Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 510251, 10 pages
http://dx.doi.org/10.1155/2015/510251

Research Article

Dual-Homomorphic Message Authentication Code Scheme for Network Coding-Enabled Wireless Sensor Networks

Alireza Esfahani,[1] Du Yang,[1] Georgios Mantas,[1] Alberto Nascimento,[1] and Jonathan Rodriguez[1,2]

[1] *Instituto de Telecomunicações (IT), Campus Universitário de Santiago, 3810-193 Aveiro, Portugal*
[2] *Universidade de Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal*

Correspondence should be addressed to Alireza Esfahani; alireza@av.it.pt

Received 27 July 2014; Revised 29 December 2014; Accepted 1 January 2015

Academic Editor: Muhammad Khurram Khan

Network coding has shown a considerable improvement in terms of capacity and robustness compared to traditional store-and-forward transmission paradigm. However, since the intermediate nodes in network coding-enabled networks have the ability to change the packets en route, network coding-enabled networks are vulnerable to pollution attacks where a small number of polluted messages can corrupt bunches of legitimate messages. Recently, research effort has been put on schemes for protecting the transmitted messages against data pollution attacks. However, most of them cannot resist tag pollution attacks. This paper presents a new homomorphic MAC-based scheme, called Dual-Homomorphic MAC (Dual-HMAC), for network coding-enabled wireless sensor networks. The proposed scheme makes use of two types of tags (i.e., MACs and D-MACs) to provide resistance against data pollution attacks and partially tag pollution attacks. Furthermore, our proposed scheme presents low communication overhead and low computational complexity compared to other existing schemes.

1. Introduction

Wireless sensor networks (WSNs) have been recently used in a plethora of applications of many different areas, such as surveillance monitoring, environmental monitoring, traffic control, natural disaster prevention, and e-health. Due to their wide range of applications, they have already become so attractive to research community. However, WSNs are characterized by low communication bandwidth and power consumption constraints [1]. Thus, network coding (NC) can be a promising solution for WSNs, since it is a technique that can provide network capacity improvement [2] and lower energy consumption [3].

NC was introduced by Ahlswede et al. [2]. Nowadays, NC has been used in various applications over different networks, such as wireless mesh networks [4], wireless sensor networks [5], and peer-to-peer systems [6]. In contrast to traditional and classical commodity flow, where information is only routed or replicated, information flow can also employ coding operations at intermediate nodes. In [7], random linear network coding (RLNC) was studied as a fully distributed method for performing network coding. Also, in [7], it is mentioned that there is a possibility for each node in the network to select a set of coefficients independently and randomly and use them to make linear combinations of the data symbols.

However, NC is more susceptible to data pollution attack than the traditional store-and-forward approach. If a data pollution attack is not detected at the forwarders, then the sink nodes will not be able to recover the source messages correctly. It is worthwhile to mention that even a small number of polluted messages can infect a large number of downstream nodes because the pollution propagates via recoding. Furthermore, NC-enabled networks are also vulnerable to tag pollution attack, which is a more sophisticated type of pollution attack. In tag pollution attack, the adversary targets at modifying (i.e., polluting) the tags carried by the messages rather than modifying the content of the messages. It is possible for a message with polluted tags to travel multiple hops until it is detected and discarded. Yet, this results in a waste of network bandwidth.

Thus far, several cryptographic schemes such as homomorphic hash functions, signatures, and MACs have been presented to secure network coding against data and tag pollution attacks (e.g., [8, 9]). Furthermore, information-theoretic schemes have been also proposed (e.g., [10, 11]) to detect data pollution attacks. Although information-theoretic schemes are more efficient, in terms of computation, compared to the cryptographic schemes, they can only detect polluted packets at the sink nodes.

Among all the above mentioned proposed schemes, Homomorphic MAC is considered as a low-complexity solution against data pollution [12, 13]. Specifically, a MAC or tag is a small piece of information appended to the end of the message packet. This piece of information is the output of a MAC function taking as inputs the message packet and a secret key. However, this solution is vulnerable to tag pollution attacks.

In this paper, we propose a new homomorphic MAC-based scheme, called Dual-Homomorphic MAC (Dual-HMAC), for NC-enabled WSNs, in order to mitigate both data pollution attacks and partially tag pollution attacks. In our scheme, the source generates multiple MACs and Dual MACs (D-MACs) for each message. Each MAC ensures integrity of the transmitted message and each D-MAC ensures integrity of the MACs. In other words, by appending D-MACs, the proposed scheme mitigates partially tag pollution attacks.

The rest of the paper is outlined as follows. In Section 2, we give an overview of the related work on security schemes against data pollution attacks and tag pollution attacks in NC-enabled networks. In Section 3, the problem statement is discussed. In Section 4, our proposed Dual-HMAC scheme is presented. In Section 5, the security analysis of the proposed scheme takes place. In Section 6, its performance evaluation is given. In Section 7, a comparison among the communication and computation overhead of our work (i.e., Dual-HMAC) and the communication and computation overhead of the related works in [9, 14] is provided. Finally, Section 8 concludes the paper.

2. Related Work

In this section, we present related work on security schemes against pollution attacks in NC-enabled networks. In addition, we present the key distribution mechanisms used by these security schemes.

2.1. Security Schemes against Pollution Attacks in NC-Enabled Networks. Most of the research in the field of pollution attacks in network coding is focused on two types of security schemes: information-theoretic schemes [10, 11] and cryptographic schemes [14, 15].

2.1.1. Cryptographic Schemes. These schemes rely on additional verification information which is added by the sources through cryptographic techniques. This allows intermediate nodes to verify the original messages and filter out the polluted ones. This category includes the homomorphic hashing schemes, the homomorphic signature schemes, and the homomorphic MACs.

(a) Homomorphic Hashing Schemes. Homomorphic hashing schemes are based on a homomorphic hash function which is applied by the source to the messages. These schemes rely also on additional secure communication channels in order to transmit the calculated hash values to the intermediate nodes. A well-known homomorphic hashing scheme is proposed by Krohn et al. in [8] which enables the intermediate nodes to verify on-the-fly. The extension of this work was presented in [16] which is focused on network coding-enabled networks and further reduces the expensive computation cost of hash functions in each intermediate node by enabling cooperation verification.

(b) Homomorphic Signature Schemes. Homomorphic signature schemes were introduced for the first time by Charles et al. in [17] and have been based on Weil pairing over elliptic curves. The homomorphic property of the signatures in these schemes allows nodes to sign any linear combination of the incoming packets without contacting the signing entity. The calculation of signature covers the whole augmented message. In [18], Yu et al. proposed a homomorphic signature function which allows the relay nodes to verify the received messages by generating the signatures without contacting the signing entities. In Yu's scheme, the communication does not need any extra secure communication channel. Their experimental results show that its verification efficiency is improved up to 10 times compared to the other related work (i.e., CJLs scheme [17]). Moreover, in [10], an alternative lightweight scheme is proposed. It is much faster rather than Yu's scheme, but it is not as secure as the first one.

