementation:* To study the practicality of our proposed CFA scheme for the current generation of sensors, a prototype of CFA on TelosB motes on top of the TinyOS platform was implemented (Micro-Controller: TI MSP430F1611; ROM: 48KB+256B; RAM: 10KB; Radio Chipset: ChipCon CC2420).

There are many possible combinations of parameter settings in our CFA, some of which are shown in Table IV. In the implementation, the setting of $\ell = 32$, $r = 24$, and $d = 5$ was used. To keep the prototype implementation as simple as possible, the coefficients in polynomial $f(x, y, z, w)$ are artificially selected so that the trivariate polynomial evaluation will not overflow. In general, $2(d+1)^3$ coefficients from authentication polynomial and verification polynomial should be stored in the sensor node. Nevertheless, since these two polynomials are fixed for each node, if there are only a few nonzero coefficients in these two polynomials, one would store only the nonzero coefficients for reducing storage overhead. The MAC generation function was implemented based on CBC-MAC mode. Since there is a hardware-based AES encryption function in CC2420 chipset, this hardware-based AES encryption function is launched as it is required in CBC-MAC execution. In addition, in our program code, we do not implement the function of pairwise key generation. Instead, the pairwise key is treated as a constant in the program code. As a whole, the ROM needed for our program code is 19724 bytes and the RAM needed for our program code is 1668 bytes.

Our prototype was also run on TOSSIM, which is a discrete-event simulator especially designed for TinyOS sensor networks, to evaluate the energy consumption of CFA. The TinyOS code can be directly executed on TOSSIM so that TOSSIM can report the energy consumption. Due to this feature, the energy consumption reported by TOSSIM would be convincing. In the TOSSIM simulation, since the hardward-based AES function in CC2420 is not supported by TOSSIM, we, instead, implemented a software-based AES function for CFA. We focus on the energy consumption incurred by the computation. Thus, the radio module is not included in the program code. In our simulation, the CFA computation is triggered per second. The period we conducted the simulation was 60 s. The energy consumption reported by TOSSIM is 706.197 mJ.

## VI. CONCLUSION

A CFA scheme, which can be thought of as a hash function directly supporting en-route filtering functionality, was proposed. According to CFA, we constructed a CFAEF scheme to simultaneously defend against false data injection, PDoS, and FEDoS attacks. Some theoretical and numerical analyses were provided to demonstrate the efficiency and effectiveness of CFAEF.

## APPENDIX A
## DERIVATION OF LOWER BOUND OF $|\mathfrak{F}'|$

Recall that the coefficients of the polynomial $f(x, y, z, w)$ used in CFA work as a kind of key, and should be kept secret. Thus, the polynomials in $\mathfrak{F}'$ can be thought of as the "composite" key in CFA, and $|\mathfrak{F}'|$ is an important indicator to evaluate the security of CFA because if there are only few choices in $\mathfrak{F}'$ then the adversary can obtain the same polynomial used in CFA by performing $\mathfrak{F}'$-Construction algorithm as well. We know from (8) and (11) that $|\mathfrak{F}'|$ can be represented as

$$\begin{aligned}|\mathfrak{F}'| = \Big|\big\{[\alpha]| - 2^{r-1} \\ \leq \sum_{i,m=0,j,k=1}^{d} \left(\alpha_{i,j,k,m} x^i w^m \left(y^j z^k - (y')^j (z')^k\right)\right) \\ \leq 2^{r-1}, \quad x \in [x_{\min}, x_{\max}], y \in [y_{\min}, y_{\max}], \\ z \in [z_{\min}, z_{\max}], w \in [w_{\min}, w_{\max}]\big\}\Big|. \end{aligned} \tag{19}$$

Without loss of generality, we assume that $x_{\min} = y_{\min} = z_{\min} = w_{\min} = 0$. Therefore, (19) can be written as

$$\begin{aligned}|\mathfrak{F}'| &= \left| \left\{ [\alpha] | \sum_{i,m=0,j,k=1}^{d} \left( \alpha_{i,j,k,m} x_{\max}^{i} w_{\max}^{m} \left( -y_{\max}^{j} z_{\max}^{k} \right) \right) \right. \right. \\ &\quad \geq -2^{r-1}, \\ &\quad \left. \left. \sum_{i,m=0,j,k=1}^{d} \left( \alpha_{i,j,k,m} x_{\max}^{i} w_{\max}^{m} \left( y_{\max}^{j} z_{\max}^{k} \right) \right) \right. \right. \\ &\quad \left. \left. \leq 2^{r-1} \right\} \right| \\ &= \left| \left\{ [\alpha] | | \sum_{i,j=0,k,m=1}^{d} \alpha_{i,j,k,m} x_{\max}^{i} y_{\max}^{j} z_{\max}^{k} w_{\max}^{m} \right. \right. \\ &\quad \left. \left. \leq 2^{r-1} \right\} \right|. \end{aligned} \tag{20}$$

