BECAN: A Bandwidth-Efficient Cooperative Authentication Scheme for Filtering Injected False Data in Wireless Sensor Networks

Rongxing Lu, *Student Member*, *IEEE*, Xiaodong Lin, *Member*, *IEEE*, Haojin Zhu, *Member*, *IEEE*, Xiaohui Liang, *Student Member*, *IEEE*, and Xuemin (Sherman) Shen, *Fellow*, *IEEE*

Abstract—Injecting false data attack is a well known serious threat to wireless sensor network, for which an adversary reports bogus information to *sink* causing error decision at upper level and energy waste in en-route nodes. In this paper, we propose a novel bandwidth-efficient cooperative authentication (BECAN) scheme for filtering injected false data. Based on the random graph characteristics of sensor node deployment and the cooperative bit-compressed authentication technique, the proposed BECAN scheme can save energy by *early* detecting and filtering the majority of injected false data with minor extra overheads at the en-route nodes. In addition, only a very small fraction of injected false data needs to be checked by the *sink*, which thus largely reduces the burden of the *sink*. Both theoretical and simulation results are given to demonstrate the effectiveness of the proposed scheme in terms of high filtering probability and energy saving.

Index Terms—Wireless sensor network, injecting false data attack, random graph, cooperative bit-compressed authentication.

1 Introduction

Due to the fast booming of microelectro mechanical systems, wireless sensor networking has been subject to extensive research efforts in recent years. It has been well recognized as a ubiquitous and general approach for some emerging applications, such as environmental and habitat monitoring, surveillance and tracking for military [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. A wireless sensor network is usually composed of a large number of sensor nodes which are interconnected through wireless links to perform distributed sensing tasks. Each sensor node is low-cost but equipped with necessary sensing, data processing, and communicating components. Therefore, when a sensor node generates a report after being triggered by a special event, e.g., a surrounding temperature change, it will send the report to a data collection unit (also known as *sink*) through an established routing path [17].

Wireless sensor networks are usually deployed at unattended or hostile environments. Therefore, they are very vulnerable to various security attacks, such as selective forwarding, wormholes, and sybil attacks [12], [18]. In addition, wireless sensor networks may also suffer from injecting false data attack [10]. For an injecting false data attack, an adversary first compromises several sensor nodes, accesses all keying materials stored in the compromised nodes, and then controls these compromised nodes to inject bogus information and send the false data to the *sink* to cause upper-level error decision, as well as energy wasted in en-route nodes. For instance, an adversary could fabricate a wildfire event or report a wrong wildfire location information to the *sink*, then expensive resources will be wasted by sending rescue workers to a nonexisting or wrong wildfire location. Therefore, it is crucial to filter the false data as accurately as possible in wireless sensor networks. At the same time, if all false data are flooding into the *sink* simultaneously, then not only huge energy will be wasted in the en-route nodes, but also heavy verification burdens will undoubtedly fall on the *sink*. As a result, the whole network could be paralyzed quickly. Therefore, filtering false data should also be executed as early as possible to mitigate the energy waste. To tackle this challenging issue, some false data filtering mechanisms have been developed [7], [8], [9], [10], [11], [12], [13]. Since most of these filtering mechanisms use the symmetric key technique, once a node is compromised, it is hard to identify the node. In other words, the compromised node can abuse its keys to generate false reports, and the reliability of the filtering mechanisms will be degraded.

In this paper, we propose a novel bandwidth-efficient cooperative authentication (BECAN) scheme for filtering injected false data in wireless sensor networks. Compared with the previously reported mechanisms, the BECAN scheme achieves not only high filtering probability but also high reliability. The main contributions of this paper are threefold.

- First, we study the random graph characteristics of wireless sensor node deployment, and estimate the

- *R. Lu, X. Liang, and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada N2L 3G1. E-mail: {rxlu, x27liang, xshen}@bbcr.uwaterloo.ca.*
- *X. Lin is with the Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, ON, Canada L1H 7K4. E-mail: xiaodong.lin@uoit.ca.*
- *H. Zhu is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: zhu-hj@cs.sjtu.edu.cn.*

Manuscript received 2 June 2010; revised 3 Aug. 2010; accepted 12 Aug. 2010; published online 14 Mar. 2011.
Recommended for acceptance by A. Nayak.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2010-06-0326.
Digital Object Identifier no. 10.1109/TPDS.2011.95.

probability of k-neighbors, which provides the necessary condition for BECAN authentication;
- Second, we propose the BECAN scheme to filter the injected false data with cooperative bit-compressed authentication technique. With the proposed mechanism, injected false data can be early detected and filtered by the en-route sensor nodes. In addition, the accompanied authentication information is bandwidth-efficient; and
- Third, we develop a custom Java simulator to demonstrate the effectiveness of the proposed BECAN scheme in terms of en-routing filtering probability and false negative rate on true reports.

The remainder of this paper is organized as follows: In Section 2, we introduce the system model and design goal. In Section 3, we review some preliminaries including TinyECC-based noninteractive keypair establishment [19] and message authentication code in $\mathbb{Z}_{2^n}$. Then, we present the BECAN scheme in Section 4, followed by the security analysis and performance evaluation in Section 5 and Section 6, respectively. We review some related works in Section 7. In the end, we draw our conclusions in Section 8.

2 Model and Design Goal

In this section, we formulate the network model, the security model, and identify the design goal.

2.1 Network Model

We consider a typical wireless sensor network which consists of a *sink* and a large number of sensor nodes $\mathcal{N} = \{N_0, N_1, \ldots\}$ randomly deployed at a certain interest region (CIR) with the area $\mathcal{S}$. The *sink* is a trustable and powerful data collection device, which has sufficient computation and storage capabilities and is responsible for initializing the sensor nodes and collecting the data sensed by these nodes. Each sensor node $N_i \in \mathcal{N}$ is stationary in a location. For differentiation purpose, we assume each sensor node has a unique nonzero identifer. The communication is bidirectional, i.e., two sensor nodes within their wireless transmission range (R) may communicate with each other. Therefore, if a sensor node is close to the *sink*, it can directly contact the *sink*. However, if a sensor node is far from the transmission range of the *sink*, it should resort to other nodes to establish a route and then communicate with the *sink*. Formally, such a wireless sensor network, as shown in Fig. 1, can be represented as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots\}$ is the set of all sensors $\mathcal{N} = \{N_0, N_1, \ldots\}$ plus the *sink*, and $\mathcal{E} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}\}$ is the set of edges. Let $d(v_i, v_j)$ denote as the distance between v_i and v_j, then each e_{ij}, which indicates whether there exists a communication edge between two nodes v_i and v_j or not, is defined as

$$e_{ij} = \begin{cases} 1, & d(v_i, v_j) \le R; \\ 0, & d(v_i, v_j) > R. \end{cases} \tag{1}$$

Let v_1 denote the *sink*. All sensor nodes $\mathcal{V}/\{v_1\} = \{v_2, v_3, \ldots\}$ can run the Dijkstra shortest path algorithm (see Appendix) to find their shortest paths to the *sink* v_1, only if the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is fully connected.

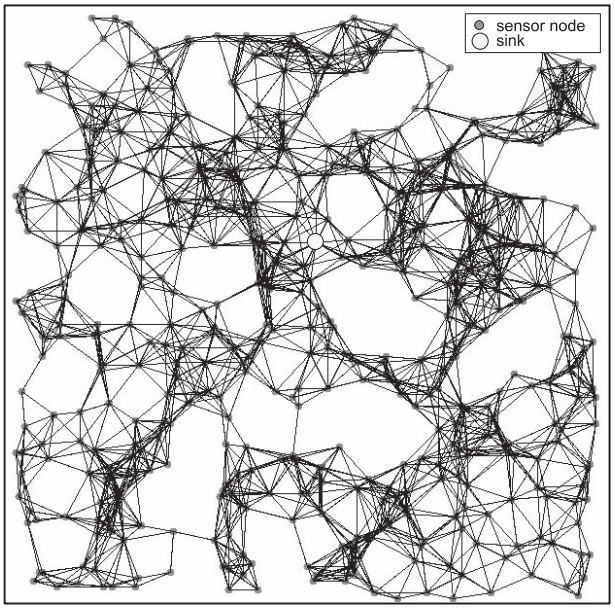


Fig. 1. Wireless sensor network under consideration.