(c) Homomorphic MACs. Homomorphic MACs are based on appending some extra information to the original message. The basic definition of homomorphic MAC is defined by Agrawal and Boneh in [14] which allows the integrity checking of the network coded data. However, this idea is susceptible to tag pollution attacks. Kehdi and Baochun [19] presented a homomorphic MAC scheme to detect pollution attacks based on the subspace properties of random linear network coding, where null keys for verification are used. As a lot of null keys are used in each generation, a high bandwidth overhead can be incurred. The authors in [13] proposed an idea where the source attaches multiple MACs to each packet. This idea can be applied for XOR network coding networks; however it is vulnerable to tag pollution attacks. RIPPLE [20] was proposed to counteract against the tag pollution problem. This work is based on a symmetric key based scheme for network coding authentication. RIPPLE allows a node to efficiently detect corrupted packets and encode only the authenticated (i.e., verified) ones. However, the global synchronization among all nodes is the problem of this scheme. In [9], the authors presented a hybrid-key cryptography approach which could achieve authentication in the presence of both data and tag pollution attacks. To achieve this, a number of tags and a signature are appended to each transmitted message. However, the signature which

is necessary for preventing tag pollution is a time consuming process.

2.1.2. Information-Theoretic Schemes. The information-theoretic schemes benefit from the reasonable computation performance in comparing to the cryptographic schemes, but they have two main drawbacks. First of all, they are not able to detect the polluted messages at the intermediate nodes of the network. For example, the work presented in [10], which extends the random network coding, relies on coding redundant information into the original messages allowing only the receivers to detect the polluted messages. Furthermore, the information-theoretic schemes are characterized by the limitation on the number of compromised nodes. These two drawbacks comprise the main motivation to interest to the cryptographic schemes.

2.2. Key Distribution Mechanisms in NC-Enabled Networks. Nowadays, in emerging multicast communication systems (e.g., pay TV and video-conferencing services), the key distribution is a challenging issue. There are various proposed schemes for key distribution in multicast systems. Key distribution models can be categorized in trusted-server schemes [21], public-key schemes, and key predistribution schemes [22]. The trusted-server and public-key schemes are infeasible to be used in WSNs as the sensors have limitations in terms of energy and computing processes. On the other hand, key predistribution schemes are feasible to be used in WSNs. In these schemes, the shared keys are distributed among the nodes of a group in such a way that every node in the group is able to compute individually a common key associated with that group. Two examples of key predistribution schemes are given as follows.

In MacSig which is an efficient subspace authentication scheme for NC and presented in [9], the double-random key distribution has been proposed. This proposed keys distribution model includes two random procedures. First of all, they assign each node with a random set of keys; next, the source node randomly selects a subset of keys from the key pools and uses these keys to generate the MACs. However, it is necessary to attach the indexes of these keys to each packet. Hence, a considerable bandwidth overhead is incurred.

In KEPTE (key predistribution-based tag encoding) scheme, a key distribution model for practical NC has been proposed in [23]. According to this key distribution model, a Key Distribution Center (KDC) is responsible for the distribution of the keys. The KDC allocates N secret vectors at the source node to produce N tags. Moreover, it assigns two secret vectors to each node to check the correctness of its received packets.

3. Problem Statement

In this section, we provide our problem statement through a network coding-enabled WSN model and its corresponding adversary model.

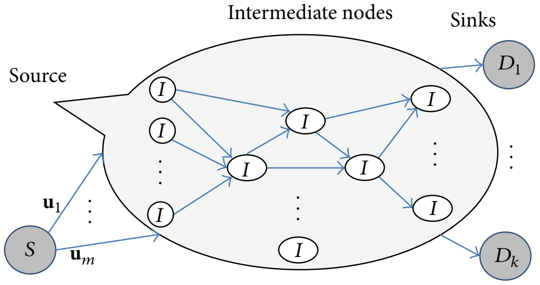


Figure 1: A general multicast network.

3.1. Network Coding-Enabled WSN Model. We consider a network coding-enabled WSN, as it is depicted in Figure 1, consisting of a source node S, a number of intermediate nodes, and a set of sink nodes. Random linear network coding is exploited in this network. In order to model our network, we define a triple (G, S, R) which consists of the following components:

(i) *directed multigraph* G: we consider a pair of V, E as a directed acyclic graph where V and E are the node sets and edge sets of G, respectively;

(ii) *source node* S: in our network model, we have a source S that wants to multicast its messages;

(iii) *non-source node* R: we define relay and sink nodes in a set of nodes which is defined as: $R = \{\forall x \in V - \{S\}\}$.

Notations section summarizes the main notations used in this paper.

Prior to transmission, the source node partitions message packets into generations. Each generation consists of m messages packets denoted as $\underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, \ldots, \underline{\mathbf{u}}_m$. Each packet $\underline{\mathbf{u}}_i$ is represented as a vector of n symbols $(\underline{u}_{i,1}, \underline{u}_{i,2}, \ldots, \underline{u}_{i,n})$ in the finite field $\mathbb{F}_p^n$. In [24], it is shown that $p = 2^8$ is usually sufficient for practical use and convenient for computation. Then, the source node generates an augmented packet $\mathbf{u}_i$ by prefixing $\underline{\mathbf{u}}_i$ with the ith unit vector of dimension m represented as

$$\mathbf{u}_i = \Big(\overbrace{\underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0}^{m}, \underline{u}_{i,1}, \ldots, \underline{u}_{i,n} \Big) \in \mathbb{F}_p^{m+n}. \quad (1)$$

After that, the source node transmits these augmented packets to its neighbour nodes. During transmission, an intermediate node buffers its received packets $\underline{\mathbf{u}}_i$ temporarily and creates an outgoing message y as linear combinations of packets $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_h$ belonging to the same generation:

$$y = \sum_{i=1}^{h} c_i \mathbf{u}_i, \quad (2)$$

where each $c_i \in \mathbb{F}_p$ is a random coefficient chosen by each forwarder. A coded packet y is considered to be valid if it is in the linear subspace spanned by the original augmented packets, denoted as $y \in \mathrm{Span}\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$. In fact, when y is valid, these linear combination coefficients are the first m