However, it is difficult to accurately estimate $|\mathfrak{F}'|$. Thus, we instead provide a lower bound of $|\mathfrak{F}'|$. To ease the subsequent calculation, we assume that $x$, $y$, $z$, and $w$ are of the same length $\bar{\ell}$. It follows that

$$\begin{aligned}|\mathfrak{F}'| &= \left| \left\{ [\alpha] | \sum_{i,m=0,j,k=1}^{d} \alpha_{i,j,k,m} 2^{\bar{\ell} i} 2^{\bar{\ell} j} 2^{\bar{\ell} k} 2^{\bar{\ell} m} \leq 2^{r-1} \right\} \right| \\ &= \left| \left\{ [\alpha] | \sum_{i,m=0,j,k=1}^{d} \alpha_{i,j,k,m} 2^{\bar{\ell}(i+j+k+m)} \leq 2^{r-1} \right\} \right| \\ &= \sum_{\pi=1}^{(d+1)^4-(d+1)^2} |\bar{B}_{\pi}| \end{aligned} \tag{21}$$

where $\bar{B}_{\pi}$ denotes the set of instances of $[\alpha]$'s, each of which has exactly $((d+1)^4 - (d+1)^2\pi)$ $\alpha_{i,j,k,m}$'s set to be zero and exactly $\pi$ $\alpha_{i,j,k,m}$'s set to be nonzero, and is defined

$$\begin{aligned}\bar{B}_{\pi} = \Big\{ & [\alpha] | \alpha_{i_1,j_1,k_1,m_1} 2^{\bar{\ell}(i_1+j_1+k_1+m_1)} + \cdots \\ & + \alpha_{i_\pi,j_\pi,k_\pi,m_\pi} 2^{\bar{\ell}(i_\pi+j_\pi+k_\pi+m_\pi)} \leq 2^{r-1}, \\ & 0 \leq i_\phi, m_\phi \leq d, 1 \leq j_\phi, k_\phi \leq d, \alpha_{i_\phi,j_\phi,k_\phi,m_\phi} \neq 0, \\ & (i_\phi, j_\phi, k_\phi, m_\phi) \neq (i_{\phi'}, j_{\phi'}, k_{\phi'}, m_{\phi'}), \\ & \phi, \phi' \in [1, \pi], \phi \neq \phi' \Big\}. \end{aligned} \tag{22}$$

Let $B_{\pi}$ be defined as

$$\begin{aligned}B_{\pi} = \Big\{ & [\alpha] | \alpha_{i_1,j_1,k_1,m_1} 2^{\bar{\ell}(i_1+j_1+k_1+m_1)} + \cdots \\ & + \alpha_{i_\pi,j_\pi,k_\pi,m_\pi} 2^{\bar{\ell}(i_\pi+j_\pi+k_\pi+m_\pi)} \leq 2^{r-1}, 0 \leq i_\phi, \\ & m_\phi \leq d, 1 \leq j_\phi, k_\phi \leq d, \alpha_{i_\phi,j_\phi,k_\phi,m_\phi} \in \{0, 1\}, \\ & (i_\phi, j_\phi, k_\phi, m_\phi) \neq (i_{\phi'}, j_{\phi'}, k_{\phi'}, m_{\phi'}), \\ & \phi, \phi' \in [1, \pi], \phi \neq \phi' \alpha_{i_1,j_1,k_1,m_1} + \cdots \\ & + \alpha_{i_\pi,j_\pi,k_\pi,m_\pi} = \pi \Big\}. \end{aligned} \tag{23}$$

