VANDER: Efficient Cooperative Watchdog Monitoring for Lossy Wireless Network Coding

Xin Lou, *Student Member, IEEE*, Hongyi Yao, *Member, IEEE*, Chee Wei Tan, *Senior Member, IEEE*, and Jianping Wang, *Member, IEEE*

***Abstract*—Network coding achieves multicast network capacity by allowing intermediate nodes to mix information in packets. However, due to the mixing operation, network coding is vulnerable to pollution attacks. Different from conventional cryptographic solutions, watchdog-based solutions, which have been recently proposed for network coding, rely on trusted nodes to monitor and verify behaviors of transmission nodes. However, the impact of lossy wireless communications has not been considered. Since false alarm and misdetection depend on the loss rate, these false results can happen with high probability when packet loss happens. In this paper, we propose VANDER, which is a novel cooperative watchdog scheme in heterogeneous wireless networks, where multiple watchdogs collaborate and efficiently detect pollution attacks in a lossy wireless environment. The novelty of our approach is that, when lossy overhearing happens, watchdogs work cooperatively to share the packet information, where no extra overhead is introduced to normal transmission nodes, and rather than retransmitting all lost packets among watchdogs, watchdogs use randomly generated Vandermonde hashes to detect corrupted packets. Moreover, VANDER is capable of detecting successive colluded adversaries. In addition to the low false alarm and misdetection probabilities, VANDER also achieves low computational complexity and communication overhead. Numerical experiments are provided to support the theoretical analysis of VANDER.**

***Index Terms*—Ad hoc networks, algorithms, network coding, network security.**

I. Introduction

NETWORK coding is a promising approach to achieving the maximum throughput of multicast networks [1]. Since network coding requires that intermediate network nodes mix received packet contents before forwarding, a single corrupted packet from the adversarial node can potentially pollute all the information reaching the destination. Previous work on pollution attacks in network coding transmissions can be categorized into three classes: information-theoretic, cryptographic, and watchdog-based approaches. The information-theoretic approaches in [2] and [3] use end-to-end error correction codes to decode source messages. Thus, it makes minimal changes to existing network coding schemes; only the source and the sink are involved in performing sophisticated computations to detect and correct errors introduced by malicious nodes. However, these schemes are geared toward a worst-case view of the adversarial action, in which adversaries locate themselves at the weakest part of the network (the bottlenecks). In cryptographic solutions [4]–[18], cryptographic primitives are used to enable honest network nodes to detect and drop corrupted packets. However, these approaches usually have high implementation complexity. In [19], they conducted detailed analysis and experimental evaluation in realistic wireless network coding settings of the representative cryptographic methods. The experimental evaluation showed that all the tested schemes induce a throughput degradation that negates the performance benefits of network coding in the presence of multiple colluding adversaries.

Recently, the broadcast nature of the wireless medium has been utilized to design schemes against pollution attacks, e.g., in [20] and [21]. To be precise, a third-party trusted node, i.e., a watchdog, overhears packet transmissions and detects pollution attacks. If the watchdog overhears *all* the incoming and outgoing data packets of the monitored node, any malicious behavior of that node can be efficiently detected by the watchdog. However, if a watchdog misses incoming packets of the monitored node, a legitimate packet would be falsely accused by this watchdog. Moreover, if a watchdog loses outgoing packets of the monitored node, this watchdog may misdetect the corrupted packet, which then pollutes the whole network transmissions. Previous work [22] combined the idea of maximum distance separable (MDS) codes to deal with watchdog lossy overhearing in multihop routing networks. We note that their work cannot be directly generalized to network coding transmissions in which each node performs mixing rather than routing. In this paper, we propose a new watchdog monitoring scheme against pollution attacks and analyze the benefits of this scheme under the additional constraint of lossy wireless transmissions.

For the wireless network coding transmissions, a straightforward solution for lossy overhearing is to *resend* all packets that the watchdog lost (we term it as WD-RESEND). In WD-RESEND, finite-field linear operations suffice for verification.

Manuscript received June 17, 2013; revised March 2, 2014; accepted April 8, 2014. Date of publication May 9, 2014; date of current version February 9, 2015. This work was supported in part by the Research Grants Council of Hong Kong under Project RGC AoE/E-02/08, CityU 122013 and CityU 120612, and by the National Natural Science Foundation of China under Grant 61272462. The review of this paper was coordinated by Dr. C. Yuen.

X. Lou, C. W. Tan, and J. Wang are with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: louxin86@gmail.com; cheewtan@cityu.edu.hk; jianwang@cityu.edu.hk).

H. Yao is with the Tower Research LLC, New York, NY 10013 USA (e-mail: yaohongyi03@gmail.com).

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

Digital Object Identifier 10.1109/TVT.2014.2322939

However, this approach introduces a large overhead. In addition, if upstream nodes of the monitored node are also attacked by the adversary, they can collude with the monitored node to retransmit fake packets to the watchdog to help the adversary avoid detection. A reasonable solution is to use multiple watchdogs to monitor the same node and to share information about lost packets among one another (we term it MWD-RESEND). However, MWD-RESEND causes a large overhead because of retransmissions of lost packets among watchdogs. In this paper, a method that has a low overhead and low computational complexity is proposed in heterogeneous wireless networks, where two kinds of wireless nodes exist, i.e., normal transmission nodes and helper nodes [23], [24]. In our scheme, we require that any intermediate normal transmission node in the network is monitored by multiple trustable watchdogs (at least two), i.e., helper nodes that have larger covering areas play as watchdogs. As most watchdogs can cover multiple normal transmission nodes, the number of watchdogs is much less than that of normal transmission nodes. Our contributions are as follows.

1) We analyze how lossy overhearing leads to the false alarm and the misdetection, as well as their consequences in wireless network coding with watchdog monitoring.
2) To achieve both computation and communication efficiencies in preventing pollution attacks of lossy wireless network coding transmissions, we propose VANDER, which is a novel watchdog cooperative scheme, that leverages the property of the Vandermonde[1] matrix to detect pollution attacks through cooperation. In this scheme, finite-field linear operations at the watchdog are sufficient to detect pollution attacks with the effective security guarantee.
3) We analyze the communication overhead and the computational complexity of VANDER. Assuming lossy overhearing, VANDER can achieve both low false alarm probability and low misdetection probability, which decreases *exponentially* with the collaborative communication overhead (cf. Theorem 1). Further, we characterize the tradeoffs between the overhead and the misdetection probability, as well as the false alarm probability.
4) We conduct extensive numerical simulations to evaluate the performance of VANDER and compare it with other schemes, where VANDER significantly reduces approximately 80% of the communication overhead, and the verification process is at least six to eight times faster than that of typical cryptographic schemes in average.

The remainder of this paper is organized as follows. Section II introduces the related work. Section III describes the system model and the problem formulation. The VANDER algorithm is proposed in Section IV. The overhead analysis and tradeoff discussion are presented in Section V. Section VI evaluates the performance of VANDER. Section VII concludes this paper.

[1]In linear algebra, a Vandermonde matrix is a matrix with the terms of a geometric progression in each row.

II. Related Work

The seminal work of Ahlswede *et al.* [1] first proved that multicast capacity (i.e., optimal throughput) can be achieved by introducing coding to the network nodes. Later, the work in [25] and that in [26] showed that linear network coding suffices to attain the capacity. Additionally, the work of Ho *et al.* [27] showed that, rather than using global network topology information to construct network codes, low-complexity random linear network coding (RLNC) could achieve the capacity in a distributed manner. As pollution attacks can make network coding quite inefficient or it can even damage transmissions in the whole network in the worst case, researchers have proposed many methods to defend it. Existing work in defending against pollution attacks in network coding falls into three categories: information-theoretic, cryptographic, and watchdog-based schemes.

Information-Theoretic Schemes: In the information-theoretic approach, defense is handled in an end-to-end manner. Therefore, one can leverage on error correction schemes while letting the intermediate nodes to implement standard network coding operations. In particular, Ho [2] proposed a scheme in which receivers can detect pollution attacks. Later, Jaggi [3] and Silva *et al.* [28] developed the first polynomial-time network coding schemes that tightly achieve the error correction rate bounds. Recently, in [29], efficient network coding construction has been proposed to attain the optimal error correction rates in the multiple-source scenarios.

Cryptographic Schemes: In [4]–[6], homomorphic hash functions were used to verify the integrity of the packet. However, both of [4] and [5] assumed secure channels to transmit hash values, and the homomorphic signature scheme in [6] involves high computational complexity. Yu *et al.* [7] combined hash functions and RSA signatures to detect pollution attacks. However, a recent work [8] has proved that this scheme did not satisfy the required homomorphic property. Zhao *et al.* [9] proposed a nonhomomorphic signature scheme that used subspace checking to verify the packet. However, their scheme requires the source to know the whole file before the transmission. Boneh *et al.* [10] generalized the scheme in [7] to support data streaming by involving public key signatures for each individual vector.