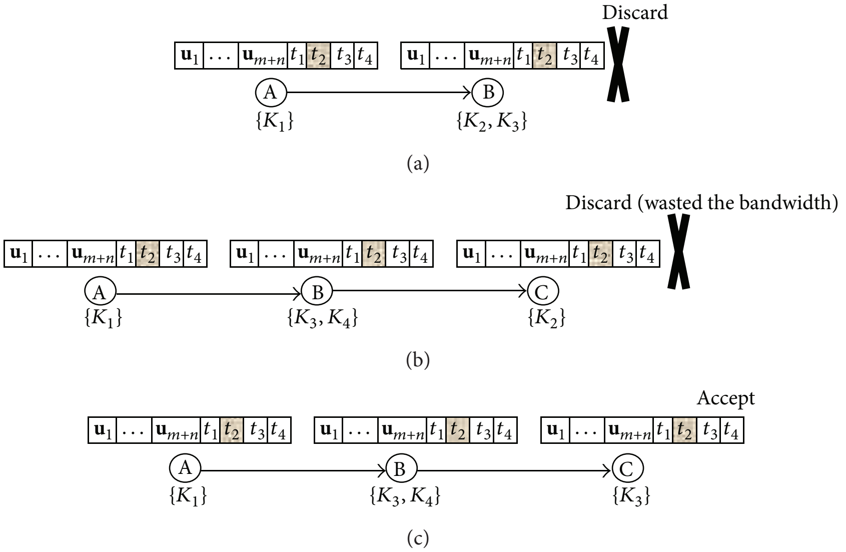

Discard
$\mathbf{u}_1$ ... $\mathbf{u}_{m+n}$ t_1 t_2 t_3 t_4
A
B
$\{K_1\}$
$\{K_2, K_3\}$
(a)
Discard (wasted the bandwidth)
C
$\{K_3, K_4\}$
$\{K_2\}$
(b)
Accept
$\{K_3\}$
(c)

Figure 2: A basic scenario of tag pollution. We consider a scenario which has a packet and 4 tags. We consider a node A as an adversary that pollutes the second tag t_2 of the packet. In (a), we see that the next hop, node B, has two sets of keys (K_2, K_3) and is able to detect tag pollution. As a result node B discards the packet. In (b), the next hop, node B does not have (K_2), so it allows the polluted packet to pass which results in wasted bandwidth; however, the hop, node C is able to detect the polluted packet. In the last scenario shown in (c), we demonstrate the case, where totally wasted bandwidth occurs since none of the nodes holds key K_2.

symbols of the packet y. Otherwise, y is invalid and it is denoted as $y \notin \text{Span}\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$, which may be caused by transmission errors or pollution attacks. After transmission, when a sink node has obtained m linearly independent coded packets, it can decode those using Gaussian eliminations, so as to recover the original messages [7].

3.2. Adversary Model. For the above mentioned network coding-enabled model, we consider that the source is always trusted and there is no possibility to forge it, but the intermediate and sink nodes can be compromised. The adversary is able to wiretap all the data packets that are transmitted over a network. The adversary's goal is to achieve pollution attack. There are two types of pollution attack.

Data Pollution Attack. An adversary can inject fake data packets into the network. The objective of data pollution attack is to pass the verification of other innocent nodes and to cause incorrect decoding at the sink node, as well as wasting of bandwidth.

Tag Pollution Attack. An adversary injects a corrupted packet consisting of correct data but modified tags to the network. The objective of tag pollution is to discard the correct data packets due to the corresponding corrupted tags. This results in the waste of bandwidth. A basic scenario of tag pollution attack is presented in Figure 2.

4. Proposed Scheme: Dual-Homomorphic MAC (Dual-HMAC)

In this section, we present our Dual-HMAC scheme, which mitigates data pollution and partially tags pollution attacks for a network coding-enabled WSN. The Dual-HMAC scheme's outline, the construction and the correctness of the scheme, and the key distribution model are provided in this section.

Our proposed solution is based on the homomorphic MAC solution, defined by Agrawal and Boneh [14], and the MacSig scheme proposed in [9]. Additionally, our proposed scheme makes use of additional MACs, termed as D-MACs, in order to resist data pollution and partially tag pollution attacks. In contrast to the homomorphic MAC solution in [14], our proposed scheme is not susceptible to tag pollution attacks. Particularly, our proposed scheme can achieve resistance for the half of the tags against tag pollution attack with significant low computational complexity compared to the MacSig scheme, which uses a signature scheme characterized by a much considerable computational overhead.

Our scheme consists of three steps, as it is depicted in Figure 3, in order to generate the required MACs and D-MACs. In step 1, it computes a number of MACs (i.e., tags), for each message, according to the keys which are chosen randomly from the first pool of keys. Then, in step 2, it computes a number of D-MACs for the initial MACs, according to the keys which are chosen from the second pool of keys. Finally, in step 3, the computed MACs and D-MACs are appended to the message. More specifically, the main idea of our work is based on the orthogonal subspace property [9].

4.1. The Outline of the Dual-HMAC Scheme. In this section, we provide description of the proposed Dual-HMAC scheme with a formal method. This scheme includes four steps: *setup, tag generation, verification,* and *encoding* detailed as follows.

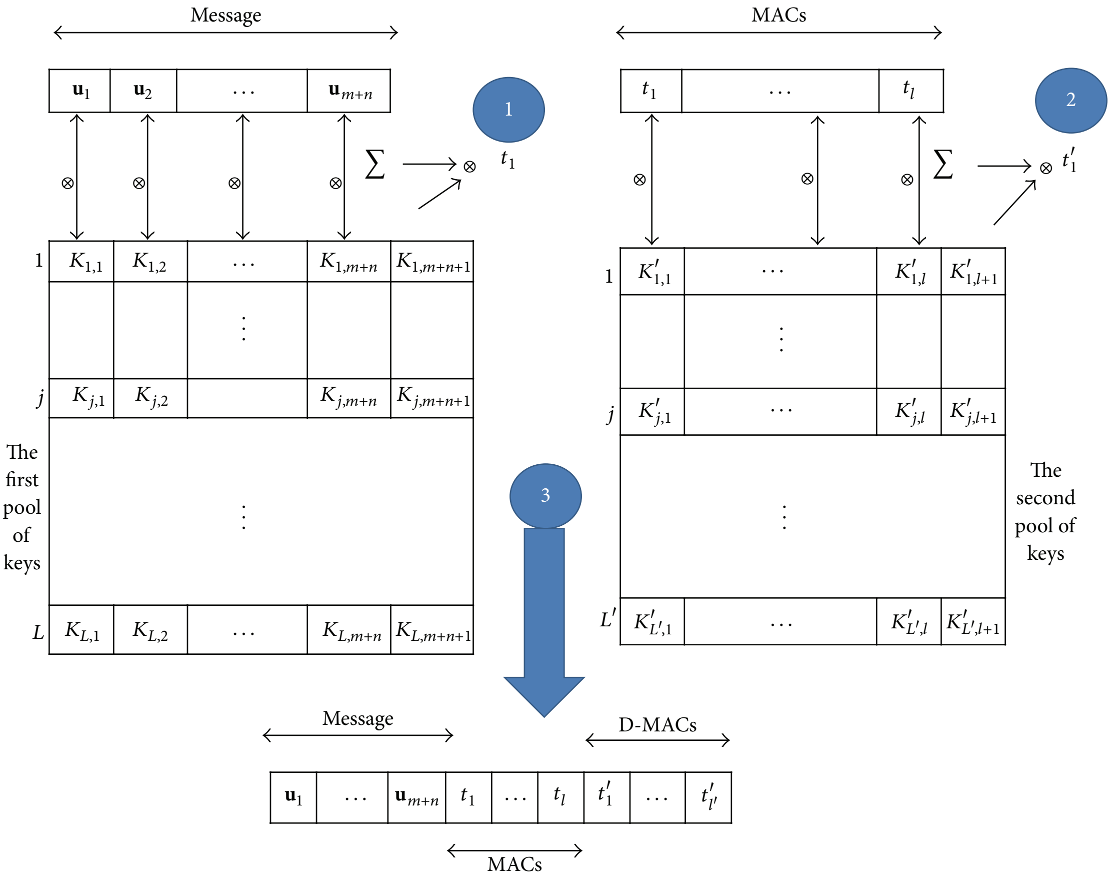


Figure 3: The basic idea of Dual-HMAC scheme which is related to the padding orthogonality property.