Probability of fully connected $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Assume that the positions of these vertexes $\mathcal{V} = \{v_0, v_1, \ldots\}$ are uniformly distributed in the area $\mathcal{S}$ with network density λ, where $\lambda = \frac{|\mathcal{V}|}{\mathcal{S}}$, and $|\mathcal{V}|$ denotes the cardinality of $\mathcal{V}$. Based on the random graph theory, the probability that there are n nodes in an arbitrary region A with the area $\mathcal{A}$ is

$$\begin{aligned} P(N = n|\mathcal{A}) &= \binom{|\mathcal{V}|}{n} \left(\frac{\lambda \cdot \mathcal{A}}{|\mathcal{V}|}\right)^n \cdot \left(1 - \frac{\lambda \cdot \mathcal{A}}{|\mathcal{V}|}\right)^{|\mathcal{V}|-n} \\ &= \binom{|\mathcal{V}|}{n} \left(\frac{\mathcal{A}}{\mathcal{S}}\right)^n \cdot \left(1 - \frac{\mathcal{A}}{\mathcal{S}}\right)^{|\mathcal{V}|-n}. \end{aligned} \tag{2}$$

To calculate the full connection probability P_{con}, we first compute P_{iso}, the isolation probability of any node in $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where a node is called *isolated* if there exists no link between it and any other nodes. In other words, in some circle coverage with the area πR^2, except one node lies at the center, no other node exists. If the border effects are neglected, we have

$$\begin{aligned} P_{\text{iso}} &= P(N = 0|\pi R^2) \\ &= \binom{|\mathcal{V}| - 1}{0} \cdot \left(\frac{\pi R^2}{\mathcal{S}}\right)^0 \cdot \left(1 - \frac{\pi R^2}{\mathcal{S}}\right)^{|\mathcal{V}|-1} \\ &= \left(1 - \frac{\pi R^2}{\mathcal{S}}\right)^{|\mathcal{V}|-1}. \end{aligned} \tag{3}$$

Based on the isolation probability P_{iso}, we can compute the full connection probability P_{con} [20] as

$$\begin{aligned} P_{\text{con}} &\ge (1 - P_{\text{iso}})^{|\mathcal{V}|} \\ &= \left(1 - \left(1 - \frac{\pi R^2}{\mathcal{S}}\right)^{|\mathcal{V}|-1}\right)^{|\mathcal{V}|}. \end{aligned} \tag{4}$$

Fig. 2 shows the full connection probability P_{iso} versus different transmission range R and $|V|$. It can be seen that the expected fully connected $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be achieved by choosing proper R and $|V|$.

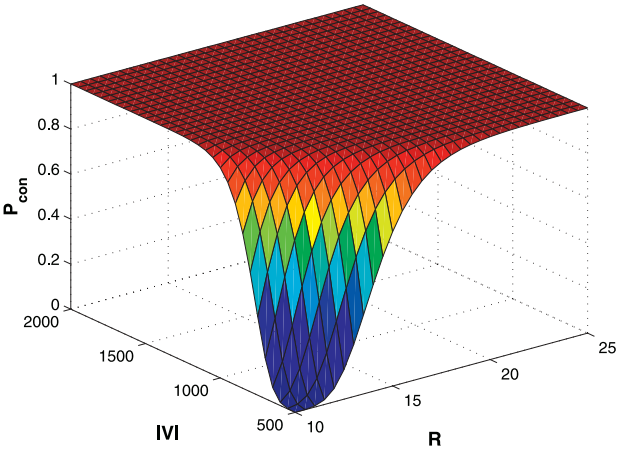


Fig. 2. Probability of fully connected $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{S} = 200 \times 200\ \mathrm{m}^2$, $500 \leq |\mathcal{V}| \leq 2{,}000$, and $10\ \mathrm{m} \leq R \leq 25\ \mathrm{m}$.

2.2 Security Model

Since a wireless sensor network is unattended, a malicious adversary may readily launch some security attacks to degrade the network functionalities. In addition, due to the low-cost constraints, sensor nodes $\mathcal{N} = \{N_0, N_1, \ldots\}$ are not equipped with expensive tamper-proof device and could be easily compromised in such an unprotected wireless sensor network. Therefore, in our security model, we assume an adversary $\mathcal{A}$ can compromise a fraction of sensor nodes and obtain their stored keying materials. Then, after being controlled and reprogrammed by the adversary $\mathcal{A}$, these compromised sensor nodes can collude to launch some injected false data attacks.

Since our work focuses on filtering injected false data attack, other attacks launched by the compromised sensor nodes in wireless sensor network, such as building bogus routing information, selectively dropping true data packet, and creating routing loops to waste the energy of network [18], are not addressed in this paper.

2.3 Design Goal

The design goal is to develop an efficient cooperative bandwidth-efficient authentication scheme for filtering the injected false data. Specifically, the following two desirable objectives will be achieved.

2.3.1 Early Detecting the Injected False Data by the En-Route Sensor Nodes

The *sink* is a powerful data collection device. Nevertheless, if all authentication tasks are fulfilled at the *sink*, it is undoubted that the *sink* becomes a bottleneck. At the same time, if too much injected false data floods into the *sink*, the *sink* will surely suffer from the Denial of Service (DoS) attack. Therefore, it is critical to share the authentication tasks with the en-route sensor nodes such that the injected false data can be detected and discarded early. The earlier the injected false data is detected, the more energy can be saved in the whole network.

2.3.2 Achieving Bandwidth-Efficient Authentication

Since the sensor nodes are low-cost and energy constraint, it is desirable to design a bandwidth efficient authentication scheme.

3 Preliminaries

3.1 TinyECC-Based Noninteractive Keypair Establishment

TinyECC is a configurable library for Elliptic Curve Cryptography (ECC), which allows flexible integration of ECC-based public key cryptography in sensor network applications. A substantially experimental evaluation using representative sensor platforms, such as MICAz [21] and Imote2 [22], is performed, and the results show that the *ready-to-use* TinyECC is suitable for wireless sensor networks to provide convenient authentications and pairkey establishments [19]. Let p be a large prime and $\mathbf{E}(\mathbb{F}_p)$ represent an elliptic curve defined over $\mathbb{F}_p$. Let $G \in \mathbf{E}(\mathbb{F}_p)$ be a base point of prime order q. Then, each sensor node $N_i \in \mathcal{N}$ can preload a TinyECC based public-private key pair (Y_i, x_i), where the private key x_i is randomly chosen from $\mathbb{Z}_q^*$ and the public key $Y_i = x_i G$.

Noninteractive keypair establishment. For *any* two sensor nodes $v_i, v_j \in \mathcal{G} = (\mathcal{V}, \mathcal{E})$, no matter what $e_{ij} \in \{0, 1\}$ is, sensor nodes v_i with the key pair (Y_i, x_i) and v_j with the key pair (Y_j, x_j) can establish a secure Elliptic Curve Diffie-Hellman (ECDH) keypair without direct contacting [23], where

$$k_{ij} = x_i Y_j = x_i x_j G = x_j x_i G = x_j Y_i = k_{ji}. \tag{5}$$

Because of the hardness of Elliptic Curve Discrete Logarithm (ECDL) problem, only v_i and v_j can secretly share a key. At the same time, the established keys are independent. In other words, if a sensor node v_i is compromised, then the key k_{ij} shared between v_i and v_j will be disclosed. However, the key $k_{jj'}$ shared between v_j and another sensor node $v_{j'}$ is not affected. For unattended wireless sensor networks, the property of *key independence* is useful, since it can limit the scope of key disclosure to the adversary $\mathcal{A}$.

3.2 Message Authentication Code in $\mathbb{Z}_{2^n}$

Message authentication code (MAC) provides assurance to the recipient of the message which came from the expected sender and has not been altered in transit [24]. Let $h(\cdot)$ be a secure cryptographic hash function [25]. A MAC in $\mathbb{Z}_{2^n}$ can be considered as a keyed hash, and defined as

$$MAC(m, k, n) = h(m \| k) \bmod 2^n, \tag{6}$$

where m, k, n are a message, a key, and an adjustable parameter, respectively. When $n = 1$, $MAC(m, k, 1)$ provides one-bit authentication, which can filter a false message with the probability $\frac{1}{2}$; while $n = \alpha$, $MAC(m, k, \alpha)$ can filter a false message with a higher probability $1 - \frac{1}{2^\alpha}$.

4 Proposed BECAN Scheme

In this section, we will propose BECAN scheme for filtering injected false data in wireless sensor networks. Before proceeding the BECAN scheme, the design rationale is introduced.

4.1 Design Rationale

To filter the false data injected by compromised sensor nodes, the BECAN adopts cooperative neighbor $\times$ router (CNR)-based filtering mechanism. As shown in Fig. 3, in the

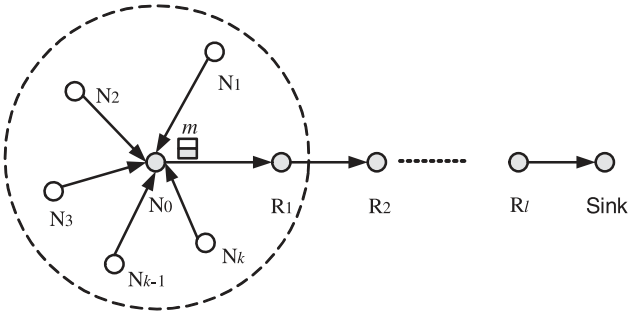


Fig. 3. Cooperative CNR-based authentication mechanism.