Compared with digital signature schemes, message authentication codes are used in [11] and [12] against pollution attacks. However, in [11], the corrupted packet may not be identified at the first-hop downstream node; thus, it may pollute other packets. In [12], it suffers tag pollution attacks. Li *et al.* [13] developed RIPPLE, and Zhang [14] proposed MacSig to effectively deal with pollution attacks and prevent tag pollution attacks. However, RIPPLE requires global synchronization among nodes, which is similar to DART [15]. Although MacSig is an innovative hybrid-key scheme, it still has a large overhead. In [16], the SpaceMac was proposed to detect the attacker in the intraflow network coding systems. However, in this scheme, a central entity, i.e., the controller, is needed to master the complete network topology and then performs the coordination. In [17], they proposed a new defense scheme that is based on the null space properties, and it does not rely on assumptions about the network topology or time synchronization. However,

the secure channel among wireless nodes is needed. In [18], a key predistribution-based tag encoding scheme was proposed, in which all intermediate nodes and sinks can detect the correctness of the received data packets.

Watchdog-Based Schemes: Recently, trusted third-party (i.e., watchdog) monitoring schemes have been proposed to solve pollution attacks by leveraging the broadcast nature of wireless communications. The first version [30] of the watchdog monitoring was introduced to detect malicious nodes in wireless routing networks. In [22], also for the routing scenario, Liang *et al.* used the properties of MDS codes to deal with lossy overhearing at the watchdog.

In [20], the watchdog was first used to detect pollution attacks in two-hop wireless network coding transmissions. In particular, it assumed that the wireless channel had certain transition probabilities, and this was known to all the nodes. Then, a graphical model was proposed to infer whether a node intentionally pollutes the packets or random error occurs in the channel. In [21], the scheme in [20] was extended to the multihop and multisource networks. In both [20] and [21] (with an extended version in [31]), the proposed schemes cannot deal with the false alarm problem and the misdetection problem in lossy wireless transmissions.

III. System Models and Problem Description

The following notation is used in this paper. Calligraphic letters represent sets. Boldface lowercase letters denote column vectors, and italic uppercase letters denote the wireless node. Let $\tilde{\mathbf{a}}_j$ denote the jth column vector of the Vandermonde matrix [32] and the lowercase letter $\tilde{a}_j$ denote the element to construct vector $\tilde{\mathbf{a}}_j$, which is randomly chosen from the finite field $\mathbb{F}_q$. We also let $\tilde{\mathbf{a}}_j^T \mathbf{y}_i$ be the random hash (see Section IV-A), where the superscript $(\cdot)^T$ denotes the transpose.

A. Network Coding Transmission

We consider a heterogeneous wireless network $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, in which $\mathcal{U}$ is the set of nodes, and $\mathcal{E}$ is the set of links. In node set $\mathcal{U}$, there are two kinds of wireless nodes: normal transmission nodes and helper nodes (also called the helping node in [23] and [24]). Note that the helper node only serves as the intermediate nodes in the pure network coding scheme (when not considering the watchdog scheme). We consider the multicast scenario where a source node $S \in \mathcal{U}$ wants to transmit information to receivers $N \subset \mathcal{U}$ using RLNC [27], whose basic operations are described as follows.

Source Encoder: For each stream of messages, the source S arranges them into m*source packets* $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ over $\mathbb{F}_q^n$, where $\mathbb{F}_q^n$ is a finite space with cardinality q and dimension n. Each source packet, e.g., $\mathbf{v}_i$, is prefixed with the ith unit vector in $\mathbb{F}_q^m$ to track the linear transform induced by RLNC. The source takes independently and uniformly random linear combinations over $\mathbb{F}_q$ of the packets in $\mathcal{V}$ to generate, respectively, the *coded packets*, which are then sent into the network.

Internal Node Encoders: Each internal node randomly and linearly combines received coded packets to generate the outgoing coded packets. Note that, due to linearity in RLNC, each coded packet, e.g., $\mathbf{y}$, that is received (or generated) by each internal node can be represented as

$$\mathbf{y} = \sum_{i=1}^{\ell} \alpha_i \mathbf{y}_i \tag{1}$$

where $\mathbf{y}_i$ represents the incoming packet, and $\ell (\ell \leq m)$ is the number of received incoming packets.

Receiver Decoder: At the receiver, after receiving m linearly independent coded packets, source packets can be correctly decoded. As proved in [27], when the network capacity is no less than m, the receiver can receive m linearly independent coded packets with a high probability. Moreover, by the result in [27], when choosing a field with larger size, the probability that all the receivers can correctly decode the information is higher.

B. Attack Model

We consider pollution attacks from the omniscient adversarial node, in which the adversarial node is assumed to have the network topology information, its neighbors' transmissions, source messages, and the unbounded computation ability. The adversary aims at injecting erroneous packets into the network to cause pollution. The definition is given as follows.

Definition 1 (Pollution Attack): Let $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_\ell$ be the packets received at node $U \in \mathcal{U}$, where ℓ is the number of incoming packets used to encode the new packet. Let $\mathbf{y}$ be the encoded packet at U. Then, $\mathbf{y}$ is said to be a polluted packet if and only if $\mathbf{y}$ is not in the linear space spanned by $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_\ell$.

Due to the combining feature of RLNC, a single polluted packet suffices to pollute the whole network transmissions. In the following, we introduce the watchdog model.

C. Watchdog Model

In the heterogeneous wireless network $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, as the helper node has much larger transmission range than the normal transmission node [23], [24], we make use of it to play as the watchdog node. Note that, as the cost (both the cost of the equipment and their maintenance cost) of helper nodes is much higher than that of the normal transmission nodes, it is not possible to replace all normal transmission nodes by these powerful helper nodes. For simplicity, in the following, we use watchdog to represent the helper node. Similar to [24], we divide the bandwidth of watchdogs into two parts: bandwidth for communication with normal transmission nodes and bandwidth for communication among watchdogs. Moreover, since the number of watchdogs is much less than that of normal transmission nodes, it is possible to ensure that these watchdogs are protected from any physical attack, such as in [33], which then implies that watchdogs are trustable. Assume that we have β watchdogs $\mathcal{W} = \{W_k, k = 1, 2, \ldots, \beta\}$, where $W_k \notin \mathcal{U}$, to detect pollution attacks among normal transmission nodes. Each watchdog can overhear the transmission among the monitored nodes and their neighbors. We say node $U_i \in \mathcal{U}$ is monitored by a

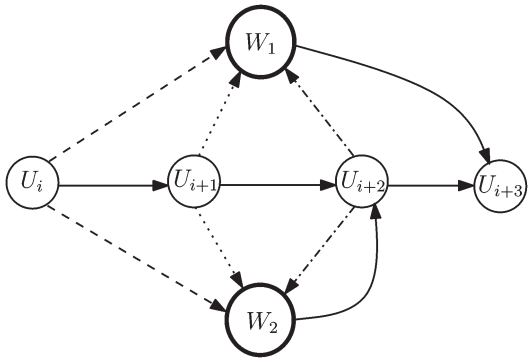


Fig. 1. Multiple-watchdog model. The thick lines denote the transmission between wireless nodes. The dashed lines denote that the watchdog overhears transmissions.

watchdog when the covering range of a watchdog can cover U_i and upstream nodes of U_i.[2] We also define the *monitoring set* of U_i as $\bar{\mathcal{U}}_i = \{W_{ik} : k = 1, 2, \ldots, \lambda\}$, such that U_i is monitored by any $W_{ik} \in \bar{\mathcal{U}}_i$. Moreover, a watchdog can communicate with other watchdogs in its transmission range through the dedicated channel. Thus, the communication among them does not affect normal transmission of data packets. We assume that efficient channel error correction schemes are used to correct the random errors. Since these schemes are also necessary in the common wireless transmissions, no additional overhead is introduced by using them.

We illustrate an example of the watchdog model as follows. Fig. 1 shows four normal transmission nodes $\{U_i, U_{i+1}, U_{i+2}, U_{i+3}\}$ and two watchdogs $\{W_1, W_2\}$ in a multihop wireless network. In this network, these four transmission nodes $\{U_i, U_{i+1}, U_{i+2}, U_{i+3}\}$ are all monitored by watchdogs. For simplicity, we just plot $\{W_1, W_2\}$ that can monitor U_{i+1} and U_{i+2}. Thus, the monitoring set of U_{i+1} and U_{i+2} can be expressed as $\bar{\mathcal{U}}_{i+1} = \bar{\mathcal{U}}_{i+2} = \{W_1, W_2\}$. When node U_{i+1} sends the encoded packet, each watchdog in $\bar{\mathcal{U}}_{i+1}$ verifies whether this encoded packet lies in the linear space spanned by the packets sent from U_i. After that, either W_1 or W_2 can send the decision packet to U_{i+2}. Then, U_{i+2} drops or accepts this encoded packet according to the decision from the watchdog. Similarly, watchdogs in $\bar{\mathcal{U}}_{i+2}$ monitor U_{i+2} by overhearing the transmissions from U_{i+1} to U_{i+2}, as well as transmissions from U_{i+2} to U_{i+3}.