4.1.1. Setup

(a) Key Distribution Centre (KDC) distributes the following two sets of secrete key vectors to the source node S: $\mathscr{K}_S = \{\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_L\}$, $\mathbf{k}_l \in \mathbb{F}_p^{m+n+1}$ and $\mathscr{K}'_S = \{\mathbf{k}'_1, \mathbf{k}'_2, \ldots, \mathbf{k}'_{L'}\}$, $\mathbf{k}'_{l'} \in \mathbb{F}_p^{L+1}$. KDC also distributes the following two subsets of secrete vectors to all intermediate nodes and sink nodes: $\mathscr{K}_{n_i} = \{\mathbf{k}_1, \ldots, \mathbf{k}_R\}$, $\mathbf{k}_r \in \mathbb{F}_p^{m+n+1}$, and $\mathscr{K}'_{n_i} = \{\mathbf{k}'_1, \ldots, \mathbf{k}'_{R'}\}$, $\mathbf{k}'_{r'} \in \mathbb{F}_p^{L+1}$.

(b) Note that $\mathscr{K}_S \cap \mathscr{K}'_S = \varnothing$, $\mathscr{K}_{n_i} \subset \mathscr{K}_S$, $\mathscr{K}'_{n_i} \subset \mathscr{K}'_S$, and if $i \neq j$, then $\mathscr{K}_{n_i} \neq \mathscr{K}_{n_j}$ and $\mathscr{K}'_{n_i} \neq \mathscr{K}'_{n_j}$.

(c) Note that the subscriptions l, l', r, and r' represent the index of the key, which do not change in different subsets. In other words, if $\mathbf{k}_l = \mathbf{k}_r$, then $l = r$; if $\mathbf{k}_{l'} = \mathbf{k}_{r'}$, then $l' = r'$.

The detail of probabilistic key distribution algorithm will be described in Section 4.4, as well as how to choose the key set size.

4.1.2. Tag Generation. For every data packet $\mathbf{u}_i \in \mathbb{F}_p^{m+n}$, the source uses the $\mathrm{MAC}(\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_L, \mathbf{u}_i)$ and $\text{D-MAC}(\mathbf{k}'_1, \mathbf{k}'_2, \ldots, \mathbf{k}'_{L'}, t_{\mathbf{u}_i,1}, \ldots, t_{\mathbf{u}_i,L})$ algorithms to generate the $(L + L')$ tags.

4.1.3. Verification. When a relay or sink node receives a coded packet $y \in \mathbb{F}_p^{m+n}$ with its tags, this node checks the correctness of packet y using the algorithm Verify via its predistributed keys $\mathscr{K}_{n_i}$ and $\mathscr{K}'_{n_i}$. If the output is 1, then the received coded packet y is considered to be a correct one. Otherwise, this packet is discarded.

4.1.4. Encoding. When an intermediate node receives h encoded packets $u_i \in \mathbb{F}_p^{m+n}$ $(1 \leq i \leq h)$, and they all are checked or considered to be correct, a forward coded packet along with new tags is generated using the $\mathrm{Combine}((\mathbf{u}_i, t_{\mathbf{u}_i,1}, \ldots, t_{\mathbf{u}_i,L}, t'_{\mathbf{u}_i,1}, \ldots, t'_{\mathbf{u}_i,L'})_{i=1}^{h}, (c_i)_{i=1}^{h})$ algorithm with locally randomly generated coefficients c_i.

4.2. The Construction. In this subsection, we provide the construction of the scheme. Particularly, we present the above-mentioned four algorithms, namely, MAC, D-MAC, Combine, and Verify, in detail. The algorithm MAC computes L tags for each data packet $\mathbf{u}_i$, and the algorithm D-MAC computes another L' tags for the L tags. Moreover, the algorithm Combine generates new tags for RLNC encoded packet. Furthermore, the algorithm Verify checks the correctness of a data packet with all tags. Details of these four algorithms are described as follows.

4.2.1. MAC($\mathscr{K}_S$*,*u_i*)*

(i) *Input*: $\mathscr{K}_S$ consists of L secrete vectors $\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_L \in \mathbb{F}_p^{m+n+1}$, and the ith data packet $\mathbf{u}_i \in \mathbb{F}_p^{m+n}$ as shown in (1).

(ii) *Output*: L tags $t_{\mathbf{u}_i,1}, t_{\mathbf{u}_i,2}, \ldots, t_{\mathbf{u}_i,L} \in \mathbb{F}_p$ for packet $\mathbf{u}_i$, where

$$t_{\mathbf{u}_i,l} = -\frac{\left(\sum_{j=1}^{m+n} u_{i,j} k_{l,j}\right)}{k_{l,m+n+1}}. \tag{3}$$

4.2.2. D-MAC($\mathscr{K}'_S$*,*$t_{u_i,1}$*,. . .,*$t_{u_i,L}$*)*

(i) *Input*: $\mathscr{K}'_S$ consists of L' secrete vectors $\mathbf{k}'_1, \mathbf{k}'_2, \ldots, \mathbf{k}'_{L'} \in \mathbb{F}_p^{L+1}$, and the L tags $t_{\mathbf{u}_i,l} \in \mathbb{F}_p$ generated in (3) for the packet $\mathbf{u}_i$.

(ii) *Output*: L' tags $t'_{\mathbf{u}_i,1}, t'_{\mathbf{u}_i,2}, \ldots, t'_{\mathbf{u}_i,L'} \in \mathbb{F}_p$ for packet $\mathbf{u}_i$, where

$$t'_{\mathbf{u}_i,l'} = -\frac{\left(\sum_{j=1}^{L} t_{\mathbf{u}_i,j} k'_{l',j}\right)}{k_{l',L+1}}. \tag{4}$$

4.2.3. Combine((u_i*,*$t_{u_i,1}$*,. . .,*$t_{u_i,L}$*,*$t'_{u_i,1}$*,. . .,*$t'_{u_i,L'}$*)*$_{i=1}^{h}$*,(*c_i*)*$_{i=1}^{h}$*)*

(i) *Input*: h vectors $\mathbf{u}_i \in \mathbb{F}_p^{m+n}$ $(1 \le i \le h)$ as shown in (2), each with $(L+L')$ tags $t_{u_i,1}, \ldots, t_{u_i,L}, t'_{u_i,1}, \ldots, t'_{u_i,L'} \in \mathbb{F}_p$ $(1 \le i \le h)$, and h constants $c_1, \ldots, c_h \in \mathbb{F}_p$.