CNR-based mechanism, when a source node N_0 is ready to send a report m to the *sink* via an established routing path $R_{N_0} : [R_1 \rightarrow R_2 \rightarrow \cdots \rightarrow R_l \rightarrow Sink]$, it first resorts to its k neighboring nodes $N_{N_0} : \{N_1, N_2, \ldots, N_k\}$ to cooperatively authenticate the report m, and then sends the report m and the authentication information **MAC** from $N_0 \cup N_{N_0}$ to the *sink* via routing R_{N_0}, where

$$\mathbf{MAC} = \begin{pmatrix} mac_{01} & \cdots & mac_{0l} & mac_{0s} \\ mac_{11} & \cdots & mac_{1l} & mac_{1s} \\ mac_{21} & \cdots & mac_{2l} & mac_{2s} \\ \vdots & \vdots & \vdots & \vdots \\ mac_{k1} & \cdots & mac_{kl} & mac_{ks} \end{pmatrix}, \tag{7}$$

each mac_{ij}, $0 \le i \le k, 1 \le j \le l$, represents N_i's MAC on m for R_j's authentication, and each mac_{is} represents N_i's MAC on m for the *sink*'s authentication. As indicated in network model, the *sink* initializes all sensor nodes, then each sensor node shares its private key with the *sink*. At the same time, according to the TinyECC-based noninteractive keypair establishment [19], the full bipartite key graph between $N_0 \cup N_{N_0}$ and R_{N_0} can be established, as shown in Fig. 4. Because of the existence of full bipartite key graph, the **MAC** design is reasonable. Therefore, when a compromised sensor node sends a false data to the *sink*, the false data can be filtered if there is at least one uncompromised neighboring node participating in the reporting. To achieve the bandwidth-efficient authentication, each mac_{ij} is set as one bit and each mac_{is} is α bits by using the above MAC in $\mathbb{Z}_{2^n}$ technique. Then, the scale of **MAC** is only $(l + \alpha) \times (k + 1)$ bits.

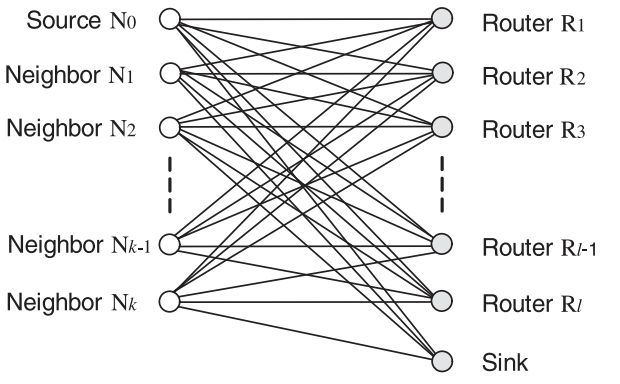


Fig. 4. Bipartite graph representing the relationships between the (source + neighbors) and (sink + routers).

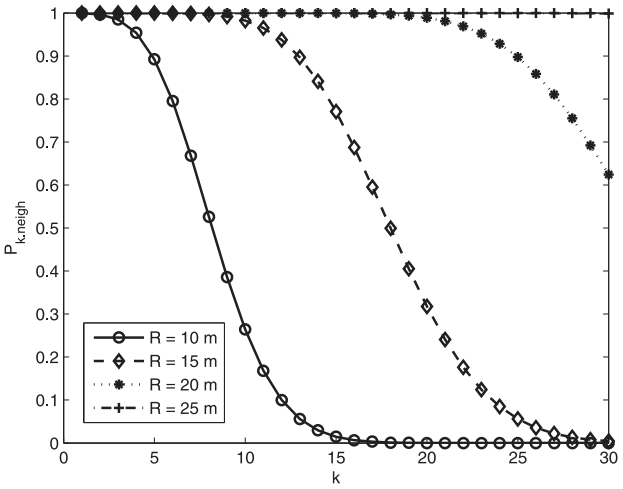


Fig. 5. Probability of k neighbors $P_{\text{k·neigh}}$ with $\mathcal{S} = 200 \times 200\ \text{m}^2$, $|V| = 1{,}000$, $R = 10, 15, 20, 25$ m, and $1 \le k \le 7$.

Probability of k neighbors. In the cooperative CNR-based authentication, if the number of the neighbors of the source node is less than a preset threshold k, the **MAC** authentication does not work. Let $P_{\text{k·neigh}}$ denote the probability that there are at least k neighbors in the transmission range of a source node, then

$$\begin{aligned} P_{\text{k·neigh}} &= P(N \ge k | \pi R^2) \\ &= 1 - P(N < k | \pi R^2) \\ &= 1 - \sum_{j=0}^{k-1} P(N = j | \pi R^2) \\ &= 1 - \sum_{j=0}^{k-1} \binom{|\mathcal{V}| - 1}{j} \cdot \left(\frac{\pi R^2}{\mathcal{S}}\right)^j \cdot \left(1 - \frac{\pi R^2}{\mathcal{S}}\right)^{|\mathcal{V}|-j-1}. \end{aligned} \tag{8}$$

Fig. 5 shows the probability $P_{\text{k·neigh}}$ in a parameterized wireless sensor network with different k, $(1 \le k \le 30)$. It can be seen the expected high probability can be achieved when choosing a proper k, i.e., $k \le 6$. As a result, the CNR-based **MAC** authentication mechanism is feasible.

4.2 Description of BECAN Authentication

The BECAN authentication scheme consists of two phases: sensor nodes initialization and deployment, and sensed results reporting protocol.

4.2.1 Sensor Nodes Initialization and Deployment

Given the security parameter κ, the *sink* first chooses an elliptic curve $(\mathbf{E}(\mathbb{F}_p), G, q)$ defined over $\mathbb{F}_p$, where p is a large prime and $G \in \mathbf{E}(\mathbb{F}_p)$ is a base point of prime order q with $|q| = \kappa$. Then, the *sink* selects a secure cryptographic hash function $h()$, where $h : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$. Finally, the *sink* sets the public parameters as $params = \{\mathbf{E}(\mathbb{F}_p), G, q, h()\}$.

To initialize sensor nodes $\mathcal{N} = \{N_0, N_1, N_2, \ldots, \}$, the *sink* invokes the Algorithm 1. Then, the *sink* deploys these initialized sensor nodes at a CIR in various ways, such as by air or by land. Given the rich literature in wireless sensor node deployment [26], [27], we do not address the deployment in detail. Without loss of generality, we assume that all sensor nodes are uniformly distributed in CIR after

deployment. When these sensor nodes are not occupied by the reporting task, they cooperatively establish or adjust their routing to the sink either a shortest path or a path adapted to some resource constrains with some existing routing protocol. Note that, the established routing path can accelerate the reporting. Once an event occurs, a report can be immediately relayed along the established routing path.

Algorithm 1. Sensor Nodes Initialization Algorithm
1: **Procedure** SENSORNODESINITIALIZATION
Input: $params$ and un-initialized $\mathcal{N} = \{N_0, N_1, N_2, \ldots\}$
Output: initialized $\mathcal{N} = \{N_0, N_1, N_2, \ldots\}$
2: **for** each sensor node $N_i \in \mathcal{N}$ **do**
3: preload N_i with TinyECC, *params* and energy
4: choose a random number $x_i \in \mathbb{Z}_q^*$ as the private key, compute the public key $Y_i = x_i G$, and install (Y_i, x_i) in N_i
5: **end for**
6: **return** initialized $\mathcal{N} = \{N_0, N_1, N_2, \ldots, N_n\}$
7: **end procedure**

4.2.2 Sensed Results Reporting Protocol

When a sensor node generates a report m after being triggered by a special event, e.g., a temperature change or in response to a query from the *sink*, it will send the report to the *sink* via an established routing. Assume that, the sensor (source) node N_0 has sensed some data m and is ready to report m to the *sink* via the routing path $R_{N_0} : [R_1 \rightarrow R_2 \rightarrow \cdots \rightarrow R_l \rightarrow Sink]$, as shown in Fig. 3, the following protocol steps will be executed:

Step 1. The source node N_0 gains the current timestamp T, chooses k neighboring nodes $N_{N_0} : \{N_1, N_2, \ldots, N_k\}$, and sends the event (m, T) and routing R_{N_0} to N_{N_0}.

Step 2. With (m, T, R_{N_0}) as input, each sensor node $N_i \in (N_{N_0} \cup \{N_0\})$ invokes the Algorithm 2 to generate a row authentication vector

$$Row_i = (mac_{i1}, mac_{i2}, \ldots, mac_{il}, mac_{is}), \tag{9}$$

and reports Row_i to the source node N_0.