D. Lossy Overhearing at the Watchdog

Due to the wireless transmission collision, packet loss happens over time. In the following, we show that conventional naive subspace checking at the watchdog fails to detect adversarial nodes in the lossy wireless environment.

- *False alarm (i.e., false positive)* means that a legitimate packet is accused as the polluted packet (or a packet cannot be verified) by the watchdog. In particular, consider the example in Fig. 1. Assume that the internal node U_{i+1} is honest, but the watchdog loses some incoming packets of U_{i+1}. Due to the randomness of RLNC, with a high probability, the outgoing packet of U_{i+1} fails the verification at W_1 and W_2. Therefore, legitimate packets will be dropped by U_{i+2}. Moreover, the event of false alarm is more serious than the ordinary packet loss at the internal node. To be precise, let p be the probability of packet loss, and let the internal node U_{i+1} use ℓ received packets to generate the outgoing packet. With probability $1-(1-p)^{\ell}$, a watchdog loses at least one of the received packets of U_i. Thus, the false alarm probability is $(1-(1-p)^{\ell})^2$ in this example.
- *Misdetection (i.e., false negative)* means that a corrupted packet is recognized as the legitimate packet by the watchdog. Now, also consider the example in Fig. 1. Assume a corrupted packet is sent by the adversarial node U_{i+1} and that this packet is lost by both W_1 and W_2. Then, this corrupted packet would be accepted as a valid packet by subsequent nodes, e.g., U_{i+2} and U_{i+3}. Since RLNC mixes packets, a single corrupted packet can end up corrupting all the information reaching a destination.

In the following, we propose a novel watchdog cooperative monitoring scheme VANDER to address the false alarm and misdetection problems in preventing pollution attacks through efficient cooperation.

IV. VANDER Watchdog Cooperative Monitoring

Without loss of generality, we focus on the VANDER construction of γ watchdogs in each monitoring set $\bar{\mathcal{U}}_i = \{W_{ik}, k = 1, 2, \ldots, \gamma\}$ of U_i, where $U_i \in \mathcal{U}$ is monitored by these γ watchdogs. We assume that these watchdogs have independent packet loss probabilities. For simplicity, we use a common notation p to represent the packet loss probability at each watchdog and use ℓ to represent the encoding packets from ℓ upstream nodes in the following. In general, the value of p and ℓ is different at different nodes. Moreover, we also assume that every watchdog has a unique ID so that watchdogs can identify one another.

To illustrate, we use the network in Fig. 2 to describe VANDER. In Fig. 2, the transmission node U, which receives packets from upstream nodes $\{U_1, U_2, \ldots, U_\ell\}$ and transmits packets to node U', corresponds to the monitoring set $\bar{\mathcal{U}} = \{W_k, k = 1, 2, \ldots, \gamma\}$. In the coding process, U combines ℓ packets to create the outgoing packet, where these ℓ packets either come from nodes in the whole upstream neighboring set $\{U_1, U_2, \ldots, U_\ell\}$ or its subset. Watchdogs in other monitoring sets work similarly as watchdogs in $\bar{\mathcal{U}}$. Packets sent from upstream nodes of U are monitored by the same scheme. We first sketch the key idea of VANDER and then present the algorithm in detail.

A. Key Ideas

We combine the cooperative watchdog scheme and the Vandermonde hash to mitigate the authentication cost and the communication overhead of preventing pollution attacks in a lossy wireless network, in which watchdogs may only partially

[2]Upstream nodes of U_i refer to the nodes that pass encoded packets from the source node side to U_i. Similarly, in the following, we refer downstream nodes as the nodes that receive packets from U_i and then pass the encoded packets to the sink side.

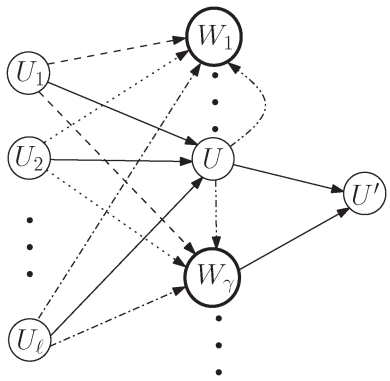


Fig. 2. VANDER model. The thick lines denote the transmission between wireless nodes. The dashed lines denote that the watchdog overhears transmissions.

overhear some packets from their upstream nodes. Instead of requesting the neighbors (i.e., upstream nodes) to retransmit lost packets, these multiple watchdogs can work cooperatively to detect the adversarial behavior of monitored nodes with Vandermonde hashes. Thus, no extra overhead is introduced in the normal transmission. VANDER is supported by two key ideas as follows.

Use of Cooperative Watchdogs: To avoid interrupting normal transmission nodes, VANDER finds out the adversarial node by the cooperation of watchdogs in the same monitoring set. To avoid the redundant information in the control packet, watchdogs in VANDER advertise brief "summary" information of individually overheard packet contents. To be precise, we use the idea similar to the buffer map [34] in peer-to-peer networks to decrease the advertising overhead among watchdogs. Instead of broadcasting the whole index of an overheard packet, a watchdog just broadcasts a small advertising packet, in which one b is used to represent whether this watchdog has received a corresponding incoming packet. Thus, the overhead is greatly decreased from ℓs b to ℓ b, where s is the index length, and ℓ is the number of incoming packets, The fact that watchdogs are able to get these ℓ packet indexes from the outgoing packet of the monitored node (see the packet format in the following) makes sure that the buffer map scheme can work well in VANDER. After the advertising phase, watchdogs just transmit the required packets to the "*leader*," which is responsible for verifying the outgoing packet. We will describe this specifically in the algorithm description and prove that our watchdog cooperation scheme provides security guarantee and only incurs a low overhead among watchdogs (see Section V).

Use of Vandermonde Hashes: It is observed that instead of verifying the whole packet, verifying the *random hash* (which is a projection on a randomly chosen *subspace* of $\mathbb{F}_q^n$) of a packet suffices to detect the corrupted packet. For instance, we assume that the subspace is rank *one* and is spanned by a randomly chosen vector $\tilde{\mathbf{a}} \in \mathbb{F}_q^n$. Let $\mathbf{y}$ be the packet that needs to be verified and $\mathbf{y}'$ be the corresponding legitimate packet that follows the coding rule. Instead of verifying $\mathbf{y} = \mathbf{y}'$, the watchdog only needs to check whether

$$\tilde{\mathbf{a}}^T \mathbf{y} = \tilde{\mathbf{a}}^T \mathbf{y}'. \tag{2}$$

TABLE I
SUMMARY OF PACKETS IN VANDER

Packet type	Function / Definition
Data packet	Information delivering
Incoming packet	Data packet from an upstream node
Outgoing packet	Data packet from U
VAP	Advertising packet information
VLP	Packet to indicate the Leader Watchdog
RRP	Responding to the corresponding VAP
Alarm packet	Informing corrupted/uncertain packet
Consent packet	Informing legitimate packet

If $\mathbf{y} = \mathbf{y}'$, (2) is always true; otherwise, (2) is false with a high probability.

Therefore, when packets suffer lossy overhearing at watchdogs, instead of sharing the whole overheard packets, watchdogs may share only random hashes of overheard packets with one another. However, when random subspace is *naively* chosen over $\mathbb{F}_q^n$, dn distinct symbols over $\mathbb{F}_q$ are needed to encode a subspace of rank d. Thus, to advertise individual subspace information to other watchdogs, a watchdog must broadcast these dn hash symbols, thus incurring a high communication overhead.

To reduce this communication overhead, we exploit a key property of the Vandermonde matrix. In particular, instead of the "completely" random subspace, a watchdog randomly chooses an $n \times d$ *Vandermonde matrix*, and then, it sets the rank d subspace as the column space of the Vandermonde matrix. Since each column of the Vandermonde matrix can be computed from its first component, the transmission of these d distinct *symbols* suffices to encode the subspace information. Thus, the broadcasting overhead is significantly reduced from dn symbols to d symbols.

B. Packets in VANDER

From the key idea description, we know that watchdogs cooperate with one another if lossy overhearing happens. Therefore, control packets are needed to collect relevant Vandermonde hashes. Here, we define packets used in VANDER before giving the algorithm details (see also Table I).

- *Data packet* is used for data transmission under RLNC. Each data packet is composed of three components as follows.
  1) *Packet index*. We assume that each data packet has the distinct packet index.
  2) *RLNC vector*. The RLNC vector is the data to be delivered, which is in $\mathbb{F}_q^n$ and defined as the coded packet in Section III.
  3) *Coding information*. The coding information includes the number of verified incoming packets that are used to generate the RLNC vector (i.e., ℓ), indexes of these incoming packets, and corresponding linear coefficients (global and local encoding coefficients) under RLNC.