(ii) *Output*: a coded packet with $(L + L')$ tags generated by

$$\begin{aligned} &\left(y, t_{y,1}, \ldots, t_{y,L}, t'_{y,1}, \ldots, t'_{y,L'}\right) \\ &\quad = \sum_{i=1}^{h} c_i \left(u_i, t_{u_i,1}, \ldots, t_{u_i,L}, t'_{u_i,1}, \ldots, t'_{u_i,L'}\right). \end{aligned} \tag{5}$$

4.2.4. Verify($\mathscr{K}_{n_i}$*,*$\mathscr{K}'_{n_i}$*,(*y*,*$t_{y,1}$*,. . .,*$t_{y,L}$*,*$t'_{y,1}$*,. . .,*$t'_{y,L'}$*))*

(i) *Input*: $\mathscr{K}_{n_i}$ consists of secrete vectors $\mathbf{k}_1, \ldots, \mathbf{k}_R \in \mathbb{F}_p^{m+n+1}$, $\{\mathbf{k}_1, \ldots, \mathbf{k}_R\} \subset \{\mathbf{k}_1, \ldots, \mathbf{k}_L\}$ and $\mathscr{K}'_{n_i}$ consists of secrete vectors $\mathbf{k}'_1, \ldots, \mathbf{k}'_{R'} \in \mathbb{F}_p^{L+1}$, $\{\mathbf{k}'_1, \ldots, \mathbf{k}'_{R'}\} \subset \{\mathbf{k}'_1, \mathbf{k}'_2, \ldots, \mathbf{k}'_{L'}\}$, as well as a RLNC encoded packet $y \in \mathbb{F}_p^{m+n}$ with $(L+L')$ tags $t_{y,1}, \ldots, t_{y,L}, t'_{y,1}, \ldots, t'_{y,L'} \in \mathbb{F}_p$.

(ii) *Output*: if

$$\forall \mathbf{k}_r \in \{\mathbf{k}_1, \ldots, \mathbf{k}_R\}, \quad \delta_r = \left(\sum_{j=1}^{m+n} y_j k_{r,j}\right) + t_{y,r} k_{r,m+n+1} = 0, \tag{6}$$

$$\forall \mathbf{k}'_{r'} \in \{\mathbf{k}'_1, \ldots, \mathbf{k}'_{R'}\}, \quad \delta'_{r'} = \left(\sum_{j=1}^{L} t_{y,j} k'_{r',j}\right) + t'_{y,r'} k'_{r',L+1} = 0 \tag{7}$$

then output 1; otherwise output 0.

4.3. The Correctness of Dual-HMAC. The proposed Dual-HMAC is going to be correct if the Verify algorithm could produce output 1 for the valid data packet and the valid RLNC encoded packet. Rigorously speaking, it should satisfy the following condition:

> If Verify($\mathscr{K}_{n_i}$, $\mathscr{K}'_{n_i}$, ($\mathbf{u}_i$, MAC($\mathscr{K}_S$, $\mathbf{u}_i$), D-MAC($\mathscr{K}'_S$, $\mathbf{u}_i$)) = 1 and
> If Verify($\mathscr{K}_{n_j}$, $\mathscr{K}'_{nj}$, ($\mathbf{u}_j, t_{\mathbf{u}_j,1}, \ldots, t_{\mathbf{u}_j,L}, t'_{\mathbf{u}_j,1}, \ldots, t'_{\mathbf{u}_j,L'}$)) = 1, $\forall j = 1, \ldots, h$, then
> Verify($\mathscr{K}_{n_j}$, $\mathscr{K}'_{n_j}$, Combine(($\mathbf{u}_j, t_{\mathbf{u}_j,1}, \ldots, t_{\mathbf{u}_j,L}$, $t'_{\mathbf{u}_j,1}$, . . ., $t'_{\mathbf{u}_j,L'}$)$_{j=1}^{h}$, $(c_j)_{j=1}^{h}$)) = 1.

Theorem 1. *The first verification is correct.*

Proof. According to the description of "*Setup*," since $\mathscr{K}_{n_i} \subset \mathscr{K}_S$, $\forall \mathbf{k}_r \in \mathscr{K}_{n_i}$, there is a key $\mathbf{k}_l = \mathbf{k}_r$ and $l = r$. As a result, by letting $y = \mathbf{u}_i$, $\mathbf{k}_l = \mathbf{k}_r$, and substituting $t_{y,r} = t_{\mathbf{u}_i,l}$ using (3), it is for sure that (6) $\delta_r = 0$, $\forall \mathbf{k}_r \in \mathscr{K}_{n_i}$.

Similarly, since $\mathscr{K}'_{n_i} \subset \mathscr{K}'_S$, $\forall \mathbf{k}'_{r'} \in \mathscr{K}'_{n_i}$, there is a key $\mathbf{k}_{l'} = \mathbf{k}_{r'}$ and $l' = r'$. As a result, by letting $y = \mathbf{u}_i$, $\mathbf{k}_{l'} = \mathbf{k}_{r'}$, and substituting $t'_{y,r'} = t'_{\mathbf{u}_i,l'}$ using (4), it is for sure that (7) $\delta'_{r'} = 0$, $\forall \mathbf{k}'_{r'} \in \mathscr{K}'_{n_i}$.

As a result, Verify($\mathscr{K}_{n_i}$, $\mathscr{K}'_{n_i}$, ($\mathbf{u}_i$, MAC($\mathscr{K}_S$, $\mathbf{u}_i$), D-MAC($\mathscr{K}'_S$, $\mathbf{u}_i$)) = 1. □

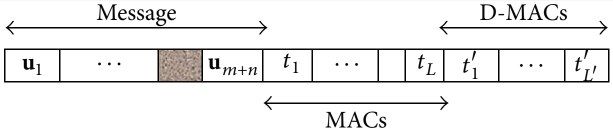
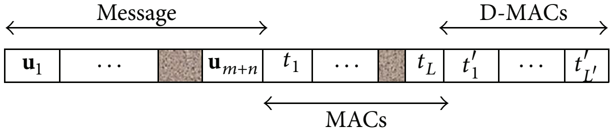
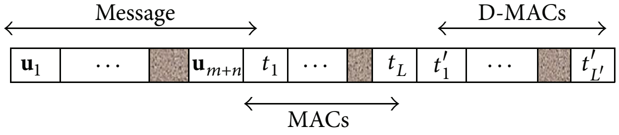
Theorem 2. *The second verification is correct.*

Proof. According to (3), the tags are created in a way that $\mathbf{k}_l$ is orthogonal to the concatenation of $\mathbf{u}_i$ and its l tags $\mathbf{k}_l \perp (\mathbf{u}_i \parallel t_{\mathbf{u}_i,l})$. This relationship is true for all packets in the same generation, which represents as $\mathbf{k}_l \perp \mathrm{Span}\{(\mathbf{u}_i \parallel t_{\mathbf{u}_i,l})_{i=1}^{m}\}$.