Algorithm 2. CNR Based MAC Generation
1: **procedure** CNRBASEDMACGENERATION
Input: $params$, $N_i \in (N_{N_0} \cup N_0)$, m, T, R_{N_0}
Output: Row_i
2: N_i uses the non-interactive keypair establishment to compute shared keys with each node in $R_{N_0} : [R_1 \rightarrow R_2 \rightarrow \cdots \rightarrow R_l \rightarrow Sink]$ as $k_{i1}, k_{i2}, \ldots, k_{il}, k_{is}$, where k_{is} is N_i's private key distributed by the *sink*
3: **if** N_i believes the report m is true **then** ▷ a neighboring node is assumed having the same ability to detect a true event as the source node and correctly judge the report m.
4: **for** $j = 1$ to l **do**
5: $mac_{ij} = MAC(m \| T, k_{ij}, 1)$
6: **end for**
7: $mac_{is} = MAC(m \| T, k_{is}, \alpha)$
8: **else**
9: **for** $j = 1$ to l **do**
10: mac_{ij} is set as a random bit
11: **end for**
12: mac_{is} is set as a random bit string of length α
13: **end if**
14: **return** $Row_i = (mac_{i1}, mac_{i2}, \ldots, mac_{il}, mac_{is})$
15: **end procedure**

Step 3. After the source node N_0 aggregates all row vectors $(Row_0, Row_1, \ldots, Row_k)$, it formats the authentication information $\mathbf{MAC}$ as

$$\mathbf{MAC} = \begin{pmatrix} Row_0 \\ Row_1 \\ Row_2 \\ \vdots \\ Row_k \end{pmatrix} = \begin{pmatrix} mac_{01} & \cdots & mac_{0l} & mac_{0s} \\ mac_{11} & \cdots & mac_{1l} & mac_{1s} \\ mac_{21} & \cdots & mac_{2l} & mac_{2s} \\ \vdots & \vdots & \vdots & \vdots \\ mac_{k1} & \cdots & mac_{kl} & mac_{ks} \end{pmatrix}, \tag{10}$$

and reports $(m, T, \mathbf{MAC})$ as well as N_{N_0} to the *sink* along the routing R_{N_0}.

4.2.3 En-Routing Filtering

When each sensor node R_i, $(1 \leq i \leq l)$, along the routing R_{N_0} receives $(m, T, \mathbf{MAC})$ from its upstream node, it checks the integrity of the message m and the timestamp T. If the timestamp T is out of date, the message $(m, T, \mathbf{MAC})$ will be discarded. Otherwise, R_i invokes the Algorithm 3. If the returned value is "*accept*," R_i will forward the message $(m, T, \mathbf{MAC})$ to its downstream node, Otherwise, $(m, T, \mathbf{MAC})$ will be discarded.

Algorithm 3. CNR Based MAC Verification
1: **procedure** CNRBASEDMACVERIFICATION
Input: $params$, $R_j \in \{R_1, \ldots, R_l\}$, m, T, N_{N_0}
Output: *accept* or *reject*
2: R_j uses the noninteractive keypair establishment to compute shared keys with each node in $\{N_0, N_1, \ldots, N_k\}$ as $k_{0j}, k_{1j}, \ldots, k_{kj}$
3: set returnvalue = "*accept*"
4: **for** $i = 0$ to k **do**
5: $\overline{mac}_{ij} = MAC(m \| T, k_{ij}, 1)$
6: **if** $\overline{mac}_{ij} \oplus mac_{ij} \neq 0$ **then**
7: set returnvalue = "*reject*"
8: **break**
9: **end if**
10: **end for**
11: **return** returnvalue
12: **end procedure**

4.2.4 Sink Verification

If the *sink* receives the report $(m, T, \mathbf{MAC})$, it checks the integrity of the message m and the timestamp T. If the timestamp is out of date, the report $(m, T, \mathbf{MAC})$ will be immediately discarded. Otherwise, the *sink* looks up all private keys k_{is} of N_i, $0 \leq i \leq k$, and invokes the Algorithm 4. If the returned value of Algorithm 4 is "*accept*," the *sink* accepts the report m; otherwise, the *sink* rejects the report.

Algorithm 4. Sink Verification
1: **procedure** SINKVERIFICATION
Input: $params$, $k_{0s}, k_{1s}, \ldots, k_{ks}$, m, T
Output: *accept* or *reject*

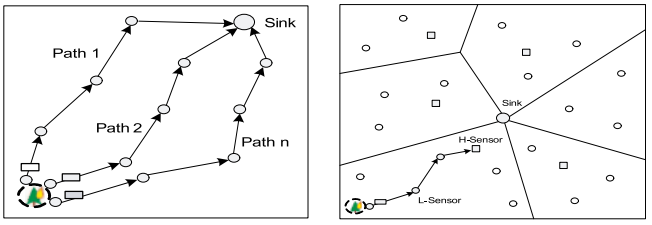


Fig. 6. Reliability and Scalability of the BECAN scheme. (a) Reliability with multireports. (b) Scalability with heterogenous deployment.

2: set returnvalue = "*accept*"
3: **for** $i = 0$ to k **do**
4: $\overline{mac}_{is} = MAC(m\|T, k_{is}, \alpha)$
5: **if** $\overline{mac}_{is} \oplus mac_{is} \neq 0$ **then**
6: set returnvalue = "*reject*"
7: **break**
8: **end if**
9: **end for**
10: **return** returnvalue
11: **end procedure**

Reliability and scalability. For the BECAN scheme, once a compromised sensor node participates in the report confirmation, the report will be polluted and cannot reach the *sink*. To improve the reliability, multireports solution is naturally introduced in the BECAN scheme. As shown in Fig. 6a, once a true wildfire event occurs, multisource nodes close to the event independently choose k different neighbors, produce the multireports and send them to the *sink* via different paths. Only if one report reaches the *sink*, the true event will successfully reported. As a result, the reliability of the BECAN scheme can be improved. In the BECAN scheme, the additional $(l+\alpha)\times(k+1)$ authentication bits are in linear with the length of the path l. If l is too long, the authentication bits become large. To resolve the scalability issue, we can devise a large-scale sensor network into a heterogenous sensor network [28], where each partition consists of a powerful High-end sensor (H-sensor) and a number of Low-end sensors (L-sensors), as shown in Fig. 6b. Each H-sensor serves as a cluster header. When a L-sensor senses some event, it can report to the nearby H-sensor, but not to the remote sink. Therefore, the heterogenous deployment can provide a good solution to the scalability issue of BECAN scheme.

5 Security Analysis

In this section, we analyze the security of the BECAN authentication scheme with respect to our main design goal, i.e., the effectiveness of filtering the injected false data.

5.1 Theoretical Analysis

Since the timestamp T is embedded in the report, the replay attack, a special injecting false data attack, can be filtered obviously. Therefore, how the BECAN scheme is resistent to the generic injecting false data attack will be studied here. Because the adversary $\mathcal{A}$ can compromise some sensor nodes in the network, without loss of generality, we assume the compromised probability for each sensor node is ρ, and study the filtering probability.

Let a compromised sensor node N_0 be ready to report an injected false data m^* with a valid timestamp T^* to the *sink*. According to the protocol, N_0 should select k neighboring sensor nodes to generate the authentication information $\mathbf{MAC}$ together, and then send $(m^*, T^*, \mathbf{MAC})$ to the *sink* via the routing $R_{N_0} : [R_1 \rightarrow R_2 \rightarrow \cdots \rightarrow R_l \rightarrow Sink]$. In the selected k neighboring sensor nodes $N_{N_0} : \{N_1, N_2, \ldots, N_k\}$, as we know, with the probability $\binom{k}{i}\rho^i(1-\rho)^{k-i}$, there are i compromised nodes. At the same time, in the routing R_{N_0}, with the probability $\binom{l}{j}\rho^j(1-\rho)^{l-j}$, there are j compromised nodes among l routing nodes. Because all keys are *key-independence*, then in order to pass the false data $(m^*, T^*, \mathbf{MAC})$ to the BECAN authentication, the sensor node N_0 must correctly guess all authentication bits between $k-i$ uncompromised neighboring nodes and $l-j$ uncompromised routing nodes plus the *sink*. Therefore, the guess probability is

$$Pr = P_{\text{k·neigh}} \cdot \binom{k}{i}\rho^i(1-\rho)^{k-i} \cdot \binom{l}{j}\rho^j(1-\rho)^{l-j} \cdot \frac{1}{2^{(k-i)(l+\alpha-j)}}. \tag{11}$$

Then, the false positive authentication probability is

$$\text{FPA} = \sum_{i=0}^{k}\sum_{j=0}^{l} P_{\text{k·neigh}} \cdot \binom{k}{i}\binom{l}{j}\rho^{i+j}(1-\rho)^{k+l-i-j} \cdot \frac{1}{2^{(k-i)(l+\alpha-j)}}. \tag{12}$$

Furthermore, we can obtain the filtering probability under this circumstance as

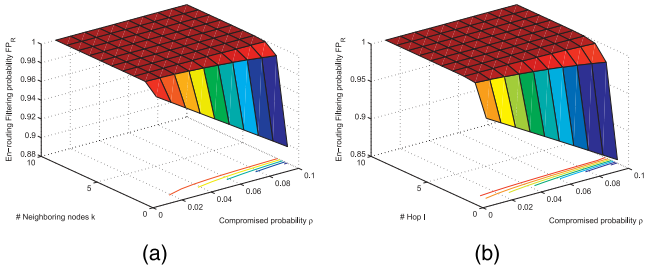
$$\text{FP} = 1 - \text{FPA} = 1 - P_{\text{k·neigh}} \cdot \sum_{i=0}^{k}\sum_{j=0}^{l}\binom{k}{i}\binom{l}{j}\rho^{i+j} (1-\rho)^{k+l-i-j} \cdot \frac{1}{2^{(k-i)(l+\alpha-j)}}. \tag{13}$$

When $\alpha = 0$, FP is rewritten as

$$\text{FP}_R = 1 - P_{\text{k·neigh}} \cdot \sum_{i=0}^{k}\sum_{j=0}^{l}\binom{k}{i}\binom{l}{j}\rho^{i+j} (1-\rho)^{k+l-i-j} \cdot \frac{1}{2^{(k-i)(l-j)}}. \tag{14}$$

which represents the en-routing filtering probability of the BECAN scheme and measures how much injected false data can be filtered as early as possible before their reaching the *sink*, in such a way the energy waster can be reduced, and the *sink* can avert the DoS attack due to large number of injected false data.