- *Incoming packets* are data packets received at U from upstream neighbors. Incoming packets are verified by watchdogs in the upstream nodes' monitoring sets.
- *Outgoing packets* are data packets transmitted by U.
- *VANDER advertising packet* (VAP) is generated by the watchdog to inform the Vandermonde subspace information and the overheard packet information among watchdogs. For watchdog k, VAP contains d randomly and independently chosen elements $\{\tilde{a}_{k1}, \tilde{a}_{k2}, \ldots, \tilde{a}_{kd}\} \in \mathbb{F}_q$ for a positive integer d and the binary vector $\mathbf{x}_k = [x_{k1}, x_{k2}, \ldots, x_{k\ell}]^T$, in which ℓ is the number of encoded incoming packets. In the VAP, $x_{ki} = 1$ denotes that the kth watchdog has overheard the ith incoming packet. Otherwise, the kth watchdog has lost the ith incoming packet. Moreover, these x_{ki}'s should be arranged based on the order of the index, i.e., either ascending or descending by the indexes.
- *VANDER leader packet* (VLP) is the packet used to indicate the leader watchdog (LW). In particular, VLP contains the same information as one of the γ VAPs, i.e., each element of $\tilde{\mathbf{a}}$ in VLP satisfies $\tilde{a}_j = \tilde{a}_{kj}, \forall j = 1, 2, \ldots, d$ and $\mathbf{x} = \mathbf{x}_k$, where the kth VAP satisfies $\|\mathbf{x}_k\|_1 \geq \|\mathbf{x}_i\|_1, \forall i \in \{1, 2, \ldots, \gamma\}$, and $i \neq k$.
- *Request response packet* (RRP) is the response for the VLP. Each RRP contains d Vandermonde hashes $\{\tilde{\mathbf{a}}_1^T\mathbf{z}_i, \tilde{\mathbf{a}}_2^T\mathbf{z}_i, \ldots, \tilde{\mathbf{a}}_d^T\mathbf{z}_i\}$, where $\mathbf{z}_i$'s are the RLNC vector of *LW's* lost packets.
- *Alarm packet* is used to inform the index of the corrupted packet or the packet that cannot be verified by watchdogs (lack of enough information).
- *Consent packet* is used to inform the index of the legitimate packet.

As the alarm packet and the consent packet relay decisions from the watchdog, we term these packets as decision packets. Moreover, we assume that, if a watchdog lies in multiple monitoring sets, it can identify these control packets from different monitoring sets. Now, based on these packets, we describe the VANDER algorithm in the following.

C. VANDER Algorithm

Upon receiving a data packet from U, the downstream node U' would wait for the decision packet from one of the watchdogs. If U' does not receive the decision packet within a preset time threshold, U' would drop this unverified packet.

Fig. 3 shows the main block diagram of the VANDER algorithm for a watchdog, e.g., W_k. When W_k discovers an outgoing packet from U, it may launch VANDER verification. However, as other watchdogs may have verified the outgoing packet with their individually overheard packets, the VANDER verification would be stopped once the decision of the unverified outgoing packet is informed by other watchdogs.

VANDER Algorithm:

Let $\mathbf{y}$ be the unverified outgoing packet from U and $\bar{\mathcal{U}} = \{W_k, k = 1, 2, \ldots, \gamma\}$ be the monitoring set of U. Upon overhearing $\mathbf{y}$, the watchdog W_k in $\bar{\mathcal{U}}$ first extracts the coding information and the RLNC vector $\mathbf{z}$ from $\mathbf{y}$. Using this information, W_k checks whether it has overheard all the incoming packets that generate $\mathbf{y}$.

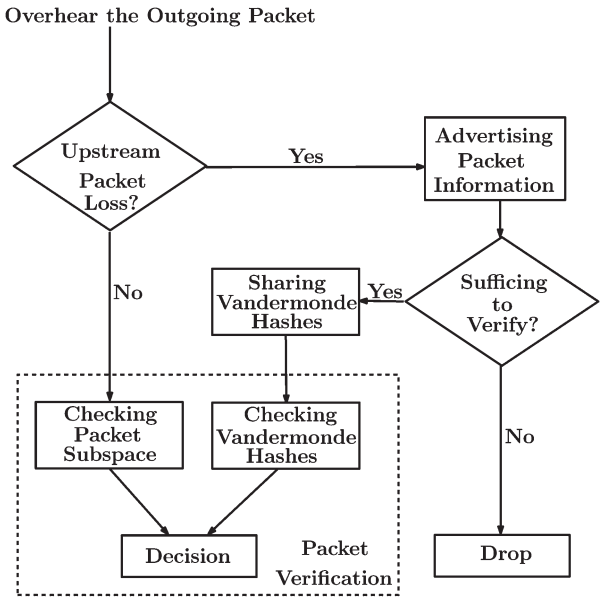


Fig. 3. Block diagram of VANDER for a watchdog. When a watchdog has overheard the outgoing packet from U, it proceeds with the VANDER verification to check whether the adversarial node exists.

1) If yes, let $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_\ell\}$ be the upstream RLNC vectors that generate $\mathbf{z}$, and $\{\alpha_1, \alpha_2, \ldots, \alpha_\ell\}$ be the corresponding linear coefficients (local encoding coefficient) in the coding information of $\mathbf{y}$. Then, W_k directly performs packet subspace checking as follows.

Packet Subspace Checking. Using overheard RLNC vector $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_\ell\}$ and coefficients $\{\alpha_1, \alpha_2, \ldots, \alpha_\ell\}$, W_k checks whether the following is true:

$$\mathbf{z} = \sum_{i=1}^{\ell} \alpha_i \mathbf{z}_i.$$

If it is true, W_k sends the consent packet to the downstream node U'; otherwise, W_k sends the alarm packet.

2) If no, W_k lacks some incoming packets that generate $\mathbf{y}$. W_k first independently and randomly generates d elements $\{\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_d\} \in \mathbb{F}_q$ with uniform probability. Then, it starts cooperating with other watchdogs in $\bar{\mathcal{U}}$ as follows.

- *Advertising among Watchdogs*. W_k first generates VAP that contains the information of $\{\tilde{a}_{k1}, \tilde{a}_{k2}, \ldots, \tilde{a}_{kd}\} \in \mathbb{F}_q$ for a positive integer d and $\mathbf{x}_k = [x_{k1}, x_{k2}, \ldots, x_{k\ell}]^T$. After generating the VAP, W_k advertises it to other watchdogs in $\bar{\mathcal{U}}$ by broadcasting.
- *Checking Subspace and Selecting the Leader Watchdog*. For any watchdog, e.g., W_k, if it receives $(\gamma - 1)$ VAPs (from the same monitoring set), we then denote it as $\tilde{W}_k$. $\tilde{W}_k$ is responsible for checking whether the incoming packets that are overheard by all other watchdogs in $\bar{\mathcal{U}}$ can verify $\mathbf{y}$.
  a) If no, $\tilde{W}_k$ sends the alarm packet to the downstream node U'. Then, U' drops this unverified packet. When

other watchdogs in $\bar{\mathcal{U}}$ receive this alarm packet, they terminate the monitoring scheme for $\mathbf{y}$.

b) If yes, $\tilde{W}_k$ checks which watchdog has overheard the most encoded incoming packets. Then, this watchdog is recognized as the LW, which is responsible for verifying $\mathbf{y}$. (If more than one watchdogs have received the most incoming packets, the one with the least ID is recognized as LW.) However, although LW has the most incoming packet information, it may not realize this when it losses VAPs from other watchdogs in the same monitoring set. Thus, $\tilde{W}_k$ that has received $(\gamma-1)$ VAPs advertise this information. $\tilde{W}_k$ first copies the Vandermonde subspace information and the binary buffer map vector from LW's VAP to create VLP. After that, VLP is broadcast to other watchdogs. When more than one watchdogs in $\bar{\mathcal{U}}$ receive all $(\gamma-1)$ VAPs, any of them may broadcast the same VLP. Note that if a watchdog has lost both $(\gamma-1)$ VAPs and the VLP, it will not involve in the Vandermonde verification process.

- *Sharing Vandermonde Hashes*: Watchdogs that have LW's lost incoming packets broadcast RRPs. An RRP is constructed as follows: For each RLNC vector $\mathbf{z}_i$ that is indicated as lost in the VLP, the RRP contains the Vandermonde hashes $\{\tilde{\mathbf{a}}_j^T\mathbf{z}_i : j=1,2,\ldots,d\}$ and the packet index respect to $\mathbf{z}_i$, where $\tilde{\mathbf{a}}_j=[\tilde{a}_j^1,\tilde{a}_j^2,\ldots,\tilde{a}_j^n]^T\in\mathbb{F}_q^n$ is calculated from the $\tilde{a}_j$ contained in the VLP (or VAP, if this watchdog has overheard all the VAPs).

