Lightweight Integrity for XOR Network Coding in Wireless Sensor Networks

Kazuya Izawa, Atsuko Miyaji and Kazumasa Omote

Japan Advanced Institute of Science and Technology (JAIST)
{s0910201,miyaji,omote}@jaist.ac.jp

Abstract. In INFOCOM 2009, Yu, Wei, Ramkumar and Guan have proposed the novel mechanism (called Yu's scheme), in which a forwarder can filter polluted messages before spreading the pollution in the XOR network coding systems. In order to perform such filtering, two or more message authentication codes (MACs) are used for this scheme. However, Yu's scheme has a problem that the number of MACs increases at every coding point, since it cannot operate MACs with the XOR network coding. This means that the MAC of Yu's scheme does not have homomorphic property.
In this paper, we propose the first symmetric-key-based scheme not only to filter polluted messages but also to operate MACs with the XOR network coding on a forwarder. The XOR network coding of MACs produces improvement which does not increase the number of MACs at a coding point. Our scheme uses the UHFs-based MAC with a homomorphic property to hold homomorphic MAC, and hence it can aggregate MACs in our XOR network coding systems. We emphasize that a forwarder cannot straightforward filter polluted messages even if our scheme uses the UHFs-based MACs.

1 Introduction

Wireless Sensor Networks (WSNs) consist of small, battery-operated, limited memory and limited computational power sensor nodes. Most of existing secure schemes in WSNs are not based on public key cryptography. More importantly, reducing communication traffic is desirable to save the energy of the relay nodes (forwarders). It is especially necessary to reduce the amount of useless communication. For instance, it is important to remove polluted messages quickly or to conduct data aggregation.

Unlike the traditional message forwarding approaches, network coding [1] allows forwarders to combine multiple input messages into one or more encoded ones. This technique has novel advantages to maximize network throughput and to reduce the number of retransmissions. While a network coding is normally operated over large finite fields (called normal network coding), we focus on a special network coding based only on XOR operations, named *XOR network coding* [3, 6, 10, 11, 13, 16–18]. It is easy to apply XOR network coding to wireless networks such as WSNs, owing to its simplicity.

Network coding systems are vulnerable to *pollution attacks*, in which adversaries inject polluted messages into the systems on the compromised forwarders. In a worst case scenario, a single corrupted message can end up corrupting all the information reaching a destination. These attacks not only prevent the sinks from recovering the source messages but also drain out the energy of the forwarders. Hence, it is crucial to filter polluted messages in network coding systems as early as possible. In order to achieve such filtering in the XOR network coding systems, Yu, Wei, Ramkumar and Guan [16] have proposed the novel mechanism (called Yu's scheme), in which a forwarder can filter polluted messages before spreading the pollution. In this scheme, two or more message authentication codes (MACs) are used to filter polluted messages.

However, Yu's scheme has a problem that the number of MACs increases at every coding point, since it cannot operate MACs with the XOR network coding. This means that the MAC of Yu's scheme does not have homomorphic property. If two or more MACs are operated with XOR network coding, their MACs are aggregated (encoded) to one MAC. Otherwise, as Yu's scheme, their MACs are just forwarded to downstream nodes without XOR network coding. It is meaningless that in spite of coding a message, their MACs is not aggregated (encoded). On the other hand, Apavatjrut et al. [3] have simply applied the homomorphic MAC based on universal hash functions (UHFs) to the XOR network coding systems, in which two or more MACs are operated with XOR network coding. However, a forwarder cannot filter polluted messages in this scheme. More particularly, a forwarder cannot verify the MACs for filtering since the UHFs-based MAC has one-time pad. Only a sink can verify the encoded (aggregated) MACs since it knows all the seeds which generate one-time pad.

In this paper, we propose the first symmetric-key-based scheme not only to filter polluted messages but also to operate MACs with the XOR network coding on a forwarder. The XOR network coding of MACs produces improvement which does not increase the number of MACs at a coding point. Our scheme uses the UHFs-based MAC with a homomorphic property to hold homomorphic MAC, and hence it can aggregate MACs in our XOR network coding systems. As a result, it can reduce the amount of extra space associated with communication complexity for integrity protection. We emphasize that a forwarder cannot straightforward filter polluted messages even if our scheme uses the UHFs-based MAC. Our scheme improves how to generate a pseudo-random function (PRF) in the UHFs-base MAC so that a forwarder can filter polluted messages.

2 Related Work

Working on network coding started with the pioneering paper by Ahlswede et al. [1], which established the value of coding in the routers and provided theoretical bounds on the capacity of such networks. Network coding systems can be divided into two classes of normal and XOR network coding [16]. There are several lightweight authentication schemes for both of network coding, based on symmetric-key-cryptography such as message authentication codes (MACs).