Fig. 7 plots how the en-routing filtering probability FP_R varies with the number of neighboring node k, the number of en-routing nodes l, and the compromised probability ρ. From the figure, when k and l are properly set, FP_R approaches to 1 in theory. However, in reality, when an experienced and astute adversary $\mathcal{A}$ launches an attack, it may first choose those compromised nodes as its neighbors participating in the injecting false data attack to increase the success probability, then the FP_R would be reduced. Therefore, it is of interest to use simulation to evaluate the en-routing filtering probability FP_R of the BECAN scheme.



(a) (b)

Fig. 7. The en-routing filtering probability FP_R as the functions of the number of neighboring nodes k and the compromised probability ρ, and the number of en-routing node l and the compromised probability ρ. (a) FP_R versus k and ρ, (b) FP_R versus l and ρ.

5.2 Simulation-Based En-Routing Filtering Evaluation

In the simulation, the en-routing filtering probability can be tested as

$$\mathrm{FP}_R = \frac{\text{number of false data filtered by en-route nodes}}{\text{total number of false data}}. \quad (15)$$

In what follows, we provide the simulation results for FP_R.

5.2.1 Simulation Settings

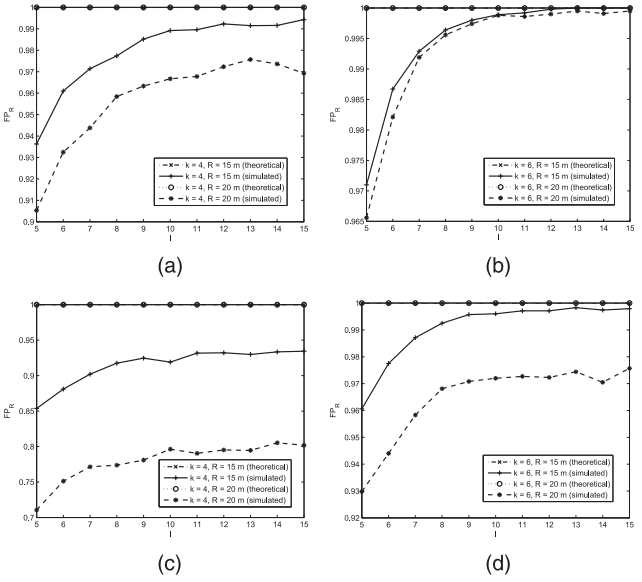
We study FP_R of the BECAN scheme using a simulator built in Java. In the simulations, 1,000 sensor nodes with a transmission range R are randomly deployed in a CIR of region $200 \times 200\ \mathrm{m}^2$ interest region. We consider each sensor node could be compromised with the probability ρ. In Table 1, we list the simulation parameters. Then, we test the networks when the number of en-routing nodes in the interest areas is varied from 5 to 15 in increment of 1. For each case, 10,000 networks are randomly generated, and the average of en-routing filtering probabilities over all of these randomly sampled networks are reported.

5.2.2 Simulation Results

Fig. 8 shows the en-routing filtering probability FP_R in terms of different number of en-routing nodes. As the number of routing nodes increases, FP_R increases. At the same time, by choosing more neighboring nodes involved in the protocol, i.e., the parameter k increases, FP_R will further increase, even the compromised probability is 5 percent. Further observing the FP_R with different transmission range R, we can see a relatively low FP_R for $R = 20$ m compared with that for $R = 15$ m. The reason is that, under the same settings, when the transmission range increases, the number of compromised neighboring nodes will also increase, so the experienced and astute $\mathcal{A}$ has more chances to choose more

TABLE 1
Parameter Settings

Parameter	Value
Simulation area	200m × 200m
Number of sensor nodes	1000
Transmission range R	15m, 20m
Compromised probability ρ	2%, 5%
# neighboring nodes k	4, 6
# routing nodes l	$5, \cdots, 15$



(a) (b) (c) (d)

Fig. 8. En-routing filtering probability FP_R versus the different number of routing nodes l, where $5 \le l \le 15$. (a) $\rho = 2\%, k = 4$, (b) $\rho = 2\%, k = 6$, (c) $\rho = 5\%, k = 4$, and (d) $\rho = 5\%, k = 6$.

compromised nodes participating in the attack to increase the success attack probability. Based on these observations, we have the following theorem.

Theorem 1. *The BECAN scheme can effectively resist the injecting false data attack launched by the experienced and astute $\mathcal{A}$, only if the number of compromised nodes in the transmission range R is less than the security parameter k.*

Proof. From (13)-(14), we have the following relationship between FP and FP_R, i.e.,

$$\mathrm{FP} = 1 - (1 - \mathrm{FP}_R) \cdot \frac{1}{2^\alpha}, \quad (16)$$

where $1 - \mathrm{FP}_R$ is the success probability of injecting false attack escaping from the en-routing filtering, which consists of two parts: 1) $\mathrm{FPA}_{|N_c=k}$, the false positive probability when the number of participating neighboring compromised nodes $N_c = k$ in the attack, and 2) $\mathrm{FPA}_{|N_c<k}$, the false positive probability when $N_c < k$. Therefore, we have

$$\mathrm{FP} = 1 - (\mathrm{FPA}_{|N_c=k} + \mathrm{FPA}_{|N_c<k}) \cdot \frac{1}{2^\alpha}. \quad (17)$$

When the parameter α is well chosen, the item $\mathrm{FPA}_{|N_c<k} \cdot \frac{1}{2^\alpha} \to 0$. However, $\frac{1}{2^\alpha}$ does not affect $\mathrm{FPA}_{|N_c=k}$, since all participating t neighboring nodes are compromised. Thus, we have $\mathrm{FP} = 1 - \mathrm{FPA}_{|N_c=k}$. Because the condition $N_c = k$ is determined by the number of compromised nodes in the transmission range R, if this condition does not hold, $\mathrm{FP} = 1 - \mathrm{FPA}_{|N_c=k} = 1$. Therefore, only if the number of compromised nodes in the transmission range R is less than the parameter k, the BECAN scheme can effectively resist the injecting false data attack launched by the experienced and astute $\mathcal{A}$. □

Fig. 9 also shows the filtering ratio at each en-routing node R_i in R_{N_0}, where $1 \le i \le 10$. The results confirm our design

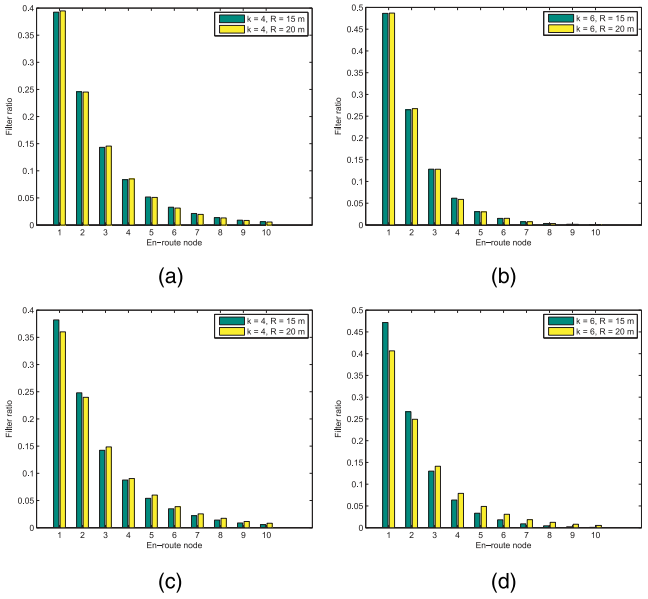


Fig. 9. The filtering ratio at each routing node R_i in R_{N_0}, where $1 \leq i \leq 10$. (a) $\rho = 2\%, k = 4$, (b) $\rho = 2\%, k = 6$, (c) $\rho = 5\%, k = 4$, and (d) $\rho = 5\%, k = 6$.