If Verify($\mathscr{K}_{n_j}$, $\mathscr{K}'_{nj}$, ($\mathbf{u}_j, t_{\mathbf{u}_j,1}, \ldots, t_{\mathbf{u}_j,L}, t'_{\mathbf{u}_j,1}, \ldots, t'_{\mathbf{u}_j,L'}$)) = 1, $\forall j = 1, \ldots, h$, then $\mathbf{k}_l \perp \mathrm{Span}\{(\mathbf{u}_j \parallel t_{\mathbf{u}_j,l})_{j=1}^{h}\}$. Moreover, the algorithm Combine is a linear combination, which does not change the subspace. Hence, $(y \parallel t_{y,l}) \in \mathrm{Span}\{(\mathbf{u}_j \parallel t_{\mathbf{u}_j,l})_{j=1}^{h}\}$, $\mathbf{k}_l \perp (y \parallel t_{y,l})$, and (6) $\delta_r = 0$, $\forall \mathbf{k}_r \in \mathscr{K}_{n_i}$. Similarly, we could derive $\delta'_{r'} = 0$, $\forall \mathbf{k}'_{r'} \in \mathscr{K}'_{n_i}$. As a result, the output of Verify is 1, and the second condition is also correct. □

4.4. Key Distribution Model. In our work, we assume that a set of symmetric keys are distributed to all participant nodes in a secure and authenticated manner through a key

Table 1: Security analysis.

Adversary behaviour	Adversary behaviour description	Max hops
Data pollution	 Message D-MACs $\mathbf{u}_1$ $\cdots$ $\mathbf{u}_{m+n}$ t_1 $\cdots$ t_L t'_1 $\cdots$ $t'_{L'}$ MACs	$c-1$
Tag pollution (MAC)	 Message D-MACs $\mathbf{u}_1$ $\cdots$ $\mathbf{u}_{m+n}$ t_1 $\cdots$ t_L t'_1 $\cdots$ $t'_{L'}$ MACs	$c-1$
Dual-tag pollution (D-MAC)	 Message D-MACs $\mathbf{u}_1$ $\cdots$ $\mathbf{u}_{m+n}$ t_1 $\cdots$ t_L t'_1 $\cdots$ $t'_{L'}$ MACs	$N-i$

A representation of adversary behavior. Max hops shows the maximum hops which the polluted packet can travel through the network. We consider the worst case when the adversary node can have access to all the next $c-1$ nodes keys (say, a chain of c compromised nodes; moreover, $(N-i)$ is the maximum number of hops between the adversary and the sink node).

distribution scheme. In this section, we provide the model of the key distribution scheme that we have adopted.

A set system is a pair (X, F), where X is a finite set of elements and F is an ordered set of subsets of X.

Definition A. A set system (X, F) is called a c cover-free family (c-CFF) if, for any c sets $A_1, A_2, \ldots, A_c \in F$ and any other set $B \in F$, one has

$$B \not\subseteq \bigcup_{J=1}^{c} A_i. \tag{8}$$

Definition B. A set system (X, F) is called a (c, d) cover-free family ((c, d)-CFF) if, for any c sets $A_1, A_2, \ldots, A_c \in F$ and any other set $B \in F$, one has

$$\left| B \setminus \bigcup_{J=1}^{c} A_j \right| > d. \tag{9}$$

We believe that if key assignment is done properly, no coalition of c nodes can fool another node. Our key predistribution scheme considers two key pools which are available at the source node. We use all these two sets for checking the message's integrity. It is shown in [25] that, in order to resist against less than c compromised nodes, the cardinality of the key size at the source must be $(c+1)$ times larger than those key sizes at the relay/sink nodes, which could be expressed as $L \geq (c+1)R$, and $L' \geq (c+1)R'$. More specifically, we claim that, by this model, we can be sure each verifier has some shared keys which can verify the integrity of messages. Our scheme assigns two sets $\mathcal{K}_S = \{\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_L\}$, $\mathbf{k}_l \in \mathbb{F}_p^{m+n+1}$, $\mathcal{K}'_S = \{\mathbf{k}'_1, \mathbf{k}'_2, \ldots, \mathbf{k}'_{L'}\}$, $\mathbf{k}'_{l'} \in \mathbb{F}_p^{L+1}$ to a source node where each has $L = L' = |\mathcal{K}_S| = |\mathcal{K}'_S| = e(c+1)\ln(1/q)$ keys. Moreover, we distribute two sets $\mathcal{K}_{n_i} = \{\mathbf{k}_1, \ldots, \mathbf{k}_R\}$, $\mathbf{k}_r \in \mathbb{F}_p^{m+n+1}$, and $\mathcal{K}'_{n_i} = \{\mathbf{k}'_1, \ldots, \mathbf{k}'_{R'}\}$, $\mathbf{k}'_{r'} \in \mathbb{F}_p^{L+1}$ to each node where each of these sets has $R = R' = |\mathcal{K}_{n_i}| = |\mathcal{K}'_{n_i}| = e\ln(1/\sqrt{q})$ keys. In addition, q is the security parameter and it stands as $q = \left(\mathrm{N} \cdot \binom{N}{c}\right)^{-1}$, where N is the total number of recipients (usually $q = 10^{-3}$ would be sufficient) and c is a fixed number of corrupted users. Every key $\mathbf{k}_r$ or $\mathbf{k}'_{r'}$ is included in a set of each verifier with probability of $1/2(c+1)$ (for simplicity, we consider the same length for both poll keys).

5. Security Analysis

In this section, we analyse the possibilities of an adversary launching pollution attack successfully. Hence, three possible pollution attack behaviours are analysed, and their security levels are quantified, respectively. We summarize our analysis in Table 1. Finally, the possible knowledge that an adversary can access is analysed.

5.1. The Resistance against Pollution Attack. We consider the following three possible pollution attack scenarios.

5.1.1. 1st Pollution Behaviour: Data Pollution Attack. An adversary node intends to make changes in a message. This polluted message can be detected immediately by the next hop, or in a few hops later (maximum $c-1$ hops later). It means the polluted message should travel some hops to be detected. The worst case happens whenever an adversary can do coalition with the other $c-1$ compromised nodes.

5.1.2. 2nd Pollution Behaviour: Tag (i.e., MAC) Pollution Attack. An adversary node intends to make changes in the MACs. This attack can be detected immediately by the next hop if he cannot compromise the next hop and get the next hop keys, or in a few hops later (maximum $c-1$ hops later). It is similar to the 1st pollution behaviour.

5.1.3. 3rd Pollution Behaviour: Dual-Tag (i.e., D-MAC) Pollution Attack. An adversary i which is one of the N legitimate

Table 2: Communication and computation overhead of Dual-HMAC and the two related works in [14] and [9].