For a lightweight authentication scheme in a normal network coding against pollution attacks, the homomorphic MAC [2] and RIPPLE [12] have been proposed so far. Agrawal and Boneh [2] design a homomorphic MAC which allows checking the integrity of normal network encoded data. It converts a homomorphic MAC into a broadcast homomorphic MAC, in which a forwarder can verify the integrity of MACs. Li et al. [12] have proposed a symmetric-key-based scheme for network coding authentication (named RIPPLE). Despite using symmetric-key-based homomorphic MAC algorithms, RIPPLE achieves asymmetry by delayed disclosure of the MAC keys, inspired by TESLA [14]. While these schemes focus on normal network coding systems, the following two recent schemes [16, 3] focus on a lightweight authentication in a XOR network coding against pollution attacks, which is more suitable for WSNs.

Yu's scheme [16] exploits probabilistic key pre-distribution and MACs. In this scheme, the source node generates multiple MACs for each message using its secret keys, where each MAC can authenticate only a part of the message and the parts authenticated by different MACs are overlapped. Every encoded message is attached with the MACs of the source messages. Therefore, multiple downstream forwarders can collaboratively verify different parts of the encoded message using the MACs and their own shared keys. However, Yu's scheme has a problem that the number of MACs increases at every coding point, since it cannot operate MACs with the XOR network coding. The details of Yu's scheme will be described in Section 4.

Apavatjrut et al. [3] have naively applied the homomorphic MAC based on UHFs to the XOR network coding systems. Such a XOR homomorphic MAC is given by $MAC_k(M) = h_k(M) \oplus r$, where h_k is a homomorphic UHF with the secret key k, M is a message, and r is one-time pad. However, a forwarder cannot filter the polluted messages in this scheme. We explain this reason as follows. We assume that a forwarder F_3 is connected with two upstream nodes F_1 and F_2, for example. Let M_1 and M_2 denote the source messages of s_1 and s_2, respectively. F_3 receives M_1, M_2, $MAC_k(M_1) = h_k(M_1) \oplus r_1$ and $MAC_k(M_2) = h_k(M_2) \oplus r_2$ from F_1 and F_2. Then, F_3 can compute the encoded message $M_1 \oplus M_2$ and its MAC ($MAC_k(M_1 \oplus M_2) = h_k(M_1 \oplus M_2) \oplus (r_1 \oplus r_2)$). However, F_3 cannot compute $r_1 \oplus r_2$, since $r_1 \oplus r_2$ is random number generated by s_1 and s_2. Therefore, in this scheme, a forwarder cannot filter the polluted messages because it cannot verify the MACs.

3 Preliminaries

3.1 Requirements

The following requirements need to be considered when designing a lightweight integrity of XOR network coding systems in WSNs.

Early filtering of polluted messages. Network coding systems (including XOR and normal network coding) suffer from pollution propagation, i.e., a small number of polluted messages can quickly propagate in the systems and infect

a large proportion messages. When a forwarder receives a polluted message, all of its encoded messages will be polluted. Then, these polluted messages are further used by downstream forwarders for encoding, thus, more messages will be polluted. It is therefore necessary to filter polluted messages as early as possible.
Encoding of MACs. The MAC is computed from the source message. The forwarder, who is not directly connected with source nodes, cannot obtain the source message but can obtain only the encoded messages. So, it is necessary for a forwarder to verify the MAC of encoded messages in order to filter the polluted messages. Hence, the encoding of MACs is essential for a forwarder to check the integrity of encoded messages. This MAC encoding also has an advantage of traffic reduction.
Restricted resources. It is required that the WSNs consist of small, battery-operated devices with limited memory and limited computational power. XOR network coding and the symmetric-key-based MAC are more suitable for such resource-constrained WSNs.

3.2 Notation

We explain the following common notations in the paper:

Symbol	Explanation
n	the number of source messages transmitted ($n \geq 2$)
m	the number of codewords of each message
M_i, $m_{i,j}$	i-th source message and its j-th codeword; $M_i = (m_{i,1} \cdots m_{i,m})$
t	the number of random keys each node has
u	the number of codeword hashed in each MAC
q	security parameter (e.g., $q = 128$)
$\mathcal{K}_{\mathsf{UHF}}$, $\mathcal{K}'_{\mathsf{PRF}}$	global key pools for UHF and PRF
$k_{s,i}$, $k'_{s,i}$	i-th (q-bit) keys of the source node; $k_{s,i} \in \mathcal{K}_{\mathsf{UHF}}$, $k'_{s,i} \in \mathcal{K}_{\mathsf{PRF}}$
sid	the session ID ($sid \in \{0,1\}^q$)
mid	the set of message indeces
h_k	universal hash function using key k: $\{0,1\}^* \mapsto \{0,1\}^q$
$f_{k'}$	pseudo-random function family indexed by the key k': $\{0,1\}^* \mapsto \{0,1\}^q$
g	pseudo-random permutation function: $[1,m] \to [1,m]$
H	Non-cryptographic hash function: $\{0,1\}^q \to [1,m]$

3.3 System and Network Assumptions

We consider a general multicast network in which there are one source node, multiple sinks (receivers) and a number of forwarders. The source node sends n messages $M_1, \ldots, M_n$ in every unit of time, that is, session (the source can actually generate messages continuously). A forwarder can use XOR network coding technique to generate and forward the encoded messages.