After receiving enough RRPs, LW uses the Vandermonde hashes from other watchdogs and those computed from overheard incoming packets to perform the Vandermonde hashes checking as follows.

Vandermonde Hashes Checking. Using Vandermonde hashes $\{\tilde{\mathbf{a}}_j^T\mathbf{z}_i : j=1,2,\ldots,d; i=1,2,\ldots,\ell\}$ and $\{\tilde{\mathbf{a}}_j^T\mathbf{z} : j=1,2,\ldots,,d\}$, LW checks whether the following is true:

$$\tilde{\mathbf{a}}_j^T\mathbf{z}=\sum_{i=1}^{\ell}\alpha_i\tilde{\mathbf{a}}_j^T\mathbf{z}_i$$

for each $j=1,2,\ldots,d$. If it is true, LW sends the consent packet to the downstream node U'; otherwise, LW sends the alarm packet.

Note that $\mathbf{y}$ may have fake packet indexes; thus, LW would not receive enough Vandermonde hashes for the fake indexes. In that case, LW sends the alarm packet to inform the downstream node. Therefore, this polluted packet from the adversarial U can be also detected.

D. Colluded Adversaries

When two or more adversarial nodes are successively connected to one another (i.e., they form a path segment), these adversaries can work together to pass the watchdog's monitoring if no additional scheme is considered. For example, in Fig. 2, suppose U_1 and U are two successive adversarial nodes and U' is an honest transmission node. Then, U_1 and U can collude to pass the verification. Specifically, if U_1 transmits a corrupted packet $\mathbf{y}_1$ to U, then U can directly use $\mathbf{y}_1$ and other $\ell-1$ legitimate incoming packets to generate a corrupted packet $\mathbf{y}$ without waiting for the decision from watchdogs in U_1's monitoring set. In such a case, $\mathbf{y}$ from U would pass the verification of $W_1, W_2, \ldots, W_\gamma$ and pollute the successive transmissions.

Therefore, to avoid two successive adversarial nodes from colluding with each other to pass the packet verification, when W_k overhears an incoming packet, it first buffers this incoming packet in a temporary queue. Only when W_k also overhears the confirmation of this incoming packet can this packet be marked as the verified incoming packet and stored in W_k. Otherwise, this unverified incoming packet would be dropped after the preset time threshold is reached. Thus, we also assume that there must be at least one watchdog in a monitoring set lie in the transmission range of every watchdog in its neighboring monitoring set. Since the transmission range of watchdog is large and each watchdog can lie in different monitoring sets, this assumption is mild. Thus, the newly generated corrupted packet by the second adversary also cannot pass the watchdog verification. To explain this scheme clearer, we take the given example again. This time, since $W_1, W_2, \ldots, W_\gamma$ (or some of them) would receive an alarm packet from a watchdog from U_1's monitoring set, $\mathbf{y}_1$ would be dropped by them. Thus, if U includes $\mathbf{y}_1$ to generate $\mathbf{y}$, then $\mathbf{y}$ is treated as the uncertain packet (as $W_1, W_2, \ldots, W_\gamma$ have dropped $\mathbf{y}_1$ and they do not have enough information to verify $\mathbf{y}$), and $\mathbf{y}$ would be dropped by U'. Therefore, from the given description, we see that colluded adversaries can be efficiently identified by VANDER.

TABLE II
NOTATIONS USED FOR PERFORMANCE ANALYSIS OF VANDER

Notation	Definition
p	Packet loss probability
$\|q\|$	Symbol length of the finite filed used for RLNC
n	Length of the RLNC vector (over symbols in $\mathbb{F}_q$)
ℓ	Number of incoming packets that generate the outgoing packet
P_m	The desired misdetection probability
P_f	The desired false alarm probability
d	Number of Vandermonde hashes (rank of the subspace)
γ	Number of watchdogs in the monitoring set
$T_{\max}$	The upper bound on the communication overhead

E. Security of VANDER

We present the main theorem of VANDER as follows.

Theorem 1: VANDER has the following security guarantee.

- *False alarm*: The probability of false alarm is at most $(1-(1-p)^l)^\gamma(1-(1-p)^{2\gamma}(1-p^2)^\ell)$.
- *Misdetection*: The probability of misdetection is at most $(n/q)^d$.

Proof: See the Appendix. □

V. PERFORMANCE ANALYSIS OF VANDER

Here, we analyze the communication overhead and the computational complexity of VANDER. In Table II, we summarize the notations used for the following performance analysis.

A. Communication Overhead of VANDER

We first analyze the communication overhead that is introduced by VANDER.[3] Compared with the data packet under the standard RLNC setting, each outgoing packet of U contains additional coding information, which corresponds to the overhead $\ell|q|$ b.

With probability p, each watchdog suffers lossy overhearing. The expectation of the number of incoming packets the watchdog missed is ℓp. Thus, the corresponding communication overhead introduced among watchdogs consists of the following three components:

- Each VAP contains the symbols $\{\tilde{a}_{k1}, \tilde{a}_{k2}, \ldots, \tilde{a}_{kd}\} \in \mathbb{F}_q$ and ℓ b to represent the ℓ incoming packets, where $k = 1, 2, \ldots, \gamma$. The total overhead introduced by VAPs is $\gamma d|q| + \gamma\ell$ b.
- Each VLP contains $\{\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_d\} \in \mathbb{F}_q$ and ℓ b from LW. The total VLP has $d|q| + \ell$ b.
- RRPs also introduce the overhead. As there are at most ℓ Vandermonde hashes transmitted to LW and each hash has d symbols over $\mathbb{F}_q$, the communication overhead introduced by RRPs is $\ell d|q|$ b.

Since the communication overhead among watchdogs is introduced only when packet loss happens in the monitoring set, the overall overhead is upper bounded by

$$\begin{aligned} T_{\max} &= \ell|q| + \gamma d|q|p + \gamma\ell p + d|q|p + \ell p + \ell d|q|p \\ &= |q|(\ell + (\gamma+1)dp + d\ell p) + (\gamma+1)\ell p \\ &= \Theta(d\ell + d\gamma + \gamma\ell) \text{ b}. \end{aligned}$$

B. Computational Complexity of VANDER

In the following, for each entity in VANDER, we analyze the computational complexity by counting the number of finite-field multiplications being performed.

- For every watchdog, with probability p, the packet is lost. Then, it needs to compute $\tilde{\mathbf{a}}_1$, $\tilde{\mathbf{a}}_2, \ldots, \tilde{\mathbf{a}}_d$. Moreover, at most $p\ell d$ Vandermonde hashes should be provided to perform the hash checking. Computing each of them costs n finite-field multiplications. Thus, the computational complexity is $pdn + p\ell dn$.
- For the LW, it needs to calculate $d\ell$ Vandermonde hashes. Directly computing each of them also costs n finite-field multiplications; the computational complexity of LW is $d\ell np$.

In summary, we see that VANDER has a constant (i.e., d and γ) communication overhead and the computational complexity is linear with respect to the block length n of the RLNC vector, which is the same as the computational complexity of the basic subspace checking scheme.

C. Delay Reduction

Note that VANDER can introduce delay when watchdogs cooperate with one another to collect the Vandermonde hashes due to the information exchange and the carrier-sensing mechanism in 802.11 medium access control (MAC) for preventing simultaneous transmissions on the same channel when watchdogs cooperate. However, with the multiradio and multichannel technique, the wireless node can receive and transmit at the same time on nonoverlapping orthogonal channels; thus, full duplex communication can be then achieved [35]. Specifically, for a watchdog in the monitoring set $\bar{\mathcal{U}}_i = \{W_{ik} : k = 1, 2, \ldots, \lambda\}$, each watchdog is assigned with an orthogonal channel for cooperation. When watchdogs cooperate with one another to exchange overheard information, instead of waiting for other watchdogs to transmit, each watchdog works on the preassigned channel to transmit overheard information while receiving the information in other $\gamma - 1$ channels. Furthermore, for watchdogs in the neighboring monitoring sets, we should use suitable channel allocation [36], [37], so that orthogonal channels are assigned to watchdogs to prevent radio interference and allow simultaneous transmissions for watchdogs among neighboring monitoring sets.

By using the multiradio and multichannel approach, in the worst case, there are three more times of transmissions among watchdogs for the watchdog cooperation compared with the normal watchdog scheme [20]–[22], [30]. However, different from the wireless routing scheme, where a received packet at the internal node is directly forwarded to the next-hop node, the internal node in the network coding scheme has to wait for packets from upstream nodes to encode a new packet. Thus, we can implement this watchdog scheme with a pipeline-like method. More precisely, the downstream node can keep encoding the valid data packets when the current received packet is in verification and then encode the current received packet once it has passed the verification.