$B_{\pi}$ can be regarded as the restricted version of $\bar{B}_{\pi}$; i.e., $\alpha_{i,j,k,m}$'s in $\bar{B}_{\pi}$ could be arbitrary elements in $\mathbb{F}_q$, whereas $\alpha_{i,j,k,m}$'s in $B_{\pi}$ are limited to be either 0 or 1. With such a definition of $B_{\pi}$, the lower bound of $|\mathfrak{F}'|$ can be derived as

$$|\mathfrak{F}'| > \sum_{\pi=1}^{(d+1)^4-(d+1)^2} |B_{\pi}|. \tag{24}$$

Taking $B_1 = \{[\alpha] | \alpha_{i_1,j_1,k_1,m_1} 2^{\bar{\ell}(i_1+j_1+k_1+m_1)}, 0 \leq i_1, m_1 \leq d, 1 \leq j_1, k_1 \leq d, \alpha_{i_1,j_1,k_1,m_1} \in \{0, 1\}, \}$ as an example, it turns out that $|B_1|$ can be equivalent to the number of instances $(i, j, k, m)$'s satisfying the constraint $\bar{\ell}(i+j+k+m) \leq r-1$ (or $i+j+k+m \leq \lfloor (r-1/\bar{\ell}) \rfloor$). Let $p(n)$ be the number of ways of partitioning an integer $n$ as a sum of integers. For example, $p(4) = 5$ because $4 = 4 = 3+1 = 2+2 = 2+1+1 = 1+1+1+1$. In our setting, $p(0)$ and $p(n)$, where $n$ is a negative integer, are assumed to be zero. With the use of $p(n)$, $|B_i|$ can be represented as

$$|B_i| = \sum_{\rho=1}^{\lfloor (r-1)/\bar{\ell} \rfloor} p\left(\rho - 1 - (i-1)\right).$$

As a whole, $|\mathfrak{F}'|$ is lower bounded by

$$\begin{aligned} & |B_1| + |B_2| + \cdots + |B_{(d+1)^4}| \\ & \quad = \sum_{\rho=1}^{\lfloor (r-1)/\bar{\ell} \rfloor} p(\rho - 1 - 0) + \sum_{\rho=1}^{\lfloor (r-1)/\bar{\ell} \rfloor} p(\rho - 1 - 1) + \cdots \\ & \qquad + \sum_{\rho=1}^{\lfloor (r-1)/\bar{\ell} \rfloor} p\left(\rho - 1 - \left((d+1)^4 - 1\right)\right) \\ & \quad = \sum_{i=1}^{(d+1)^4-(d+1)^2} \sum_{\rho=1}^{\lfloor (r-1)/\bar{\ell} \rfloor} p(\rho - i). \end{aligned}$$

No explicit closed-form formula of $p(n)$ exists. Nonetheless, as shown in [20], $p(n)$ can be approximated as

$$p(n) \cong \frac{1}{4n\sqrt{3}} e^{\pi\sqrt{\frac{2n}{3}}}.$$

As a result, the lower bound of $|\mathfrak{F}'|$ can be written as

$$|\mathfrak{F}'| \geq \sum_{i=1}^{(d+1)^4-(d+1)^2} \sum_{\rho=i+1}^{\lfloor (r-1)/\bar{\ell} \rfloor} \frac{1}{4(\rho-i)\sqrt{3}} e^{\pi\sqrt{\frac{2(\rho-i)}{3}}}. \tag{25}$$

In the procedure of obtaining this lower bound of $|\mathfrak{F}'|$, only the nonconstant polynomials are considered. Unfortunately, this is a loose lower bound of $|\mathfrak{F}'|$ because many possibilities of $[\alpha]$'s have been erased in the procedure of simplifying $|\bar{B}_{\pi}|$ as $|B_{\pi}|$ in (22) and (23). Thus, (25) could be useless in approximating the exact value of $|\mathfrak{F}'|$. Nonetheless, it definitely can be useful to demonstrate that $|\mathfrak{F}'|$ is sufficiently greater than zero so that the feasibility of our method can be guaranteed. For example, as the setting of $\bar{\ell} = 8$ and $r = 65$ is considered, $|\mathfrak{F}'|$ is lower bounded by 145. One may be concerned that the result of $|\mathfrak{F}'| \geq 145$ is not sufficiently large to achieve the resilience against the adversary randomly guessing the coefficients of $f(x, y, z, w)$. However, one should keep in mind that the lower bound in (25) is rather loose. Actually, from (22) and (23), one may observe that

$q \gg 2$ possibilities of $\alpha_{i_\phi,j_\phi,k_\phi,m_\phi}$, $\phi \in [1,\pi]$, in $\bar{B}_\pi$ are reduced to two possibilities of $\alpha_{i_\phi,j_\phi,k_\phi,m_\phi}\phi \in [1,\pi]$, in $B_\pi$, resulting in the significant underestimation of $|\mathfrak{F}'|$. Hence, as the setting of $\bar{\ell} = 8$ and $r = 65$ is considered, $|\mathfrak{F}'|$ should be substantially greater than 145. The more accurate approximation of $|\mathfrak{F}'|$ will be our future work.