In XOR network coding for n source messages $M_1, \ldots, M_n$, an encoded message can be represented as $E = \alpha_1 M_1 \oplus \cdots \oplus \alpha_n M_n$, where $\alpha_i \in \{0,1\}$ for

$i = 1, \ldots, n$. The bit string $(\alpha_1 \cdots \alpha_n)$ is called the encoding vector of E. Of course, M_i can be the encoded message. We adopt the model used in [8] and divide each message into m codewords of the same length. Our scheme partitions codewords only for constructing MACs.

We also assume that all of the nodes have been assigned some random secret keys using the probabilistic key pre-distribution schemes such as [7]. In particular, we assume that each node picks a fixed number of keys randomly from a large global key pool. By carefully controlling the key pool size and the number of keys that each node picks, we assure that any two nodes have certain probability to find some shared keys. The source node uses its keys to generate MACs for its messages, while each forwarder or sink verifies the MACs of received messages using their shared keys with the source node.

3.4 Threat Model [16]

We assume that the source and multiple sinks are always trusted, but the forwarders can be compromised. The adversaries can fully control the compromised forwarders and launch pollution attacks. In such attacks, they may either pollute the output messages of the compromised nodes, or inject the forged messages into systems. Formally speaking, we identify that an encoded message E has been polluted or forged, if and only if its content is not consistent with its encoding vector, for example, $E \neq \alpha_1 M_1 \oplus \alpha_2 M_2 \oplus \cdots \alpha_n M_n$ for n source messages $M_1, \ldots, M_n$.

3.5 Universal Hash Functions (UHFs)

Following Carter and Wegman [5], a universal hash function (UHF) is a family of functions indexed by a parameter called the key with the following property: for all distinct inputs, the probability over all keys that they collide is small.

Definition 1 *Let h_k be a function of an (ℓ, q)-family H from an ℓ-bit set A to an q-bit set B with the parameter k taken in a set of $\mathcal{K}_{\mathsf{UHF}}$. Let ϵ be any positive real number. Then, h_k is an* **ϵ-almost universal** *class (or ϵ-AU class) of hash function if* $\forall x, x' \neq x \in A : \Pr_k\{h_k(x) = h_k(x')\} \leq \epsilon$.

Definition 2 *h_k is $\oplus$-linear if* $\forall x, x' \neq x \in A : h_k(x \oplus x') = h_k(x) \oplus h_k(x')$.

Definition 3 *h_k is an* **ϵ-almost XOR universal** *class (or ϵ-AXU class) of hash function if* $\forall x, x' \neq x \in A$ *and* $\forall \Delta \in B : \Pr_k\{h_k(x) = h_k(x') \oplus \Delta\} \leq \epsilon$.

3.6 MAC Based on UHFs

UHF is not a cryptographically secure primitive. That is, it is not generally collision-resistant against an adversary who can choose messages after selection of k. Thus UHF is not in general a MAC. The UHFs can be used for message authentication if the output is processed with another function.

A MAC algorithm based on UHFs consists of two building blocks: an efficient keyed compression function that reduces long inputs to a fixed length and a method to process the short hash result and an output transformation. In practical constructions, the encryption with the one-time pad is typically replaced by applying a pseudo-random function with secret key $k' \in \mathcal{K}_{\mathsf{PRF}}$. In this case, one obtains computational rather than unconditional security. Informally, a pseudo-random function family is a function that a computationally limited adversary cannot distinguish with probability substantially better than 1/2 from a function chosen uniformly at random from all functions with the same range and domain.

Let $f_{k'}$ denote a pseudo-random function family indexed by the key k', which is computationally indistinguishable from a random family of functions from D to R. We define the prf-advantage of an adversary $\mathcal{A}$ for family f as $\mathsf{Adv}_f^{\mathsf{prf}}(\mathcal{A}) = \left|\Pr[k' \leftarrow \mathcal{K}_{\mathsf{PRF}} : \mathcal{A}^{f_{k'}(\cdot)} = 1] - \Pr[\zeta \leftarrow \mathcal{F}^{D\to R} : \mathcal{A}^{\zeta(\cdot)} = 1]\right|$, where $\mathcal{F}^{D\to R}$ is the set of all functions from D to R. We denote by $\mathsf{Adv}_f^{\mathsf{prf}}(q_1, t_1)$ the maximum prf-advantage of an adversary making q_1 queries to its oracle and running in time t_1.