D. Tradeoff Discussion

- d and P_m. Recall that d is the number of Vandermonde hashes used in VANDER. As shown in Theorem 1, the misdetection probability is at most $(n/q)^d$. We assume $q > 2n$,[4] to achieve desired misdetection probability P_m, so that we can set $d = -\log P_m$.
- γ and P_f. Recall also that γ is the number of watchdogs in the monitoring set, and the false alarm probability is at most $(1-(1-p)^l)^\gamma(1-(1-p)^{2\gamma}(1-p^2)^\ell)$ (cf. Theorem 1). As $P_f \le (1-(1-p)^\ell)^\gamma$, to achieve the desired false alarm probability P_f, we set $\gamma \ge \log P_f / \log(1-(1-p)^\ell)$.

As we have analyzed in Section V-A and B, the computational complexity and the communication overhead are related to d or (and) γ. By choosing $d = -\log P_m$ and $\gamma \ge \log P_f / \log(1-(1-p)^\ell)$, VANDER preserves computational complexity $\Theta(-\ell n \log P_m)$ for each watchdog and the upper bounded communication overhead $\Theta(-\ell \log P_m + (\ell - \log P_m)\log P_f / \log(1-(1-p)^\ell))$.

[3]Note that the basic setting of RLNC would introduce a communication overhead for tracking the network linear transform [27]. Here, we focus on the overhead introduced in VANDER and do not consider the overhead that is due to RLNC.

[4]We note that $q > 2n$ holds for typical scenarios. For instance, for the case where each symbol is represented by 4 B, we have $q = 2^{32}$. In this case, n can be as large as 2^{16}.

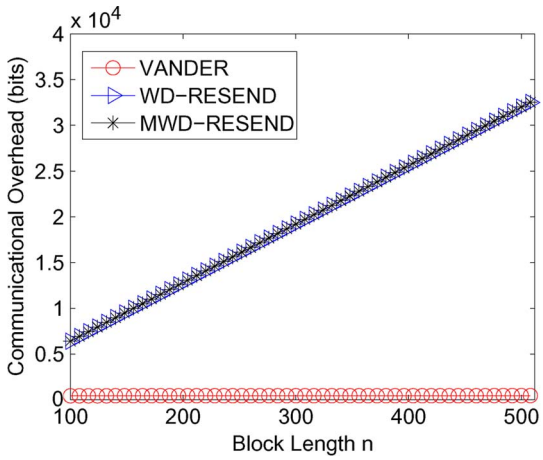


Fig. 4. Communication overhead comparison among VANDER, WD-RESEND, and MWD-RESEND, where the symbol length is fixed to 32 b and the required misdetection probability is less than 10^{-13}.

VI. Numerical Results on the Performance Evaluation

Here, we provide numerical results to evaluate the performance of VANDER according to the analysis in Section V. In the following, we consider a setup similar to that in Fig. 2. Let the number of incoming packets that generate the outgoing packet be $\ell = 8$ and the probability of lossy overhearing be $p = 0.25$. Moreover, we assume that the number of watchdogs in a monitoring set is $\gamma = 3$ unless noted otherwise. Note that as VANDER does not depend on the topology, the following results can be directly applied to other network topologies and network scales, as long as a transmission node is monitored by at least two watchdogs.

A. Communication Overhead

Fig. 4 compares the communication overhead among VANDER, WD-RESEND, and MWD-RESEND. Recall that, in WD-RESEND, upstream nodes retransmit all packets that the watchdog has missed. In MWD-RESEND, multiple watchdogs share the whole missed packets. In Fig. 4, we choose $P_m \leq 10^{-13}$; then, we can get the value of d, which is approximately 2. Moreover, we fix the symbol length (i.e., $\log_2 q$) to be 32 b and vary the block length n carefully. As shown in Fig. 4, when packet loss happens at watchdogs, both the overhead of WD-RESEND and MWD-RESEND increase linearly with the block length, whereas the overhead of VANDER stays constant. It verifies the analysis in Section V. Moreover, since, in VANDER, the introduced overhead only happens among watchdogs, the normal transmission of internal nodes would not be affected.

Fig. 5 compares the per-packet communication overhead with other two cryptographic schemes [11] and [14]. As both of these two cryptographic schemes are MAC based, the overhead comes from the tags that are appended in each packet. As shown in Fig 5, compared with [11] and [14], VANDER significantly reduces the communication overhead by four and five times, respectively.

In Fig. 6, we observe the communication overhead by increasing the number of watchdogs γ. As shown in Fig. 6, the

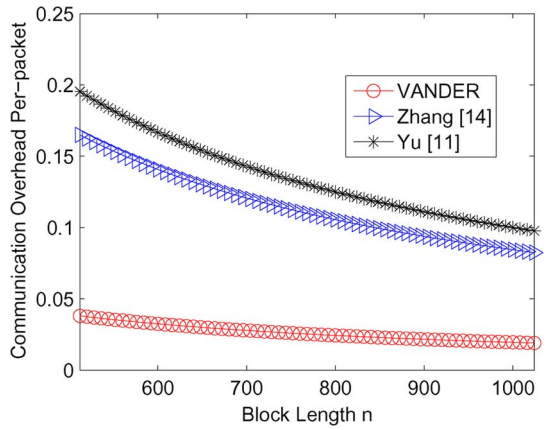


Fig. 5. Communication overhead of VANDER and other two MAC schemes. The symbol length is 64 b, and the misdetection probability in VANDER is less than 10^{-13}, whereas the misdetection probability in [14] is 0.01%.

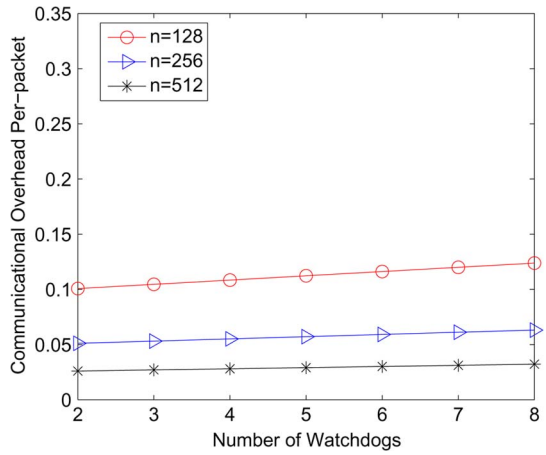


Fig. 6. Communication overhead of VANDER by varying γ. The symbol length is fixed to 32 b, and the required misdetection probability is less than 10^{-13}. The block length is chosen as 128, 256, and 512, respectively.

communication overhead just increases a little by varying γ from 2 to 8 under different block lengths. Specifically, the maximum overhead increase is just less than 5% when the block length is 128. If we choose larger block length, e.g., 256, then the overhead increase is much smaller when increasing γ.

B. Computational Complexity

We carry out experiments to evaluate the computational time of VANDER verification. The experiment is based on the implementation of fast Galois-field multiplications with C/C++ library [38]. We conduct experiments on the Intel Core i5 CPU 2.4-GHz Linux machine. We also assume that the symbol length is 32 b and that the misdetection probability of VANDER and [14] are less than 10^{-13} and 1%, respectively. Approximately 1.9×10^7 multiplications per second are conducted in the experiments. In Fig. 7, we plot the computational complexity of VANDER and four other cryptographic schemes: Gkantsidis [5], Yu [7], Zhao [9] and Zhang [14]. As shown in Fig. 7, VANDER is at least eight times faster than first three cryptographic schemes and six times faster than [14] in average while achieving much higher security guarantee in terms of the misdetection probability.

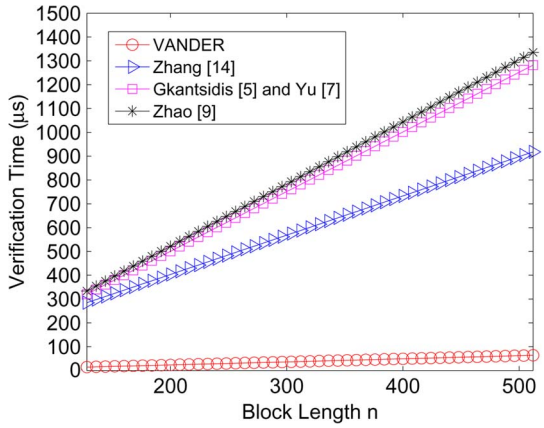


Fig. 7. Computational complexity of VANDER and other four cryptographic schemes, where the time to execute multiplications in each scheme is compared. The symbol length is 32 b, and the misdetection probability of VANDER and [14] are less than 10^{-13} and 1%, respectively.