## APPENDIX B
## DERIVATION OF ENERGY CONSUMPTION IN TABLE II

This Appendix describes how we derive the approximate energy consumption of different en-route filtering schemes shown in Table II. We refer the readers to the corresponding references for the explanation of the terms that are not explained in this paper.

When the scheme without filtering is used, intermediate nodes simply forward the message, resulting in $e^{\text{comp}}_{\text{Ord}} = 0$. Although $e^{\text{comm}}_{\text{SEF}}$ is given for SEF with the use of Bloom filter, we consider $e^{\text{comp}}_{\text{SEF}}$ as if the ordinary version of SEF is used and, therefore, $e^{\text{comp}}_{\text{SEF}} = (t/2)e_h$, where $e_h$ denotes the energy for calculating an MAC, because approximately $t/2$ keys from different partitions are considered. $e^{\text{comp}}_{\text{DEF}}$ is set to $c_1 e_h$, where $c_1$ is a random constant representing the distance between the current position in the key chain and the last position in the key chain.

IHA can guarantee that the false report can be detected within at most $t$ hops if $t$ MACs are used. Thus, the average number of hops the false report can travel is $t/2$, resulting in $e^{\text{comm}}_{\text{IHA}} = (192 + 64t)(H + (t/2))$. LBRS and LEDS can be regarded as the location-based variants of SEF and IHA, respectively. As the conception of key partition in SEF is used in LBRS in conjunction with the geographic locations of the cells, $e^{\text{comm}}_{\text{LBRS}}$ should be similar to $e^{\text{comm}}_{\text{SEF}}$ because each intermediate node checks the messages if it has an associated remote cell in which the MAC of the received message is constructed. Similar to IHA, LEDS establishes the security associations among nodes, except that the security associations are directly established among nodes in IHA but the security association is established according to the cells where the nodes locate. Thus, compared to $e^{\text{comm}}_{\text{IHA}}$, $e^{\text{comm}}_{\text{LEDS}}$ should include an additional term $c_3$ representing the average hop distance between two nodes within a cell. The probability of detecting the false report in GREF is approximately $(t - c_4/t)c_5$, where $c_4$ is the number of keys from distinct groups and $c_5$ is the key sharing probability. Thus, we can use this probability to derive $e^{\text{comm}}_{\text{GREF}}$ according to (18). Since different key groups are constructed in GREF and each intermediate node has to check the legitemacy of MACs constructed with the keys from different key groups, $e^{\text{comp}}_{\text{GREF}}$ is set to be $c_6 e_h$, where $c_6$ is the number of MACs from the same group where the node shares the key.

## ACKNOWLEDGMENT

Chun-Shien Lu is the corresponding author of this paper.

## REFERENCES

[1] T. C. Aysal and K. E. Barner, "Sensor data cryptography in wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 2, pp. 273–289, Jun. 2008.

[2] M. Albrecht, C. Gentry, S. Halevi, and J. Katz, "Attacking cryptographic schemes based on 'Perturbation Polynomials.'," in *Proc. ACM Conf. Computer and Communications Security (CCS)*, Chicago, IL, 2010.

[3] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *Proc. Int. Cryptology Conf. (CRYPTO)*, Santa Barbara, CA, 1993.

[4] M. Cagalj, S. Capkun, and J. P. Hubaux, "Wormhole-based antijamming techniques in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 1, pp. 100–114, Jan. 2007.

[5] M. Ceberio and V. Kreinovich, "Greedy algorithms for optimizing multivariate horner schemes," *ACM SIGSAM Bull.*, vol. 38, no. 1, pp. 8–15, 2004.

[6] S. Craver, N. Memon, B. L. Yeo, and M. M. Yeung, "Resolving rightful ownership with invisible watermarking techniques: Limitations, attacks, and implications," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 573–586, May 1998.

[7] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. IEEE Symp. Security and Privacy (S&P)*, Berkeley, CA, 2003.