We assume that the sender keeps the state with the counter (nonce) $c \in \mathcal{C}$. Note that we need to guarantee that c is not reused during the MAC generation. The design of MAC obtained from an ϵ-AXU and $\oplus$-linear hash function h_k is given by the following equation [9]:

$$MAC_{k||k'}(x) = h_k(x) \oplus f_{k'}(c). \tag{1}$$

Given a UHF family $h : \mathcal{K}_{\mathsf{UHF}} \times A \to B$ and a PRF family $f : \mathcal{K}_{\mathsf{PRF}} \times \mathcal{C} \to B$, we construct the MAC $\mathsf{UMAC} = (\mathsf{UGen}, \mathsf{UTag}, \mathsf{UVer})$ such as : $\mathsf{UGen}(1^q)$ generates the key (k, k') uniformly at random from $\mathcal{K}_{\mathsf{UHF}} \times \mathcal{K}_{\mathsf{PRF}}$; $\mathsf{UTag} : \mathcal{K}_{\mathsf{UHF}} \times \mathcal{K}_{\mathsf{PRF}} \times A \to \mathcal{C} \times B$ is defined as $\mathsf{UTag}_{k,k'}(M) = (c, h_k(M) \oplus f_{k'}(c))$; $\mathsf{UVer} : \mathcal{K}_{\mathsf{UHF}} \times \mathcal{K}_{\mathsf{PRF}} \times A \times C \times B$ is defined as $\mathsf{UVer}_{k,k'}(M, (c, tag)) = 1$ if and only if $h_k(M) + f_{k'}(c) = tag$.

We denote by $\mathsf{Adv}_{\mathsf{UMAC}}^{\mathsf{uf\text{-}mac}}(q_1, q_2, t_1)$ the maximum advantage of all adversaries against existentially unforgeability under an adaptive chosen message attack, making q_1 queries to UTag, q_2 queries to UVer and running in time at most t_1. The tagging algorithm of UMAC outputs, in addition to the composition of UHF and PRF, a unique counter c incremented at each invocation. Thus, the UMAC is stateful and its properties are as follows [15, 4].

Fact 1 *Assume that h is an ϵ^{UHF}-AXU family of hash functions and f is a PRF family. Then* UMAC *is a stateful MAC with advantage:* $\mathsf{Adv}_{\mathsf{UMAC}}^{\mathsf{uf\text{-}mac}}(q_1, q_2, t_1) \leq \mathsf{Adv}_f^{\mathsf{prf}}(q_1 + q_2, t_1) + \epsilon^{\mathsf{UHF}} q_2$.

4 The Yu's Scheme

Yu et al. [16] have proposed the novel mechanism, in which a forwarder can filter polluted messages before spreading the pollution in the XOR network coding systems. This scheme exploits probabilistic key pre-distribution and MACs. We describe the brief procedure of each phase of Yu's scheme.

Parameter setup phase: The source node chooses t, u and $\{r_1, \ldots, r_t\}$. Any node can compute a hash chain from a given seed r_j using a pseudo-random permutation function g. The source node has t random keys $k_{s,1}$, ..., $k_{s,t}$ from a global key pool $\mathcal{K}$, where s is the index of the source node. The index of each key $k_{s,j}$ in the key pool for $j = 1, \ldots, t$ is denoted as $id(k_{s,j})$. A forwarder picks t random keys from $\mathcal{K}$. Note that sinks have the same t keys as the source node for complete verification of messages.

MAC calculation phase: The source node attaches t MACs to each message M_i for $i = 1, \ldots, n$. More concretely, M_i is attached with $MAC_{i,1}, \ldots, MAC_{i,t}$ as well as the corresponding indeces of the random keys that are used to generate MACs. Thus, the source node actually generates and transmits:

$$M_i, (id(k_{s,1}), MAC_{i,1}), \ldots, (id(k_{s,t}), MAC_{i,t}). \tag{2}$$

For $j = 1, \ldots, t$, MAC is defined as $MAC_{i,j} = \mathsf{Enc}_{k_{s,j}}(id(k_{s,j}), r_j, \sigma_{i,j})$, where Enc denotes symmetric-key encryption function using key $k_{s,j}$ and $\sigma_{i,j}$ is the hash of u randomly selected codewords of M_i. Note that the MAC is decryptable in this scheme. The positions of codewords in M_i are randomly selected by the outputs of hash chain with a random seed r_j. The hash is computed by $\sigma_{i,j} = \bigoplus_{\ell=1}^{u} m_{i,r_{j,\ell}}$, where $r_{j,1}, \ldots, r_{j,u}$ are the indeces of selected codewords and also the output values of hash chain. Each source message is attached with t MACs, and each MAC is computed from u codewords. In other words, each MAC authenticates u codewords of M_i.

Message verification phase: Each forwarder or sink verifies its input messages based on the MACs for which it has the shared key(s) with the source node. When receiving a message along with the MACs of source messages, it first checks the indeces prefixed to each MAC to find a shared key. Then, it decrypts the corresponding MACs of source messages and generates the indeces of u codewords from r_j. After identifying the indeces of codewords, it takes the corresponding codewords out of the received message and calculates the hash of these codewords. It further takes out the hashes embedded into the decrypted MACs of source messages and encodes them using the encoding vector transmitted along with the received message. Finally, it checks if the hash of the received message equals the combination of the hashes embedded in the corresponding MACs. If equals, the verification succeeds. Otherwise, the received message is assumed to be polluted and will be discarded.