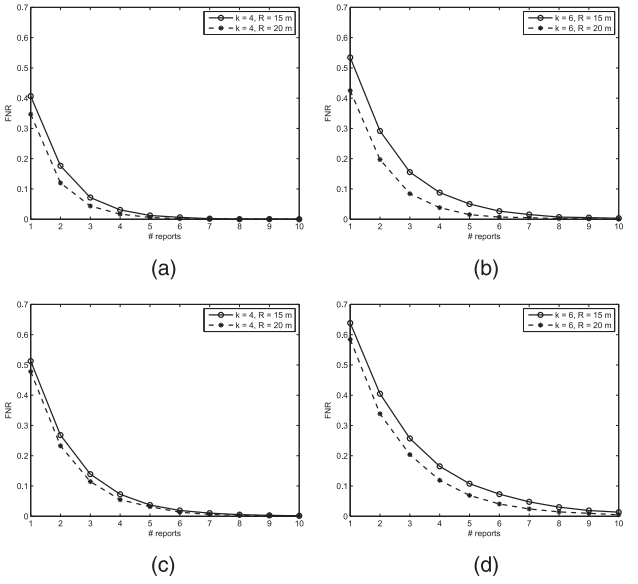


Fig. 10. The false negative rate FNR in terms of different number of independent reports, where the number is from 1 to 10. (a) $\rho = 2\%$, $k = 4$, (b) $\rho = 2\%, k = 6$, (c) $\rho = 5\%, k = 4$, and (d) $\rho = 5\%, k = 6$.

goal, i.e., the injected false data can be *early* detected and filtered by the en-routing sensor nodes. Thus, the energy wasted in relaying injected false data can be reduced.

Reliability of the BECAN scheme. In addition to the high (en-routing) filtering probability, the BECAN scheme also has high reliability, i.e., even though some sensor nodes are compromised, the true event reports still can reach the *sink* with high probability. Let FNR be the false negative rate on the true reports and tested as

$$\text{FNR} = \frac{\text{number of true data that cannot reach the } sink}{\text{total number of true data}}. \tag{18}$$

If FNR is small, the BECAN scheme is demonstrated high reliability. Note that, selectively dropping true report attack [18] can increase the FNR. However, its adverse impact can affect any routing algorithm. Thus, for fairness, we only consider FNR that caused by 1) the number of uncompromised neighboring sensor nodes being less than k, or 2) some compromised sensor nodes polluting the true report. Fig. 10 shows the false negative rate FNR versus different number of reports. It can be seen, when the number of independent reports increases, the FNR decreases. Especially, when the number is five, the FNR is less than 10 percent. In reality, when a true wildfire event takes place, usually several independent entities report the event. Thus, the multireports technology in BECAN scheme fits to the realistic scenarios. As a result, the BECAN scheme can achieve high reliability.

5.3 Discussion on Gang Injecting False Data Attack

In this section, we introduce a new stronger injecting false data attack, called gang injecting false data attack, in wireless sensor networks. This kind of attack is usually launched by a gang of compromised sensor nodes controlled and moved by an adversary $\mathcal{A}$. As shown in Fig. 11, when a compromised source node is ready to send a false data, several compromised nodes will first move and aggregate at the source node, and then collude to inject the false data. Because of the mobility, the gang injecting false data attack is more challenging and hard to resist.

To tackle this kind of attack, a possible solution with the BECAN scheme is to require each participating sensor node to provide its position information. If the current position is not consistent with the previous ones, the gang attack can be detected. Nevertheless, how to prevent/mitigate the gang injecting false data attack from mobile compromised sensor nodes is still worthy of the further investigation.

6 Performance Evaluation

Energy saving is always crucial for the lifetime of wireless sensor networks. In this section, the performance of the proposed BECAN scheme is evaluated in terms of energy efficiency.

6.1 Energy Consumption in Noninteractive Keypair Establishments

The *additional* computation costs of the proposed BECAN scheme are mainly due to the expensive ECDH operations during the noninteractive keypair establishments. Fortunately, since the noninteractive keypair establishments are *averagely* distributed in each sensor node and

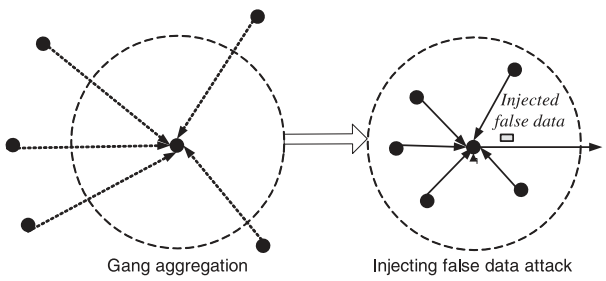


Fig. 11. Gang injecting false data attack.

only executed once during the routing establishment, the ECDH operation is not a heavy burden. When designing TinyECC-based sensor node, we can choose a 160-bit elliptic curve for achieving the same security level as 1,024-bit RSA [25]. Assume that, each sensor node is equipped with a low-power high performance sensor platform, i.e., MICAz [21]. Then, according to [19], this type of sensor platform only requires 50.82 mJ to establish a noninteractive shared key.

6.2 Energy Consumption in Transmission

As shown in Fig. 8, the majority of injected false data can be filtered by BECAN within 15 hops during transmission. Thus, BECAN can greatly save the energy of sensor nodes along the routing path. In order to quantitatively measure the energy saving in BECAN, we compare the energy consumption of BECAN with that of SEF within the length of routing path $H = 15$ hops. For fair comparison, we set the parameter $k = 4$, and 0, three among four neighboring nodes colluding with the compromised source node N_0, which corresponds to $N_c = 1, 4$ with $T = 5$ in SEF [9]. Because SEF does not consider the compromise of en-routing nodes, we also set $\rho = 0$ in BECAN.

Let p be the probability to detect and drop an injected false data at each en-routing node. Then, the expected probability of false data being detected within h hops is $p_h = 1 - (1-p)^h$. Let X be the number of hops that an injected false data can traverse. Then, the average number of hops that an injected false data traverse within total H hops is given as

$$\begin{aligned} &E[X|X \le H] \\ &= \sum_{i=1}^{\infty} iP(X = i|X \le H) = \sum_{i=1}^{\infty} i\frac{P(X = i, X \le H)}{P(X \le H)} \\ &= \sum_{i=1}^{H} i\frac{P(X = i)}{P(X \le H)} = \sum_{i=1}^{H} i\frac{(1-p)^{i-1}p}{1-(1-p)^H} \\ &= \frac{1}{p}\left(1 - \frac{Hp(1-p)^H}{1-(1-p)^H}\right). \end{aligned} \tag{19}$$

Note that, when four and one compromised nodes collude in BECAN, the detection probability p are $1 - 1/2^{5-4} = 1/2$, $1 - 1/2^{5-1} = 15/16$, respectively; while in SEF,[1] p is usually suggested as $1/20$, $1/5$ for $N_c = 4$ and 1, respectively [9].

Let $L_m = 24$ bytes be the length of a report m without any extra field, L_a the additional authentication overhead, and E_u the energy consumption in transmitting and receiving one byte. Also, we assume that there are 10^3 legitimate data traffic and $10^3 \times \beta$ injected false data traffic, where β, $(1 \le \beta \le 10)$, is the normalized amount of injected traffic. Then, the energy consumed to deliver all traffic without BECAN/SEF will be

$$E_{w/o} = 10^3(1+\beta) \cdot H \cdot L_m \cdot E_u, \tag{20}$$

and the average energy consumption with BECAN/SEF will be

1. Note that, the probability p is defined as $\frac{k(T-N_c)}{N}$ in SEF, where N is the key pool size, and k is the number of keys held by an sensor node. For security reason, $\frac{k}{N}$ cannot reach $1/2$. Otherwise, once a sensor node is compromised, $\frac{N}{2}$ key materials will be disclosed and abused. Therefore, p should not be set too large in SEF.

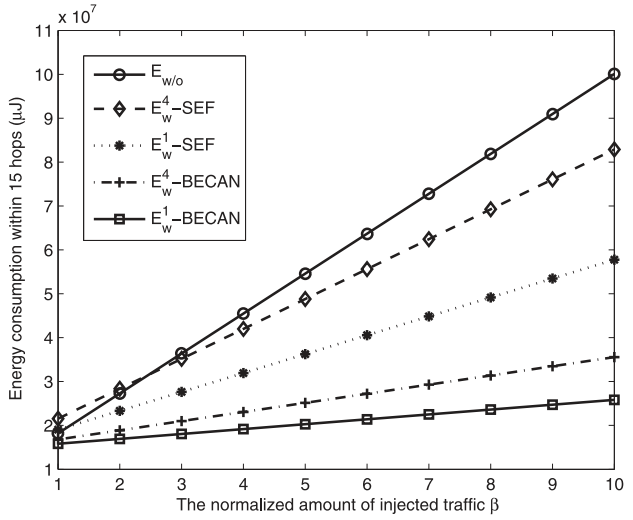


Fig. 12. The energy consumption as a function of the normalized amount of injected traffic β with the total traffic $10^3(1+\beta)$ within $H = 15$ hops. $E_{w/o}$ is the energy amount without SEF/BECAN, E_w^1-SEF and E_w^4-SEF are the amounts with SEF and the attacker has keys in 1, 4, distinct partitions with 5 carried MACs. E_w^1-BECAN and E_w^4-BECAN are the amounts with BECAN and 0, 3 neighboring compromised nodes collude with the compromised nodes N_0, respectively. BECAN uses less energy than SEF when $\beta \ge 1$ during the transmission.