[8] H. Chan, A. Perrig, and D. Song, "Secure hierarchical in-network aggregation in sensor networks," in *Proc. ACM Conf. Computer and Communications Security (CCS)*, Alexandria, VA, 2006.

[9] R. Cardell-Oliver, K. Smetten, M. Kranz, and K. Mayer, "A reactive soil moisture sensor network: Design and field evaluation," *Int. J. Distributed Sensor Netw.*, vol. 1, no. 2, pp. 149–162, 2005.

[10] S. A. Çamtepe and B. Yener, "Combinatorial design of key distribution mechanisms for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 15, no. 2, pp. 346–358, Apr. 2007.

[11] W. Du, J. Deng, Y. S. Han, and P. Varshney, "A key predistribution scheme for sensor networks using deployment knowledge," *IEEE Trans. Dependable Secure Comput.*, vol. 3, no. 2, pp. 62–77, Jan./Mar. 2006.

[12] A. Deshpande, C. Guestrin, W. Hong, and S. Madden, "Exploiting correlated attributes in acquisitional query processing," in *Proc. IEEE Int. Conf. Data Engineering (ICDE)*, Tokyo, Japan, 2005.

[13] J. Deng, R. Han, and S. Mishra, "Defending against path-based DoS attacks in wireless sensor networks," in *Proc. ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, Alexandria, VA, 2005.

[14] Q. Dong, D. Liu, and P. Ning, "Pre-authentication filters: Providing DoS resistance for signature-based broadcast authentication in wireless sensor networks," in *Proc. ACM Conf. Wireless Network Security (WiSec)*, Alexandria, VA, 2008.

[15] L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. Annu. ACM Computer and Communications Security (CCS)*, Washington, DC, 2002.

[16] V. C. Giruka, M. Singhal, J. Royalty, and S. Varanasi, "Security in wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 8, no. 1, pp. 1–24, Jan. 2008.

[17] H. Yu, "Secure and highly-available aggregation queries in large-scale sensor networks via set sampling," in *Proc. ACM/IEEE Int. Conf. Information Processing in Sensor Networks (IPSN)*, San Francisco, CA, 2009.

[18] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," *Special Issue on Wireless Sensor Networks, ACM Mobile Netw. Applicat.*, vol. 10, no. 4, pp. 519–528, 2005.

[19] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Wormhole attacks in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 370–380, Feb. 2006.

[20] G. H. Hardy and S. Ramanujan, "Asymptotic formulae in combinatory analysis," in *Proc. London Mathematical Society*, 1918, vol. 17, pp. 75–115.

[21] D. Kundur, W. Luh, U. N. Okorafor, and T. Zourntos, "Security and privacy for distributed multimedia sensor networks," *Proc. IEEE*, vol. 96, no. 1, pp. 112–130, Jan. 2008.

[22] C. Krauß, M. Schneider, K. Bayarou, and C. Eckert, "STEF: A secure ticket-based en-route filtering scheme for wireless sensor networks," in *Proc. Int. Conf. Availability, Reliability and Security (ARES)*, 2007.

[23] C. Krauß, M. Schneider, and C. Eckert, "Defending against false-endorsement-based dos attacks in wireless sensor networks," in *Proc. ACM Conf. Wireless Network Security (WiSec)*, Alexandria, VA, 2008.

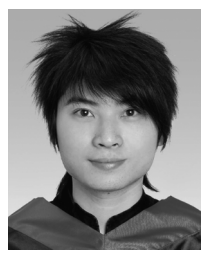
[24] M. Kutter, S. Voloshynovskiy, and A. Herrigel, "The watermark copy attack," in *Proc. SPIE Electronic Imaging, Security and Watermarking of Multimedia Contents II*, San Jose, CA, 2000, vol. 3971, pp. 371–380.

[25] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *Proc. IEEE Int. Workshop on Sensor Network Protocols and Applications*, Anchorage, AK, 2003.