When each forwarder generates its output message, it always attaches the MACs of all source messages from which this output message is produced. For example, when a forwarder generates $E = M_1 \oplus M_2$, it will attach $MAC_{1,1}$, ..., $MAC_{1,t}$ and $MAC_{2,1}$, ..., $MAC_{2,t}$ to its output message E.

4.1 Problem Statement

In Yu's scheme, all the forwarders just forward their MACs to downstream nodes without XOR network coding of MACs. Actually, the MAC is decrypted to verify the corresponding codewords of a message. Due to such a special (decryptable)

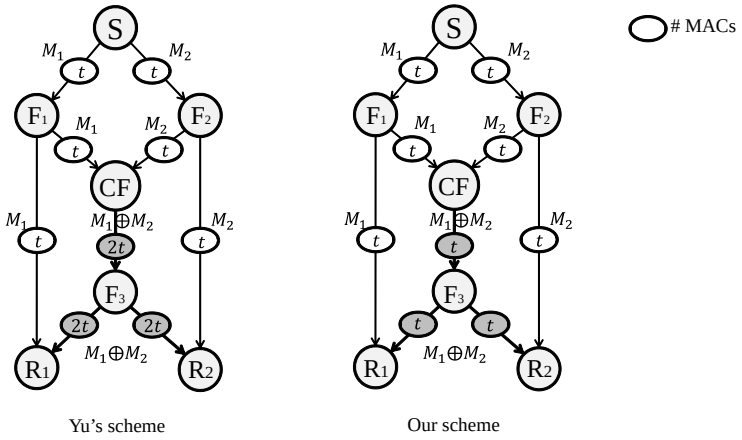

Fig. 1. The problem of Yu's scheme.

mechanism, the contents embedded into the decrypted MACs are operated with XOR network coding and hence MACs are verified, although MACs are not directly operated. Therefore, the number of MACs increases at a coding point, that is, the MACs cannot be suppressed to a certain number. In a worst case scenario, a forwarder transmits nt MACs to downstream nodes.

Figure 1 shows the difference between Yu's scheme and our scheme to explain the problem of Yu's scheme in an example. The source node S wants to send two messages M_1 and M_2 to two sinks R_1 and R_2. Let CF and F denote a coding forwarder and a mere forwarder, respectively. Where a forwarder performs coding is dependent on a network topology (CF receives two or more messages). At first, S sends M_1 and M_2 to F_1 and F_2 with their MACs, respectively. Then, a forwarder broadcasts the message and its MACs to downstream nodes. The source node attaches t MACs to each message M_1 and M_2. While CF has to forward $2t$ MACs to downstream nodes in Yu's scheme, CF has only to forward t MACs in our scheme. Hence, the communication amount of MACs from CF to sinks in Yu's scheme is twice our scheme in this example.

5 Our Scheme

In this section, we propose the first symmetric-key-based scheme not only to filter polluted messages but also to operate MACs with the XOR network coding on a forwarder. The primary aim of our scheme is to reduce the amount of extra space for integrity protection, i.e., the number of MACs on communication traffic. The XOR network coding of MACs produces improvement which does not increase the number of MACs at a coding point. We describe the detailed procedure of each phase of our scheme in the rest of this section.

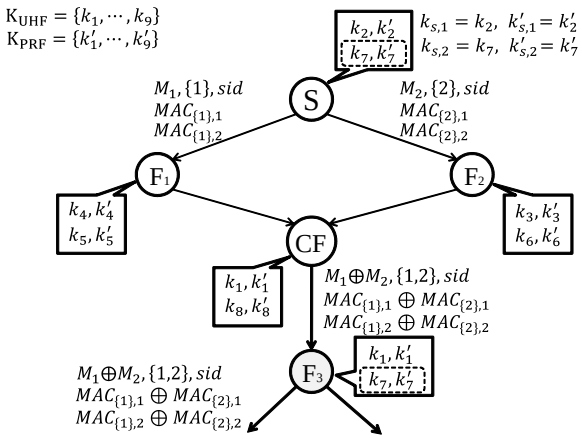

Fig. 2. Example of XOR network coding in our scheme ($t = 2$, $n = 2$).

Parameter setup phase: System parameters m and q are given in advance, since they are related to pre-determined parameters of some functions. The source node chooses t, u mid and sid. Any node can compute a hash chain from the seed r_j using a pseudo-random permutation function g. Any node can also compute an universal hash function h and a pseudo-random function[1] f. The source node has random keys $k_{s,1}, \ldots, k_{s,t}$ from a global key pool $\mathcal{K}_{\mathsf{UHF}}$ and $k'_{s,1}, \ldots, k'_{s,t}$ from another global key pool $\mathcal{K}_{\mathsf{PRF}}$. The index of each key $k_{s,j}$ is $id(k_{s,j})$. A forwarder picks t random keys from each of $\mathcal{K}_{\mathsf{UHF}}$ and $\mathcal{K}_{\mathsf{PRF}}$, i.e., $2t$ random keys in total. Note that sinks have the same $2t$ keys as the source node for complete verification of messages.