$$E_w = 10^3(L_m + L_a) \cdot E_u \cdot \left(H + \beta \cdot \frac{1}{p}\left(1 - \frac{Hp(1-p)^H}{1-(1-p)^H}\right)\right). \tag{21}$$

Let the lengths of key index, identity, and timestamp in BECAN/SEF be 10 bits, respectively. Then, L_a in BECAN and SEF are 135 bits, 144 bits, respectively. According to [21], the measurements show that MICAz node consumes 19.7 mA current when receiving, and 14 mA when transmitting. Based on the battery voltage (3 V) and data rate (32 Kbps), we can calculate that it takes to $10.5/14.77$ μJ to transmit/receive a byte. Then, $E_u = 25.27$ μJ. If we use RC5 [25] for MAC computation, each computation consumes about 15 μJ [9]. By plugging the energy consumed in MAC computation in (21), we can plot the energy consumptions as a function of the normalized amount of injected traffic β with the total traffic $10^3(1+\beta)$ within $H = 15$ hops in Fig. 12. From the figure, we can observe that the proposed BECAN scheme can save more energy than SEF, especially when β is large. The reason is that the proposed BECAN scheme has larger p than SEF. Based on (19), the average numbers of hops that an injected false data can traverse are 1, 2 hops for $p = 15/16, 1/2$, respectively, in BECAN, and $4.5, 7$ hops for $p = 1/5, 1/20$, respectively, in SEF, and thus, more energy will be saved in BECAN. Based on the above analysis, we can see the proposed BECAN scheme is indeed an efficient authentication scheme for filtering injected false data in wireless sensor networks.

7 Related Work

Recently, some research works on bandwidth-efficient filtering of injected false data in wireless sensor networks have been appeared in the literature in [9], [10], [11], [12], [13]. In [9], Ye et al. propose a statistical en-routing filtering mechanism called SEF. SEF requires that each sensing report

be validated by multiple keyed message authenticated (MACs), each generated by a node that detectes the same event. As the report being forwarded, each node along the way verifies the correctness of the MACs at earliest point. If the injected false data escapes the en-routing filtering and is delivered to the *sink*, the *sink* will further verify the correctness of each MAC carried in each report and reject false ones. In SEF, to verify the MACs, each node gets a random subset of the keys of size k from the global key pool of size N and uses them to producing the MACs. To save the bandwidth, SEF adopts the bloom filter to reduce the MAC size. By simulation, SEF can prevent the injecting false data attack with 80-90 percent probability within 10 hops. However, since n should not be large enough as described above, the filtering probability at each en-routing node $p = \frac{k(T-N_c)}{N}$ is relatively low. Besides, SEF does not consider the possibility of en-routing nodes' compromise, which is also crucial to the false data filtering. In [10], Zhu et al. present an interleaved hop-by-hop authentication (IHA) scheme for filtering of injected false data. In IHA, each node is associated with two other nodes along the path, one is the lower association node, and the other is the upper association node. An en-routing node will forward received report if it is successfully verified by its lower association node. To reduce the size of the report, the scheme compresses $t+1$ individual MACs by XORing them to one. By analyses, only if less than t nodes are compromised, the *sink* can detect the injected false data. However, the security of the scheme is mainly contingent upon the creation of associations in the association discovery phase. Once the creation fails, the security cannot be guaranteed. In addition, as pointed in [7], Zhu et al.'s scheme, similar as SEF, also adopts the symmetric keys from a key pool, which allows the compromised nodes to abuse these keys to generate false reports. Location-Based Resilient Secrecy (LBRS) is proposed by Yang et al. [11], which adopts location key binding mechanism to reduce the damage caused by node compromise, and further mitigate the false data generation in wireless sensor networks. In [12], Ren et al. propose more efficient location-aware end-to-end data security design (LEDS) to provide end-to-end security guarantee including efficient en-routing false data filtering capability and high-level assurance on data availability. Because LEDS is a symmetric key based solution, to achieve en-routing filtering, it requires location-aware key management, where each node should share at least one authentication key with one node in its upstream/downstream *report-auth cell*. In [13], Zhang et al. provide a public key based solution to the same problem. Especially, they propose the notion of location-based keys by binding private keys of individual nodes to both their IDs and geographic locations and a suite of location-based compromise-tolerant security mechanisms. To achieve en-routing filtering, additional 20 bytes authentication overheads are required.

Bit-compressed authentication technology can achieve bandwidth-efficient, which has been adopted in some research works [29], [30]. In [29], Canetti et al. use one-bit authentication to achieve multicast security. The basic idea in multicast is very similar to the BECAN scheme, where a source knows a set of keys $\mathbf{R} = \{K_1, \ldots, K_l\}$, each recipient u knows a subset $\mathbf{R}_u \subset \mathbf{R}$. When the source sends a message M, it authenticates M with each of the keys, using a MAC. That is, a message M is accompanied with $\langle MAC(K_1, M), \ldots, MAC(K_l, M)\rangle$. Each recipient u verifies all the MACs which were created using the keys in its subset $\mathbf{R}_u$. If any of these MACs is incorrect, the message M will be rejected. To achieve the bandwidth efficiency, each MAC is compressed as single bit. The security of the scheme is based on the assumption that the source is not compromised. However, once the source is compromised, the scheme obviously does not work. Therefore, it cannot be applied to filter false data injected by compromised nodes in wireless sensor networks. In [30], Benenson et al. also use 1-bit MACs to decide whether a query is legitimate in wireless sensor networks. However, similar as that in [29], once the source is compromised, the 1-bit MACs also does not work. Different from the above works, the proposed BECAN scheme adopts CNR based filtering mechanism together with multireports technology. Because of noninteractive key establishment, BECAN does not require a complicated security association [10], [12]. In addition, BECAN considers the scenario that each node could be compromised with probability ρ, i.e., some en-routing nodes could be compromised. To avoid putting all eggs in one basket, BECAN distributes the en-routing authentication to all sensor nodes along the routing path. To save the bandwidth, it also adopts the bit-compressed authentication technique. Therefore, it is compromise-tolerant and suitable for filtering false data in wireless sensor networks.

8 Conclusion

In this paper, we have proposed a novel BECAN scheme for filtering the injected false data. By theoretical analysis and simulation evaluation, the BECAN scheme has been demonstrated to achieve not only high en-routing filtering probability but also high reliability with multi-reports. Due to the simplicity and effectiveness, the BECAN scheme could be applied to other fast and distributed authentication scenarios, e.g., the efficient authentication in wireless mesh network [31]. In our future work, we will investigate how to prevent/mitigate the gang injecting false data attack from mobile compromised sensor nodes [32].

Appendix

We will show how to use Dijkstra algorithm to calculate and store the shortest path $\xi(v_i)$ for each sensor node $v_i \in \mathcal{V}/\{v_1\}$ to the *sink* v_1.

Algorithm 5. Dijkstra's Single Source All Shortest Path

procedure DIJKSTRA ($\mathcal{G} = (\mathcal{V}, \mathcal{E})$)
  $\xi(v_1) = 0$; $\widetilde{\mathcal{V}} = \mathcal{V}/\{v_1\}$;
  for each vertex $v_i \in \widetilde{\mathcal{V}}$ **do**
    $\xi(v_i) = +\infty$
  end for
  for each edge $e_{ij} \in \mathcal{E}$ **do**
    if $e_{ij} == 1$ **then**
      $w(v_i, v_j) = 1$
    else ▷ $e_{ij} == 0$
      $w(v_i, v_j) = +\infty$
    end if

12: **end for**
13: **while** $\widetilde{\mathcal{V}} \neq \phi$ **do**
14: choose vertex $v_i \in \widetilde{\mathcal{V}}$ such that $\xi(v_i)$ is minimal in $\widetilde{\mathcal{V}}$
15: $\widetilde{\mathcal{V}} = \widetilde{\mathcal{V}}/\{v_i\}$
16: **for** each vertex $v_j \in \widetilde{\mathcal{V}}$ **do**
17: $\xi(v_j) = \min[\xi(v_j), \xi(v_i) + w(v_i, v_j)]$
18: **end for**
19: **end while**
20: **end procedure**

REFERENCES

[1] R. Szewczky, A. Mainwaring, J. Anderson, and D. Culler, "An Analysis of a Large Scale Habit Monitoring Application," *Proc. Second ACM Int'l Conf. Embedded Networked Sensor Systems (Sensys '04),* 2004.

[2] L. Eschenauer and V.D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," *Proc. Ninth ACM Conf. Computer and Comm. Security (CCS '02),* 2002.

[3] R. Lu, X. Lin, C. Zhang, H. Zhu, P. Ho, and X. Shen, "AICN: An Efficient Algorithm to Identify Compromised Nodes in Wireless Sensor Network," *Proc. IEEE Int'l Conf. Comm. (ICC '08),* May 2008.

[4] X. Lin, R. Lu, and X. Shen, "MDPA: Multidimensional Privacy-Preserving Aggregation Scheme for Wireless Sensor Networks," *Wireless Comm. and Mobile Computing,* vol. 10, pp. 843-856, 2010.