[26] F. Liu, X. Cheng, L. Ma, and K. Xing, "SBK: A self-configuring framework for bootstrapping keys in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 7, pp. 858–868, Jul. 2008.
[27] C.-S. Lu and C.-Y. Hsu, "Anti-disclosure watermark giving multiple watermark embedding approaches resistance to estimation attack," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 454–467, Apr. 2007.
[28] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proc. ACM Cconf. Computer and Communications Security (CCS)*, Washington, DC, 2003.
[29] D. Liu and P. Ning, "Multi-level $\mu$TESLA: Broadcast authentication for distributed sensor networks," *ACM Trans. Embedded Comput. Syst. (TECS)*, vol. 3, no. 4, pp. 800–836, 2004.
[30] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Proc. ACM/IEEE Int. Conf. Information Processing in Sensor Networks (IPSN)*, St. Louis, MO, 2008.
[31] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Trans. Inf. Syst. Security*, vol. 8, no. 1, pp. 41–77, 2005.
[32] Z. Li, X. Zhu, Y. Lian, and Q. Sun, "Constructing secure content-dependent watermarking scheme using homomorphic encryption," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, Beijing, China, 2007.
[33] D. J. Malan, M. Walsh, and M. D. Smith, "Implementaing public-key infrastructure for sensor networks," *ACM Trans. Sensor Netw.*, vol. 4, no. 4, pp. 1–23, 2008.
[34] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar, "SPINS: Security protocols for sensor networks," in *Proc. Annu. ACM Int. Conf. Mobile Computing and Networking (MobiCom)*, Rome, Italy, 2001.
[35] K. Ren, W. Lou, and Y. Zhang, "LEDS: Providing location-aware end-to-end data security in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 5, pp. 585–598, May 2006.
[36] B. Sunar and D. Cyganski, "Comparison of Bit and Word Level Algorithms for Evaluating Unstructured Functions over Finite Rings," in *Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES)*, Edinburgh, U.K., 2005.
[37] A. Silberstein and J. Yang, "Many-to-many aggregation for sensor networks," in *Proc. IEEE Int. Conf. Data Engineering (ICDE)*, Istanbul, Turkey, 2007.
[38] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Commun. Surveys Tutorials*, vol. 8, no. 2, pp. 2–23, 2nd Quarter, 2006.
[39] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 640–652, Jun. 2006.
[40] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proc. IEEE PerCom*, Kauai, HI, 2005.
[41] W. Wang, V. Srinivasan, and K. Chua, "Coverage in hybrid mobile sensor networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 11, pp. 1374–1387, Nov. 2008.
[42] Y. Wang, X. Wang, B. Xie, D. Wang, and D. P. Agrawal, "Intrusion detection in homogeneous and heterogeneous wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 6, pp. 698–711, Jun. 2008.
[43] Z. Yu and Y. Guan, "A dynamic en-route filtering scheme for data reporting in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 150–163, Feb. 2010.
[44] H. Yang and S. Lu, "Commutative cipher based en-route filtering in wireless sensor networks," in *Proc. IEEE Vehicular Technology Conf. (VTC) 2004-Fall*, Los Angeles, CA, 2004.
[45] L. Yu and J. Li, "Grouping-based resilient statistical en-route filtering for sensor networks," in *Proc. IEEE Conf. Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, 2009.
[46] C.-M. Yu, C.-S. Lu, and S.-Y. Kuo, "Non-interactive pairwise key establishment for sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 556–569, Sep. 2010.
[47] C.-M. Yu, C.-S. Lu, and S.-Y Kuo, "A constrained function based authentication scheme for sensor networks," in *Proc. IEEE Wireless Communications & Networking Conf. (WCNC)*, Budapest, Hungary, 2009.
[48] C.-M. Yu, C.-S. Lu, and S.-Y. Kuo, "A DoS-resilient en-route filtering scheme for sensor networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, New Orleans, LA, 2009, (poster).
[49] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," in *Proc. IEEE Conf. Computer Communications (INFOCOM)*, Hong Kong, China, 2004.
[50] H. Yang, F. Ye, Y. Yuam, S. Lu, and W. Arbaugh, "Toward resilient security in wireless sensor networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, Urbana-Champaign, IL, 2005.
[51] T. Yuan, S. Zhang, Y. Zhong, and J. Ma, "KAEF: An en-route scheme of filtering false data in wireless sensor networks," in *Proc. IEEE Int. Performance Computing and Communications Conf. (IPCCC)*, Phoenix, AZ, 2008.
[52] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 247–260, Feb. 2006.
[53] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," in *Proc. ACM Conf. Computer and Communications Security (CCS)*, Washington, DC, 2003.
[54] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering false data in sensor networks," in *Proc. IEEE Symp. Security and Privacy (S&P)*, Berkeley, CA, 2004.
[55] W. Zhang, N. Subramanian, and G. Wang, "Lightweight and compromise-resilient message authentication in sensor networks," in *Proc. IEEE Conf. Computer Communications (INFOCOM)*, Phoenix, AZ, 2008.
[56] W. Zhang, H. Song, S. Zhu, and G. Cao, "Least privilege and privilege deprivation: Towards tolerating mobile sink compromises in wireless sensor networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, Urbana-Champaign, IL, 2005.