MAC calculation phase: The source node attaches t MACs to each message M_i for $i = 1, \ldots, n$. The source node generates and transmits:

$$M_i, \{i\}, sid, (id(k_{s,1}), MAC_{\{i\},1}), \ldots, (id(k_{s,t}), MAC_{\{i\},t}), \tag{3}$$

where $id(k_{s,j}) = id(k'_{s,j})$. For $j = 1, \ldots, t$, MAC is defined as follows:

$$MAC_{\{i\},j} = h_{k_{s,j}}(\sigma_{i,j}) \oplus f_{k'_{s,j}}(sid||i), \tag{4}$$

where $||$ denotes concatenation and the $\sigma_{i,j}$ is the hash of u randomly selected codewords of M_i, same as Yu's scheme. The random seed r_j, which is used to generate hash chain for $\sigma_{i,j}$, is computed as $r_j = H(f_{k'_{s,j}}(sid||i))$ in our scheme.

MAC coding phase: Two or more MACs can be operated with XOR network coding in this UHFs-based MAC. The forwarder generates and transmits:

$$E_\tau, mid, sid, (id(k_{s,1}), MAC_{mid,1}), \ldots, (id(k_{s,t}), MAC_{mid,t}), \tag{5}$$

[1] In practice, AES acts as a pseudo-random function (PRF). Other even more practical constructions of PRFs deployed in standards use MAC functions, such as HMAC.

where E_τ is encoded message of τ source messages $M_1, \ldots, M_\tau$ and mid is a set of message indeces which constitutes E_τ, that is, $mid = \{1, \ldots, \tau\}$ ($\tau \geq 2$). For $j = 1, \ldots, t$, the coded MAC is defined as follows:

$$MAC_{mid,j} = h_{k_{s,j}}(\sigma_{mid,j}) \oplus F_{k'_{s,j}}(sid||mid), \tag{6}$$

where we define $F_{k'_{s,j}}(sid||mid) = f_{k'_{s,j}}(sid||1) \oplus \cdots \oplus f_{k'_{s,j}}(sid||\tau)$ and $\sigma_{mid,j} = \sigma_{1,j} \oplus \cdots \oplus \sigma_{\tau,j}$. We assume that $F_{k'_{s,j}}()$ is a pseudo random function. The $\sigma_{i,j}$ is the hash of u randomly selected codewords of E_τ. The r_j is computed as $H(F_{k'_{s,j}}(sid||mid))$.

Figure 2 shows an example of XOR network coding in our scheme. The source node S sends M_1 and M_2 ($n = 2$) to F_1 and F_2 with their MACs, respectively. Each source message attaches two MACs ($t = 2$). Hence, the number of their keys in each forwarder is four ($= 2t$) in total. A forwarder broadcasts the message and its MACs to downstream nodes. Since this MAC has homomorphic property, two MACs are operated with XOR network coding by the node CF as follows:

$$\begin{aligned} MAC_{\{1\},j} \oplus MAC_{\{2\},j} &= h_{k_{s,j}}(\sigma_{1,j}) \oplus f_{k'_{s,j}}(sid||1) \oplus h_{k_{s,j}}(\sigma_{2,j}) \oplus f_{k'_{s,j}}(sid||2) \\ &= h_{k_{s,j}}(\sigma_{\{1,2\},j}) \oplus F_{k'_{s,j}}(sid||\{1,2\}), \quad (j = 1, 2). \end{aligned} \tag{7}$$

Note that $id(k_{s,j})$ is omitted in this figure.

Message verification phase: We consider the verification of MACs by a forwarder. This verification phase has three status; *impossible*, *valid* and *failed*. In the case of *impossible* and *valid*, the forwarder transmits data to the downstream nodes. Otherwise, it discards them. The coding forwarder conducts the XOR network coding of the message and their MACs before forwarding them.

1. A forwarder first checks $id(k_{s,j})$ prefixed to each MAC to see if it has any shared key with the source node. If it does not find any shared key (i.e., *impossible*), it forwards the messages and their MACs.
2. Once finding a shared key, it computes the seed r_j of the corresponding MACs and generates the indeces of u codewords from r_j using hash chain.
3. After identifying the indeces of codewords, it takes the corresponding codewords out of the received message and calculates the hash $\sigma_{i,j}$ of these codewords.
4. It computes $MAC_{\{i\},j}$ using the $\sigma_{i,j}$ in Equation (4) or (6). The values sid and mid are public information.
5. Finally, it checks if the MACs of the received messages equals the computed MACs. If equals (i.e., *valid*), the verification succeeds. Otherwise (i.e., *failed*), the received message is assumed to be polluted and will be discarded.