	[14]	MacSig [9]	Dual-HMAC
Type of key	Symmetric	Hybrid	Symmetric
Data pollution	Yes	Yes	Yes
Tag pollution	No	Yes	Yes
Communication overhead	l	$l + 1 + (l \text{ key indexes})$	l
Computation overhead	$l\,(m+n)$	$\frac{3}{2}\lvert p\rvert(m+l+1)+(m+n+1)l$	$\frac{l}{2}\left(m+n+\frac{l}{2}+2\right)$

nodes ($(N-i)$ is the maximum number of hops between the adversary and the sink node) intends to make changes in D-MACs. This attack can be detected by the next hop by the probability of $1/2(c+1)$. The polluted D-MAC can travel some hops, before it is detected, by the probability of $[1-1/2(c+1)]^{N-i}$ (see Theorem 3).

Theorem 3. *The probability that a downstream user receives the same key as key$_x$ of user x is not less than* $[1-1/2(c+1)]^{N-i}$.

Proof. The probability of recovering the same shared key between two nodes is $1/2(c+1)$, so the probability of the key$_x$ from a user x located in next hop $x+1$ is $1/2(c+1)$, and this probability would be $(1/2(c+1))^2$ if this key is found in two hops later. In other words, if the total number of hops between an intermediate node and a sink node, named as downstream node, is $(N-i)$, then the probability that a downstream user receives the same key as key$_x$ of user x is not less than $[1-1/2(c+1)]^{N-i}$. □

5.2. The Possible Knowledge of an Adversary. As outlined in Section 3.2, we assume the source node is trustworthy and the process of key predistribution is secure. Hence, the secret key sets, $\mathscr{K}_S$ and $\mathscr{K}'_S$, assigned to the source node are considered secure. However, an adversary can wiretap all the data communication in a network and may compromise several relay or sink nodes. Hence, the adversary can get access to the received packets from the previous hops, the key information distributed to him by the KDC, and the key information stored at the compromised nodes.

6. Performance Evaluation

In this section, communication and computation overhead of the proposed Dual-HMAC scheme are presented.

6.1. Communication Overhead. In our scheme, each source message has $m+n$ codewords and each codeword has $\log_2 p$ bits. So, the bit-length of a source message is $(m+n)\log_2 p$. The L MACs and the L' D-MACs are appended to the source message. We consider $\log_2 p$ symbols for each MAC and D-MAC. By fixing $L+L'=l$, our scheme has the following communication overhead:

$$\frac{\left(L+L'\right)\log_2 p}{(m+n)\log_2 p}=\frac{L+L'}{m+n}=\frac{l}{m+n}. \qquad (10)$$

Hence, the Dual-HMAC scheme's overhead is equal to the scheme presented in [14]. However this scheme is vulnerable to tag pollution. Additionally, the bandwidth overhead between our scheme and the scheme proposed in [9] is given by the following equation:

$$\begin{aligned}\frac{O_{b_{\text{Dual-HMAC}}}}{O_{b_{\text{MacSig}}}}&=\frac{l/(m+n)}{(l+1)/(m+n)+32l/128(m+n)}\\&=\frac{1}{1.25+1/l}.\end{aligned} \qquad (11)$$

Base on (11), the bandwidth overhead comparison between our scheme and the scheme described in [9], for different values of c, is depicted in the graphs included in Figure 4.

6.2. Computation Overhead. Recall that, to append L MACs and L' D-MACs, $L(m+n+1)$ and $L'(L+1)$ multiplication operations for MACs and D-MACs are needed, respectively. In addition, for verification, we consider that all the relay nodes have shared keys and each node needs to verify the message. So, as shown in (6) and (7), our scheme needs $L(m+n+1)$ multiplications for MACs and $L'(L+1)$ multiplications for D-MACs.

7. Discussion

The number of appended tags which is needed in [14] is l. However, the total number of tags (homomorphic MACs and a signature) which is used in [9] is $l+1+(l$ key indexes).

Moreover, the works in [9, 14] need to use $l(m+n)$ and $(3/2)|p|(m+l+1)+(m+n+1)l$ multiplications for the verification phase, respectively. Table 2 summarizes the communication and computation overhead of our work (i.e., Dual-HMAC) and the two related works in [9, 14].

To conclude this section, consider the following.

(i) Dual-HMAC incurs a relatively low communication overhead which is almost equal to [14]; however, our scheme's overhead, according to (11), saves up to $(1.25+1/l)$ in comparing to MacSig approach [9]. This means that our scheme provides at least 25% bandwidth saving.

(ii) Dual-HMAC not only slightly increases the tag pollution resistance, but also decreases the computation overhead compared to the schemes in [9, 14].