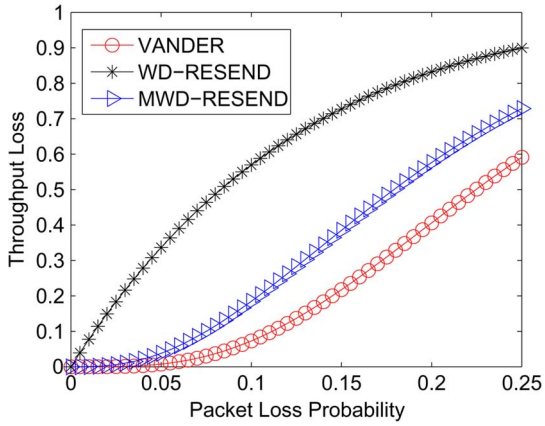


Fig. 8. Throughput loss in different schemes. We compare the influence of false alarm on the throughput loss with WD-RESEND and MWD-RESEND. We assume the number of watchdogs in each monitoring set is three in both MWD-RESEND and VANDER.

C. Throughput

Here, we show the results on the throughput, which is equivalent to the false alarm probabilities in this problem. We have proved the upper bound of false alarm probability in Section IV-E. Now, let us evaluate the throughput improvement of VANDER (not just the upper bound). We vary the packet loss probability at the watchdog to see the throughput loss.

In Fig. 8, we see that VANDER achieves more throughput than that in the common watchdog schemes. Specifically, when the throughput loss is close to 90% and 73% in the WD-RESEND and MWD-RESEND, respectively, VANDER can still preserve approximately 45% of the throughput.

In Fig. 9, we see that VANDER can achieve higher throughput by increasing the number of watchdogs. Note that when the packet loss probability is high ($p = 0.25$), increasing the number of watchdogs to eight does not improve much performance. Then, in general, choosing γ as 3 or 4 achieves high network performance while introducing a small communication overhead (see Fig. 6).

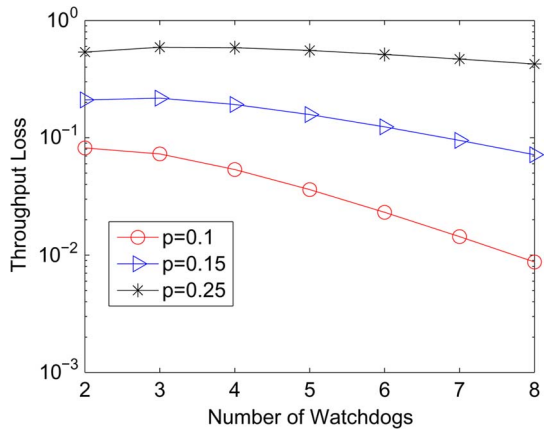


Fig. 9. Throughput loss by varying γ. We choose three kinds of packet loss probabilities, 0.1, 0.15, and 0.25.

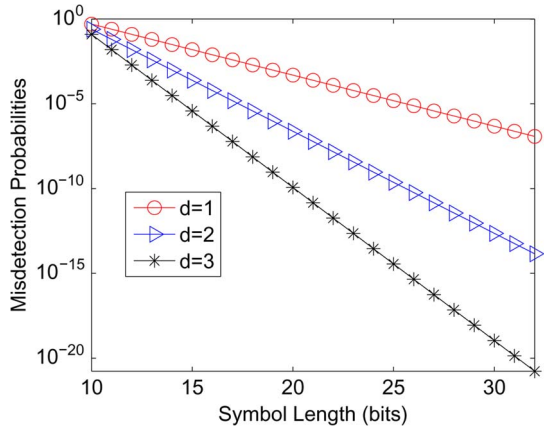


Fig. 10. Misdetection probabilities under different symbol lengths and the number of Vandermonde hashes, where the block length n is fixed to be 512.

D. Misdetection Probability

As shown in Section IV-E, the misdetection probability is mainly dependent on the symbol length and the number of Vandermonde hashes. In Fig. 10, we assume that the block length in each packet is fixed to be 512, and the number of Vandermonde hashes d is chosen from {1, 2, 3}. Then, we vary the symbol length to show its influence on the misdetection probability. Fig. 10 shows that the misdetection probability decreases exponentially when the symbol length increases. Moreover, if a larger number of Vandermonde hashes is chosen, the misdetection probability drops sharply. As shown in Fig. 10, when the symbol length reaches 24 b, the misdetection probability can be arbitrarily low, even when d is equal to 1. This means that we can achieve arbitrarily low misdetection probability while introducing a small overhead when the symbol length and the block length are chosen properly.

VII. Conclusion

In this paper, we have investigated pollution attacks of lossy network coding in heterogeneous wireless networks. In particular, helper nodes in heterogeneous wireless networks played the role of watchdogs that can overhear and monitor packet transmissions from the same wireless node. We proposed a novel watchdog cooperative monitoring scheme, i.e.,

VANDER, to address pollution attacks in the lossy wireless environment. In VANDER, low overhead packets that contained random Vandermonde hashes were introduced to share packet information and check the validity of packets when lossy overhearing happened at watchdogs. As each column subspace of a Vandermonde matrix could be calculated efficiently, only a small overhead was introduced among watchdogs, and no extra overhead was introduced to normal transmission nodes due to the cooperation. Moreover, even successive colluded adversarial nodes could be detected. We demonstrated that VANDER had provably high security guarantee, low computational complexity (linear respect to the block length n), and a constant upper bounded communication overhead when $2n$ was no more than the field size (whose typical value is 2^{32}).

APPENDIX
PROOF OF THEOREM 1

To prove the "false alarm" part, we note that if $\mathbf{z}$ is legitimate, we have

$$\mathbf{z} = \sum_{i=1}^{\ell} \alpha_i \mathbf{z}_i. \tag{3}$$

Thus, due to the linearity of inner product, for each $j = 1, 2, \ldots, d$, we always have

$$\tilde{\mathbf{a}}_j^T \mathbf{z} = \sum_{i=1}^{\ell} \alpha_i \tilde{\mathbf{a}}_j^T \mathbf{z}_i. \tag{4}$$

As in VANDER, if any one of the γ watchdogs in $\bar{\mathcal{U}}$ can collect enough information to check either (3) or (4), the outgoing packets from the honest node can always pass verification. Then, false alarms do not happen.

Otherwise, if none of the watchdogs receives enough packets for calculating (3) and (4) due to packet loss among watchdogs, false alarm happens. Let the packet loss probability at each watchdog be p (independent of one another), Pr be the probability of an event, ℓ be the number of encoded incoming packets, and γ be the number of watchdogs in the monitoring set. Before computing the false alarm probability P_f, we define four events as follows.

- Event E_1 is the event when incoming packets loss happens at the watchdog. The probability that E_1 happens is $(1-(1-p)^l)^\gamma$.
- Event E_2 is the event when at least one of the watchdogs can receive all VAPs. As there are $\gamma - 1$ VAPs that a watchdog needs to receive, the probability that E_2 happens is $1-(1-(1-p)^{(\gamma-1)})^\gamma$.
- Event E_3 is the event when all watchdogs receive the VLP. As there are γ watchdogs, the probability that E_3 happens is at least $(1-p)^{(\gamma-1)}$.
- Event E_4 is the event when LW receives the hashes of all the missed incoming packets. The probability of E_4 is defined as $\Pr(E4) = \sum_{i=0}^{l} \Pr(E_4') \Pr(E_4'')$, where E_4' represents that LW misses i ($i \in [0,\ \ell]$) incoming packets and E_4'' represents that LW receives all i corresponding missing hashes. Thus, $\Pr(E_4) = \sum_{i=0}^{\ell} \binom{\ell}{i} p^i (1-p)^{(\ell-i)} (1-p)^i = (1+p)^\ell (1-p)^\ell = (1-p^2)^\ell$.

Note that, in VANDER, false alarm happens if and only if E_1 happens and if $E_2, E_3,$ and E_4 do not happen at the same time. Thus, the false alarm probability is

$$P_f = \Pr(E_1)(1 - \Pr(E_2)\Pr(E_3)\Pr(E_4)). \tag{5}$$

Assume that $\delta = (1-p)^{(\gamma-1)}$; then, the probabilities of E_2 and E_3 can be rewritten as $\Pr(E_2) = 1-(1-\delta)^\gamma$ and $\Pr(E_3) = \delta$, respectively. Therefore

$$\begin{aligned} \Pr(E_2) - \Pr(E_3) &= 1-(1-\delta)^\gamma - \delta \\ &= 1-\delta-(1-\delta)^\gamma \geq 0. \end{aligned} \tag{6}$$

Then, (5) becomes

$$\begin{aligned} P_f &= \Pr(E_1)\left(1 - \Pr(E_2)\Pr(E_3)\Pr(E_4)\right) \\ &\leq \Pr(E_1)\left(1 - \Pr(E_3)^2 \Pr(E_4)\right) \\ &= \left(1-(1-p)^l\right)^\gamma \left(1-(1-p)^{2(\gamma-1)}(1-p^2)^\ell\right) \\ &\leq \left(1-(1-p)^l\right)^\gamma \left(1-(1-p)^{2\gamma}(1-p^2)^\ell\right). \end{aligned} \tag{7}$$

Then, we prove the "misdetection" part. First, we compute the upper bound on the "misdetection" probability of VANDER as follows.