In Figure 2, we show an example of verification by the forwarder F_3 which shares the keys $k_{s,2}$ and $k'_{s,2}$ with S. This means that F_3 can verify the coded MAC: $MAC_{\{1\},2} \oplus MAC_{\{2\},2}$ by using $k_{s,2}$ and $k'_{s,2}$. More concretely, after F_3 computes $\sigma_{\{1,2\},2} = \sigma_{1,2} \oplus \sigma_{2,2}$ directly from $M_1 \oplus M_2$, it then computes $MAC_{\{1,2\},2} = MAC_{\{1\},2} \oplus MAC_{\{2\},2}$ as Equation (7). Note that F_3 knows neither M_1 nor M_2. It finally checks if this value equals the received MAC. But other forwarders F_1, F_2 and CF cannot verify MACs since they do not share any key with S.

6 Discussion

6.1 Security

In this section, we discuss both the security of MAC used in our scheme (in the case that a MAC key is not revealed) and the security against pollution attacks (in the case that some MAC-keys are revealed).

For the composition of UHF and PRF to be a MAC, it is important that the counters used as input into the PRF be unique. In our scheme, we use as input (counter) to the PRF, the session ID and the index i of source message when computing the MAC for the message M_i. As Fact 1 shows, assuming a pseudo random function, as long as no nonce is re-used during the generation of tags, forging a new valid $(M_i, sid||mid, tag)$ tuple is infeasible, even after the attacker has seen many such tuples before, either by eavesdropping or by active manipulation of tag generation. We can prove security of our MAC by the same framework as the security proof of the MAC based on universal hash. More specifically, we can prove security of our homomorphpic MAC, assuming h is an ϵ^{UHF}-AXU family of hash functions, f is a PRF family and F is also a PRF family.

Since a UHFs-based MAC has homomorphic property, different MAC value can be generated by operating two or more MACs with network coding. Such MAC value may be considered to be forgery. However, such operation is not forgery in network coding system since it is the coding operation itself in the redundant message processing. This is implicitly included in the security definition of [2].

It is important to filter polluted messages, as described in Section 3.1. On the other hand, it is also important to reduce the amount of extra space associated with communication complexity for integrity protection in WSNs. Our scheme probabilistically prevents the pollution attack and operates MACs with the XOR network coding for communication efficiency. This means that our scheme aims to satisfy both security and efficiency for XOR network coding in WSNs. Note that our scheme uses the same probabilistic technique to prevent pollution attacks as Yu's scheme and hence we can obtain the same results, in which the number of hops from polluted node until a pollution is detected is evaluated (i.e., a forwarder can filter polluted messages in a few hops with high probability.).

The tag pollution attack and its countermeasure are described in [12]. This scheme prevents the tag pollution attack under the assumptions of time synchronous and delayed authentication, since it is based on TESLA [14]. We do not consider that our scheme prevents perfectly the tag pollution attack. Both Yu's scheme and our scheme probabilistically prevents this attack without their assumptions.

6.2 Efficiency

Sensors are usually resource-limited and power-constrained. The energy savings of performing network coding are crucial for energy-constrained WSNs. Since the

nodes with the heaviest traffic are typically the nodes which are most essential to the connectivity of the network (e.g., area near sink), their failure may cause the network to partition. It is thus important to achieve constant congestion in large-scale WSNs. Actually, all of the MACs for all messages need to be gathered to each sink. The communication complexity should not depend on the number of source nodes n because of its power-constrained.

In this section, we compare our scheme with Yu's scheme in respect to the maximum communication complexity of MACs, storage amount of MAC keys and verification cost of one MAC for integrity protection of a forwarder at each session. Let UMAC, Dec, H and G be the computation costs of UHFs-based MAC, symmetric decryption over $|2^q|$, non-cryptographic hash function and pseudo-random permutation function, respectively. While the maximum number of MACs sent by a forwarder in Yu's scheme becomes nt, that in our scheme is constant t, described in Table 1. Hence, the maximum number of MACs in our scheme becomes $1/n$ compared with Yu's scheme, although the number of keys (i.e., storage amount) doubles. In verification cost of one MAC, Dec operation is required in Yu's scheme[2]. Note that H and G are very lightweight since the size of their outputs is quite small. The XOR operation is assumed to be negligible here because of very lightweight computation. Consequently, the maximum communication complexity of MACs of our scheme is superior to those of Yu's scheme, although the storage amount of MAC keys in our scheme is somewhat worse. The verification cost of our scheme is almost the same as that of Yu's scheme.

For example, we use well-chosen parameters in WSNs described in [16], e.g., $m = 16$, $n = u = 8$, $t = 5$, $q = 128$ (bits) and $|\mathcal{K}_{\mathsf{UHF}}| = |\mathcal{K}_{\mathsf{PRF}}| = 100$, in order to evaluate the congestion of MACs on a forwarder in each session. For such parameters, it is assumed that the size of M_i is 256 bytes. While the maximum communication complexity of MACs for M_i in Yu's scheme is 720 bytes ($(128 + 8 + 8) \cdot 5 \cdot 8/8$), that in our scheme is 80 bytes ($128 \cdot 5/8$). Hence, the maximum ratio of the size of MACs for each M_i is 280% in Yu's scheme and 32% in our scheme. Especially, in Yu's scheme, the size of attached MACs is much larger than that of M_i. The primary aim of our scheme is to reduce the amount of extra space associated with communication complexity for integrity protection. This result shows that the XOR network coding (aggregation) of MACs is pretty effective to reduce the number of MACs on communication traffic.