[5] X. Lin, "CAT: Building Couples to Early Detect Node Compromise Attack in Wireless Sensor Networks," *Proc. IEEE GLOBECOM '09,* Nov.-Dec. 2009.

[6] K. Ren, W. Lou, and Y. Zhang, "Multi-User Broadcast Authentication in Wireless Sensor Networks," *Proc. IEEE Sensor Ad Hoc Comm. Networks (SECON '07),* June 2007.

[7] L. Zhou and C. Ravishankar, "A Fault Localized Scheme for False Report Filtering in Sensor Networks," *Proc. Int'l Conf. Pervasive Services, (ICPS '05),* pp. 59-68, July 2005.

[8] Z. Zhu, Q. Tan, and P. Zhu, "An Effective Secure Routing for False Data Injection Attack in Wireless Sensor Network," *Proc. 10th Asia-Pacific Network Operations and Management Symp. (APNOMS '07),* pp. 457-465, 2007.

[9] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical En-Route Detection and Filtering of Injected False Data in Sensor Networks," *Proc. IEEE INFOCOM '04,* Mar. 2004.

[10] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An Interleaved Hop-by-Hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks," *Proc. IEEE Symp. Security and Privacy,* 2004.

[11] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, "Toward Resilient Security in Wireless Sensor Networks," *Proc. Sixth ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc '05),* pp. 34-45, 2005.

[12] K. Ren, W. Lou, and Y. Zhang, "LEDS: Providing Location-Aware End-to-End Data Security in Wireless Sensor Networks," *Proc. IEEE INFOCOM '06,* Apr. 2006.

[13] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-Based Compromise-Tolerant Security Mechanisms for Wireless Sensor Networks," *IEEE J. Selected Areas in Comm.,* vol. 24, no. 2, pp. 247-260, Feb. 2006.

[14] C.-M. Yu, C.-S. Lu, and S.-Y. Kuo, "A Dos-Resilient En-Route Filtering Scheme for Sensor Networks," *Proc. Tenth ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc '09),* pp. 343-344, 2009.

[15] J. Chen, Q. Yu, Y. Zhang, H.-H. Chen, and Y. Sun, "Feedback Based Clock Synchronization in Wireless Sensor Networks: A Control Theoretic Approach," *IEEE Trans. Vehicular Technology,* vol. 59, no. 6, pp. 2963-2973, June 2010.

[16] S. He, J. Chen, Y. Sun, D.K.Y. Yau, and N.K. Yip, "On Optimal Information Capture by Energy-Constrained Mobile Sensors," *IEEE Trans. Vehicular Technology,* vol. 59, no. 5, pp. 2472-2484, June 2010.

[17] K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks," *Ad Hoc Networks,* vol. 3, no. 3, pp. 325-349, May 2005.

[18] V.C. Giruka, M. Singhal, J. Royalty, and S. Varanasi, "Security in Wireless Sensor Networks," *Wireless Comm. and Mobile Computing,* vol. 8, no. 1, pp. 1-24, Jan. 2008.

[19] A. Liu and P. Ning, "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks," *Proc. Seventh Int'l Conf. Information Processing in Sensor Networks (IPSN '08),* pp. 245-256, Apr. 2008.

[20] J. Dong, Q. Chen, and Z. Niu, "Random Graph Theory Based Connectivity Analysis in Wireless Sensor Networks with Rayleigh Fading Channels," *Proc. Asia-Pacific Conf. Comm. (APCC '07),* pp. 123-126, Oct. 2007.

[21] MICAz: Wireless Measurement System, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Data sheet.pdf, 2010.

[22] Imote2: High-Performance Wireless Sensor Network Node, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Imote2_Dat asheet.pdf, 2010.

[23] C. Boyd, W. Mao, and K.G. Paterson, "Key Agreement Using Statically Keyed Authenticators," *Proc. Second Int'l Conf. Applied Cryptography and Network Security C (ACNS '04),* pp. 248-262, 2004.

[24] J. Black and P. Rogaway, "Cbc Macs for Arbitrary-Length Messages: the Three-Key Constructions," *J. Cryptology,* vol. 18, no. 2, pp. 111-131, 2005.

[25] W. Mao, *Modern Cryptography: Theory and Practice.* Prentice Hall PTR, 2003.

[26] X. Li, N. Santoro, and I. Stojmenovic, "Localized Distance-Sensitive Service Discovery in Wireless Sensor and Actor Networks," *IEEE Trans. Computers,* vol. 58, no. 9, pp. 1275-1288, Sept. 2009.

[27] X. Li, A. Nayak, D. Simplot-Ryl, and I. Stojmenovic, "Sensor Placement in Sensor and Actuator Networks," *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication,* Wiley, 2010.

[28] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An Effective Key Management Scheme for Heterogeneous Sensor Networks," *Ad Hoc Networks,* vol. 5, pp. 24-34, Jan. 2007.

[29] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast Security: A Taxonomy and Some Efficient Constructions," *Proc. IEEE INFOCOM '99,* pp. 708-716, Mar. 1999.

[30] Z. Benenson, C. Freiling, E. Hammerschmidt, S. Lucks, and L. Pimenidis, "Authenticated Query Flooding in Sensor Networks," *Security and Privacy in Dynamic Environments,* Springer, pp. 38-49, July 2006.

[31] X. Lin, R. Lu, P. Ho, X. Shen, and Z. Cao, "TUA: A Novel Compromise-Resilient Authentication Architecture for Wireless Mesh Networks," *IEEE Trans. Wireless Comm.,* vol. 7, no. 4, pp. 1389-1399, Apr. 2008.

[32] C. Zhang, R. Lu, X. Lin, P. Ho, and X. Shen, "An Efficient Identity-Based Batch Verification Scheme for Vehicular Sensor Networks," *Proc. IEEE INFOCOM '08,* Apr. 2008.

Rongxing Lu (S'09-M'11) is currently working toward the PhD degree in the Department of Electrical and Computer Engineering from the University of Waterloo, Canada. He is currently a research assistant with the Broadband Communications Research (BBCR) Group, University of Waterloo. His research interests include wireless network security, applied cryptography, and trusted computing. He is a student member of the IEEE.

Xiaodong Lin (S'07-M '09) received the PhD degree in information engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1998, and the PhD degree (with Outstanding Achievement in Graduate Studies Award) in electrical and computer engineering from the University of Waterloo, Canada, in 2008. He is currently an assistant professor of information security with the faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, Canada. His research interests include wireless network security, applied cryptography, computer forensics, and software security. He was the recipient of a Natural Sciences and Engineering Research Council of Canada (NSERC), Canada Graduate Scholarships (CGS), Doctoral and the Best Paper Awards of the IEEE International Conference on Computer Communications and Networks (ICCCN '09), and the IEEE International Conference on Communications (ICC '07)—Computer and Communications Security Symposium. He is a member of the IEEE.

Haojin Zhu (M'09) received the BSc degree in computer science from Wuhan University, China, in 2002, the MSc degree in computer science from Shanghai Jiao Tong University, China, in 2005, and the PhD degree in electrical and computer engineering from the University of Waterloo, Canada, in 2009. Currently, he is an assistant professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His current research interests include wireless network security and distributed system security. He is a corecipient of best paper awards of IEEE ICC 2007—Computer and Communications Security Symposium and Chinacom 2008—Wireless Communication Symposium. He serves as the technical program committee chair for international conferences, such as Infocom, ICCCN, Globecom, ICC, WCNC and etc. He is a member of the IEEE.

Xiaohui Liang (S'10) is currently working toward the PhD degree in the Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently a research assistant with the Broadband Communications Research (BBCR) Group, University of Waterloo. His research interests include wireless network security, applied cryptography, and e-healthcare system. He is a student member of the IEEE.

Xuemin (Sherman) Shen (M'97-SM'02-F'09) received the BSc degree in electrical engineering from Dalian Maritime University, China, in 1982, and the MSc and PhD degrees in electrical engineering from Rutgers University, New Jersey, in 1987 and 1990, respectively. He is a professor and university research chair in the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include resource management in interconnected wireless/wired networks, UWB wireless communications networks, wireless network security, wireless body area networks and vehicular ad hoc and sensor networks. He is a coauthor of three books, and has published more than 400 papers and book chapters in wireless communications and networks, control and filtering. He has served as the technical program committee chair for IEEE VTC'10, the tutorial chair for IEEE ICC'08, the technical program committee chair for IEEE Globecom'07, the general cochair for Chinacom'07 and QShine'06, the founding chair for IEEE Communications Society Technical Committee on P2P Communications and Networking. He has also served as a founding area editor for *IEEE Transactions on Wireless Communications*; editor-in-chief for *Peer-to-Peer Networking and Application*; an associate editor for *IEEE Transactions on Vehicular Technology*; *Computer Networks*; and *ACM/Wireless Networks,* guest editor for IEEE *JSAC*, *IEEE Wireless Communications*, *IEEE Communications Magazine*, and *ACM Mobile Networks and Applications*, etc. He received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. He is a registered professional engineer of Ontario, Canada, and a distinguished lecturer of IEEE Communications Society. He is a fellow of the IEEE.