**Chia-Mu Yu** (S'09) is working toward the Ph.D. degree at the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan.

He was a research assistant at the Institute of Information Science, Academia Sinica, Taipei, Taiwan. He is currently a visiting scholar at Harvard University. His research interests include experimental and theoretical aspects of sensor network security.



**Yao-Tung Tsou** is currently working toward the Ph.D. degree at the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan.

He has been a research assistant at the Institute of Information Science, Academia Sinica, Taipei, Taiwan, since 2009. His research interests include sensor network security and applications.

**Chun-Shien Lu** (M'99) received the Ph.D. degree in electrical engineering from National Cheng-Kung University, Tainan, Taiwan, in 1998.

From October 1998 to July 2002, he joined Institute of Information Science, Academia Sinica, Taiwan, as a postdoctoral fellow for his military service. From August 2002 to July 2006, he was an assistant research fellow at the same institute. Since July 2006, he has been an associate research fellow. His current research interests mainly focus on various topics (including signal processing and security) of multimedia, sensor network security, and compressive sensing. He organized a special session on Multimedia Security in the 2nd and 3rd IEEE Pacific-Rim Conference on Multimedia, respectively (2001–2002). He co-organized two special sessions (in the area of media identification and DRM) in the 5th IEEE International Conference on Multimedia and Expo (ICME), 2004. He is a guest coeditor of the *Special Issue on Visual Sensor Network, EURASIP Journal on Applied Signal Processing*, in 2005. He serves as a Technical Committee Member of Multimedia Systems and Applications Technical Committee, IEEE Circuits and Systems Society, since 2007.

Dr. Lu holds two U.S. patents, two R.O.C. patents, and one Canadian patent in digital watermarking. He has received paper awards many times from the Image

Processing and Pattern Recognition Society of Taiwan for his work on data hiding. He won the Ta-You Wu Memorial Award, National Science Council in 2007 and was a corecipient of a National Invention and Creation Award in 2004. From July 2007, he served as a member of the Multimedia Systems and Applications Technical Committee of the IEEE Circuits and Systems Society. He is currently an Associate Editor of IEEE TRANSATIONS ON IMAGE PROCESSING. He is a member of the ACM.

**Sy-Yen Kuo** (S'85–M'88–SM'98–F'01) received the B.S. degree in electrical engineering from National Taiwan University, in 1979, the M.S. degree in electrical and computer engineering from the University of California at Santa Barbara, in 1982, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, in 1987.

He is a Distinguished Professor at the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, and was the Chairman at the same department from 2001 to 2004. He was a Chair Professor and Dean of the College of Electrical and Computer Engineering, National Taiwan University of Science and Technology from 2006 to 2009. He spent his sabbatical years as a Visiting Professor at the Computer Science and Engineering Department, the Chinese University of Hong Kong from 2004 to 2005, and as a visiting researcher at AT&T Labs-Research, New Jersey from 1999 to 2000, respectively. He was the Chairman of the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan, from 1995 to 1998, a faculty member in the Department of Electrical and Computer Engineering at the University of Arizona from 1988 to 1991, and an engineer at Fairchild Semiconductor and Silvar-Lisco, both in California, from 1982 to 1984. In 1989, he also worked as a summer faculty fellow at Jet Propulsion Laboratory of California Institute of Technology. His current research interests include dependable systems and networks, mobile computing, and quantum computing and communications. He has published more than 300 papers in journals and conferences, and also holds more than 10 U.S. and Taiwan patents.

Prof. Kuo received the distinguished research award between 1997 and 2005 consecutively from the National Science Council in Taiwan and is now a Research Fellow there. He was also a recipient of the Best Paper Award in the 1996 International Symposium on Software Reliability Engineering, the Best Paper Award in the simulation and test category at the 1986 IEEE/ACM Design Automation Conference (DAC), the National Science Foundation's Research Initiation Award in 1989, and the IEEE/ACM Design Automation Scholarship in 1990 and 1991.