7 Conclusion

We have proposed the first symmetric-key-based scheme not only to filter polluted messages but also to operate MACs with the XOR network coding on a forwarder. Our scheme uses the UHFs-based MAC with a homomorphic property to hold homomorphic MAC, and hence it can aggregate MACs in our XOR network coding systems. The evaluation results show that our scheme is very

[2] If a block cipher is used as Dec then two Dec operations are required since the size of the input or output is beyond $|2^q|$.

Table 1. The maximum communication complexity of MACs, storage amount and verification cost for integrity protection of a forwarder at each session.

	Max comm. of MACs	Storage amount of keys	Verification cost per MAC
Yu's scheme	$n\|2^q\|t$	$\|2^q\|t$	$\mathsf{Dec} + (u-1)\mathsf{G}$
Ours	$\|2^q\|t$	$2\|2^q\|t$	$\mathsf{UMAC}+\mathsf{H}+(u-1)\mathsf{G}$

effective to reduce the amount of extra space associated with communication complexity for integrity protection. While the maximum ratio of the size of MACs for each M_i is 280% in Yu's scheme, that is 32% in our scheme.

References

1. Rudolf Ahlswede, Ning Cai, Shuo-Yen Robert Li and Raymond W. Yeung. "Network information flow". In IEEE Transactions on Information Theory 46(4), pages 1204–1216, 2000.
2. Shweta Agrawal and Dan Boneh. "Homomorphic MACs: MAC-Based Integrity for Network Coding". In ACNS 2009, pages 292–305.
3. Anya Apavatjrut, Wassim Znaidi, Antoine Fraboulet, Claire Goursaud, Cedric Lauradoux and Marine Minier. "Energy Friendly Integrity for Network Coding in Wireless Sensor Networks". In NSS 2010, pages 223–230.
4. Kevin D. Bowers, Ari Juels and Alina Oprea. "HAIL: a high-availability and integrity layer for cloud storage". In ACM Conference on Computer and Communications Security 2009, pages 187–198.
5. Larry Carter and Mark N. Wegman. "Universal Classes of Hash Functions". In J. Comput. Syst. Sci. 18(2), pages 143–154, 1979.
6. Qunfeng Dong, Jianming Wu, Wenjun Hu and Jon Crowcroft. "Practical network coding in wireless networks". In MOBICOM 2007, pages 306–309.
7. Laurent Eschenauer and Virgil D. Gligor. "A key-management scheme for distributed sensor networks". In ACM Conference on Computer and Communications Security 2002, pages 41–47.
8. Christos Gkantsidis and Pablo Rodriguez. "Cooperative Security for Network Coding File Distribution". In INFOCOM 2006.
9. Helena Handschuh and Bart Preneel. "Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms". In CRYPTO 2008, pages 144–161.
10. Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Medard and Jon Crowcroft. "XORs in the air: practical wireless network coding". In SIGCOMM 2006, pages 243–254.
11. Fang-Chun Kuo, Kun Tan, Xiangyang Li, Jiansong Zhang and Xiaoming Fu. "XOR Rescue: Exploiting Network Coding in Lossy Wireless Networks". In SECON 2009, pages 1–9.
12. Yaping Li, Hongyi Yao, Minghua Chen, Sidharth Jaggi and Alon Rosen. "RIPPLE Authentication for Network Coding". In INFOCOM 2010, pages 2258–2266.
13. Tebatso Nage, F. Richard Yu and Marc St-Hilaire. "Adaptive Control of Packet Overhead in XOR Network Coding". In ICC 2010, pages 1–5.

14. Adrian Perrig, Ran Canetti, J. D. Tygar and Dawn Xiaodong Song. "Efficient Authentication and Signing of Multicast Streams over Lossy Channels". In IEEE Symposium on Security and Privacy 2000, pages 56–73.
15. Victor Shoup. "On Fast and Provably Secure Message Authentication Based on Universal Hashing". In CRYPTO 1996, pages 313–328.
16. Zhen Yu, Yawen Wei, Bhuvaneswari Ramkumar and Yong Guan. "An Efficient Scheme for Securing XOR Network Coding against Pollution Attacks". In INFOCOM 2009, pages 406–414.
17. Shengli Zhang, Soung Chang Liew and Patrick P. Lam. "Hot topic: physical-layer network coding". In MOBICOM 2006, pages 358–365.
18. Zhang Zhang, Tiejun Lv, Xin Su and Hui Gao. "Dual XOR in the Air: A Network Coding Based Retransmission Scheme for Wireless Broadcasting". In ICC 2011, pages 1–6.