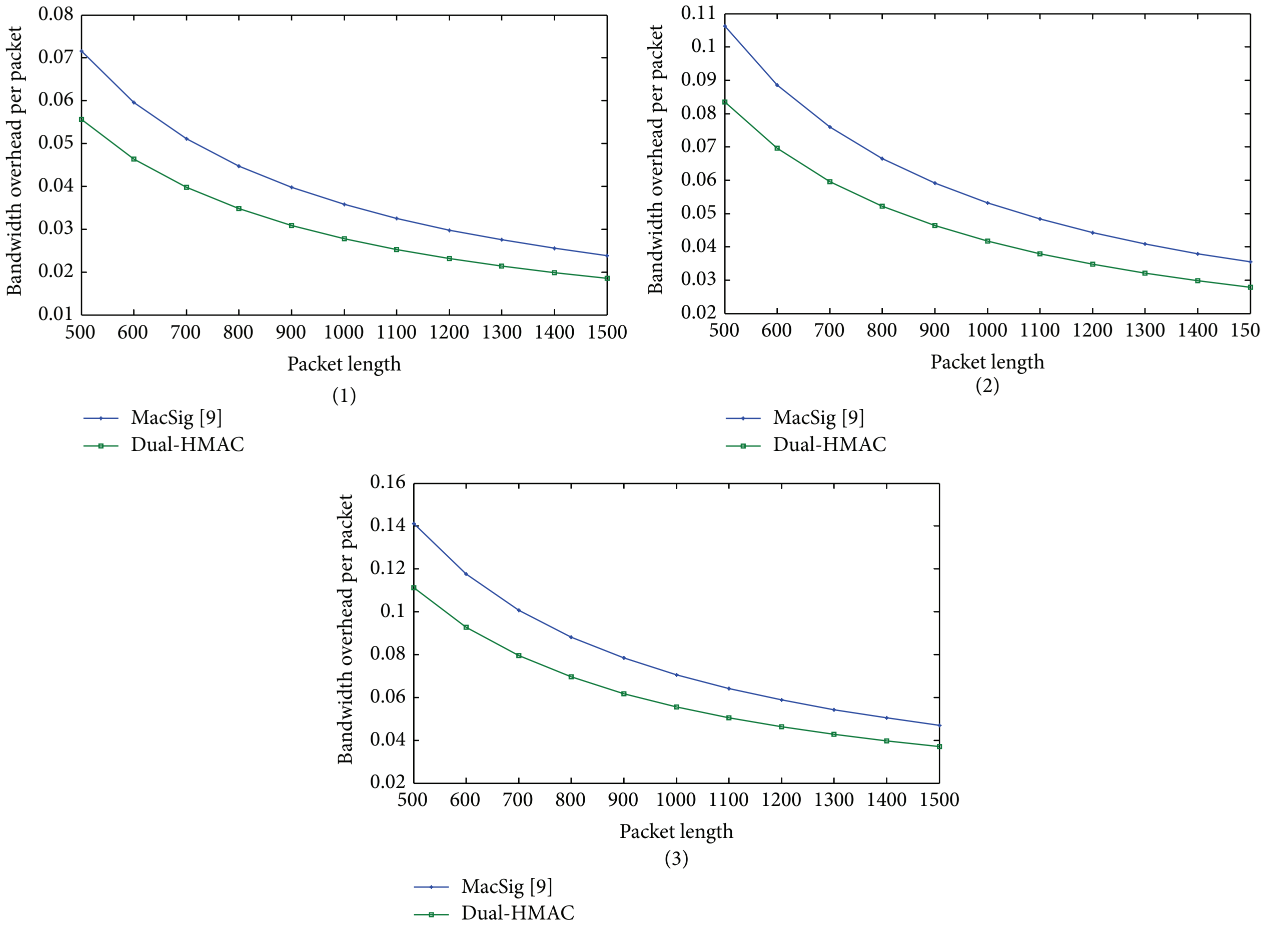


Figure 4: The bandwidth overhead comparison. By fixing $\delta = 0.1$, $n = 20$ m, $\varepsilon = 0.01$ [9], we depict three scenarios (1), (2), and (3) and we use $c = 1, 2, 3$ for each of them, respectively.

(iii) Dual-HMAC provides resistance for half of the tags against tag pollution attack. This results in reducing the possibility of tag pollution around 50% compared to the work presented in [14].

(iv) Dual-HMAC uses symmetric keys, in contrast to MacSig scheme using symmetric and public keys.

8. Conclusion

Network coding-enabled WSNs are susceptible to pollution attack where a small number of polluted messages can corrupt bunches of legitimate messages. This paper has presented a homomorphic MAC-based scheme, called Dual-HMAC, for network coding-enabled WSNs. The proposed scheme makes use of two types of tags to provide resistance against data pollution attacks and partially tag pollution attacks. The partial protection against tag pollution is related to the key predistribution model. Furthermore, our proposed scheme presents low communication overhead and low computational overhead compared to other existing schemes. Finally, as future work, we plan to propose a scheme that will mitigate not only partially tag pollution attacks but also tag pollution attacks.

Notations

m:	The number of packets per generation
$\underline{\mathbf{u}}_i$:	The ith source message
n:	The number of symbols of each packet
p:	A finite field size (say, 2^8)
q:	A security parameter (say, 10^{-3})
L:	The number of MACs
L':	The number of D-MACs
l:	The number of source keys ($l = l_1 + l_2$)
t_i:	The ith MAC
t'_i:	The ith D-MAC
N:	The number of verifiers
c:	The number of compromised nodes
$\mathscr{K}_S, \mathscr{K}'_S$:	The two polls of keys of source
$\mathscr{K}_{n_i}, \mathscr{K}'_{n_i}$:	The two polls of keys of ith node.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The research leading to these results has received funding from National Funds through FCT, Fundação para a Ciência e a Tecnologia, under the project PTDC/EEA-TEL/119228/2010 SMARTVISION, and from the European Community's Seventh Framework Programme [FP7/2007–2013] under Grant Agreement no. 285969 [CODELANCE].

References

[1] J. Sen, "A survey on wireless sensor network security," *International Journal of Communication Networks and Information Security*, vol. 1, no. 2, pp. 59–82, 2009.

[2] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[3] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1906–1918, 2005.

[4] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proceedings of the ACM SIGCOMM Conference on Computer Communications*, vol. 37, pp. 169–180, August 2007.

[5] D. Petrovic, K. Ramchandran, and J. Rabaey, "Overcoming untuned radios in wireless networks with network coding," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2649–2657, 2006.

[6] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 4, pp. 2235–2245, IEEE, Miami, Fla, USA, March 2005.

[7] T. Ho, M. Médard, R. Koetter et al., "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[8] M. N. Krohn, M. J. Freedman, and D. Mazieres, "On-the-fly verification of rateless erasure codes for efficient content distribution," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 226–240, 2004.

[9] Z. Peng, J. Yixin, L. Chuang, Y. Hongyi, A. Wasef, and S. Xuemin, "Padding for orthogonality: efficient subspace authentication for network coding," in *Proceedings of the IEEE INFOCOM*, pp. 1026–1034, 2011.

[10] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. R. Karger, "Byzantine modification detection in multicast networks with random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2798–2803, 2008.

[11] S. Jaggi, M. Langberg, S. Katti et al., "Resilient network coding in the presence of Byzantine adversaries," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2596–2603, 2008.

[12] C. Cheng, T. Jiang, and Q. Zhang, "TESLA-based homomorphic MAC for authentication in P2P system for live streaming with network coding," *IEEE Journal on Selected Areas in Communications*, vol. 31, pp. 291–298, 2013.

[13] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient scheme for securing XOR network coding against pollution attacks," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 406–414, April 2009.

[14] S. Agrawal and D. Boneh, "Homomorphic MACs: MAC-based integrity for network coding," in *Applied Cryptography and Network Security: 7th International Conference, ACNS 2009, Paris-Rocquencourt, France, June 2–5, 2009. Proceedings*, vol. 5536 of *Lecture Notes in Computer Science*, pp. 292–305, Springer, Berlin, Germany, 2009.

[15] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a linear subspace: signature schemes for network coding," in *Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography (PKC '09)*, Irvine, Calif, USA, 2009.

[16] C. Gkantsidis and P. R. Rodriguez, "Cooperative security for network coding file distribution," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, pp. 1–13, Barcelona, Spain, April 2006.

[17] D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," *International Journal of Information and Coding Theory*, vol. 1, no. 1, pp. 3–14, 2009.

[18] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature-based scheme for securing network coding against pollution attacks," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 2083–2091, April 2008.

[19] E. Kehdi and L. Baochun, "Null keys: limiting malicious attacks via null space properties of network coding," in *Proceedings of the IEEE INFOCOM*, pp. 1224–1232, Rio de Janeiro, Brazil, April 2009.

[20] Y. Li, H. Yao, M. Chen, S. Jaggi, and A. Rosen, "RIPPLE authentication for network coding," in *Proceedings of the IEEE INFOCOM*, pp. 1–9, March 2010.

[21] B. C. Neuman and T. Ts'o, "Kerberos: an authentication service for computer networks," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 33–38, 1994.

[22] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 2, pp. 228–258, 2005.

[23] W. Xiaohu, X. Yinlong, Y. Chau, and X. Liping, "A tag encoding scheme against pollution attack to linear network coding," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 33–42, 2014.

[24] P. Chou, Y. Wu, and A. K. Jain, "Practical network coding," in *Proceedings of the 41st Annual Allerton Conference on Communication, Control, and Computing*, October 2003.

[25] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast security: a taxonomy and some efficient constructions," in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societie (INFOCOM '99)*, vol. 2, pp. 708–716, New York, NY, USA, March 1999.