If all the incoming packets and the outgoing packet are successfully overheard by any one of watchdogs in the same monitoring set, this watchdog can correctly verify the outgoing packet through packet subspace checking. Thus, no misdetection would happen.

Otherwise, let $\mathbf{z}' = \sum_{i=1}^{\ell} \alpha_i \mathbf{z}_i$ be the corresponding legitimate RLNC vector that follows the coding rule. Then, misdetection happens if and only if $\mathbf{z} \neq \mathbf{z}'$, but

$$\tilde{\mathbf{a}}_j^T(\mathbf{y}' - \mathbf{y}) = 0 \ \text{ for each } j = 1, 2, \ldots, d. \tag{8}$$

Note that, for each $j = 1, 2, \ldots, d$, $\tilde{\mathbf{a}}_j = [\tilde{a}_j^1, \tilde{a}_j^2, \ldots, \tilde{a}_j^n]^T$. Thus, $\tilde{\mathbf{a}}_j^T(\mathbf{z}' - \mathbf{z})$ is in fact a nonzero polynomial of $\tilde{a}_j$ with degree n. Using the Schwartz–Zippel Lemma [27], when $\tilde{\mathbf{a}}_j$ is randomly chosen over $\mathbb{F}_q^n$ with uniform probability, $\tilde{\mathbf{a}}_j^T(\mathbf{y}' - \mathbf{y}) = 0$, with probability at most n/q. Since each of $\{\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_j\}$ is independently chosen, (8) is true with probability at most $(n/q)^d$. This completes the proof of Theorem 1. □

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that improved the quality of this paper.

REFERENCES

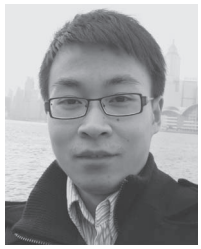
[1] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[2] T. Ho *et al.*, "Byzantine modification detection in multicast networks using randomized network coding," in *Proc. IEEE ISIT*, Jun./Jul. 2004, p. 143.

[3] S. Jaggi *et al.*, "Resilient network coding in the presence of Byzantine adversaries," *IEEE Trans. Inform. Theory*, vol. 54, no. 6, pp. 2596–2603, Jun. 2008.

[4] M. Krohn, M. Freedman, and D. Mazieres, "On-the-fly verification of rateless erasure codes for efficient content distribution," in *Proc. IEEE Symp. Security*, May 2004, pp. 226–240.

[5] C. Gkantsidis and P. Rodriguez, "Cooperative security for network coding file distribution," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–13.
[6] D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," in *Proc. Conf. Inf. Sci. Syst.*, Mar. 2006, pp. 857–863.
[7] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature-based scheme for securing network coding against pollution attacks," in *Proc. of IEEE INFOCOM*, Apr. 2008, pp. 2083–2091.
[8] A. Yun, J. H. Cheon, and Y. Kim, "On homomorphic signatures for network coding," *IEEE Trans. Comput.*, vol. 59, no. 9, pp. 1295–1296, Sep. 2010.
[9] F. Zhao, T. Kalker, M. Médard, and K. Han, "Signatures for content distribution with network coding," in *Proc. IEEE ISIT*, Jun. 2007, pp. 556–560.
[10] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a linear subspace: Signature schemes for network coding," in *Proc. Int. Conf. Practice Theory Public Key Cryptogr.*, 2009, pp. 68–87.
[11] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient scheme for securing XOR network coding against pollution attacks," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 406–414.
[12] S. Agrawal and D. Boneh, "Homomorphic MACs: MAC-based integrity for network coding," in *Proc. Appl. Cryptogr. Netw. Security*, 2009, pp. 292–305.
[13] Y. Li, H. Yao, M. Chen, S. Jaggi, and A. Rosen, "RIPPLE authentication for network coding," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[14] P. Zhang *et al.*, "Padding for orthogonality: Efficient subspace authentication for network coding," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1026–1034.
[15] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks," in *Proc. ACM Conf. Wireless Netw. Security*, 2009, pp. 111–122.
[16] A. Le and A. Markopoulou, "Cooperative defense against pollution attacks in network coding using SpaceMac," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 442–449, Feb. 2012.
[17] A. Newell and C. Nita-Rotaru, "Split null keys: A null space based defense for pollution attacks in wireless network coding," in *Proc. IEEE Commun. Soc. Conf. Sens., Mesh and Ad Hoc Commun. Netw.*, Jun. 2012, pp. 479–487.
[18] X. Wu, Y. Xu, C. Yuen, and L. Xiang, "A tag encoding scheme against pollution attack to linear network coding," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 33–42, Jan. 2014.
[19] A. Newell, J. Dong, and C. Nita-Rotaru, "On the practicality of cryptographic defenses against pollution attacks in wireless network coding," *ACM Comput. Surveys*, vol. 45, no. 3, pp. 39:1–39:26, Jun. 2013.
[20] M. J. Kim, R. Kötter, M. Médard, and J. Barros, "An algebraic watchdog for wireless network coding," in *Proc. IEEE ISIT*, Jun./Jul. 2009, pp. 1159–1163.
[21] M. J. Kim, M. Médard, and J. Barros, "A multi-hop multi-source algebraic watchdog," in *Proc. IEEE ITW*, Aug./Sep. 2010, pp. 1–5.
[22] G. Liang, R. Agarwal, and N. Vaidya, "When watchdog meets coding," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[23] P. Li and Y. Guang, "The capacity of heterogeneous wireless networks," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[24] S. Yang, X. Wang, and L. Fu, "On the topology of wireless sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2095–2103.
[25] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 371–381, Feb. 2003.
[26] R. Kötter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.
[27] T. Ho, R. Kötter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE ISIT*, Jun./Jul. 2003, pp. 1–6.
[28] D. Silva, F. R. Kschischang, and R. Kötter, "A rank-metric approach to error control in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 3951–3967, Sep. 2008.
[29] T. K. Dikaliotis *et al.*, "Multiple-access network information-flow and correction codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1067–1079, Feb. 2011.
[30] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile *ad hoc* networks," in *Proc. 6th Mobicom*, 2000, pp. 255–265.
[31] M. J. Kim, M. Médard, and J. Barros, "Algebraic watchdog: Mitigating misbehavior in wireless network coding," *IEEE J. Select. Areas Commun.*, vol. 29, no. 10, pp. 1916–1925, Dec. 2011.
[32] R. W. Yeung, S. Y. R. Li, N. Cai, and Z. Zhang, *Network Coding Theory (Foundations and Trends (R) in Communications and Information Theory)*. Hanover, MA, USA: Now, 2006.
[33] Y. Liu, P. Ning, and H. Dai, "Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures," in *Proc. IEEE Symp. Security Privacy*, May 2010, pp. 286–301.
[34] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A measurement study of a large-scale P2P IPTV system," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1672–1687, Dec. 2007.
[35] I. Ho, P. Lam, P. Chong, and S. Liew, "Harnessing the high bandwidth of multi-radio multi-channel 802.11n mesh networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 2, pp. 448–456, Feb. 2014.
[36] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *Proc. IEEE MobiCom*, 2005, pp. 58–72.
[37] K. Ramachandran, E. Belding-Royer, K. Almeroth, and M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–12.
[38] J. S. Plank, "Fast Galois Field Arithmetic Library in C/C++," Univ. Tennessee, Knoxville, TN, USA, Tech. Rep. UT-CS-07-593, 2007.



Xin Lou (S'13) received the B.Eng. degree (first-class honor) in communication engineering from Sichuan University, Chengdu, China, in 2009. He is currently working toward the Ph.D. degree with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong.

He was a Software Engineer with the Advanced Digital Sciences Center: a Singapore-based research center established by the University of Illinois at Urbana-Champaign, in 2011. His research interests include power networks, wireless networks, nonlinear optimization, and distributed algorithms.

Hongyi Yao (M'10) received the B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 2007 and 2010, respectively.

He was a Postdoctoral Researcher with California Institute of Technology, Pasadena, CA, USA. He is currently with Tower Research LLC, New York, NY, USA. His research interests include secure network transmission, reliable network control, and secure wireless communications.

Chee Wei Tan (M'08–SM'12) received the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, USA, in 2006 and 2008, respectively.

He was a Postdoctoral Scholar with the California Institute of Technology, Pasadena, CA, USA. In 2011, he was a Visiting Faculty with Qualcomm R&D, San Diego, CA. He is currently an Assistant Professor with the City University of Hong Kong, Kowloon, Hong Kong. His research interests include networks, inference in online large data analytics, and optimization theory and its applications.

Dr. Tan is the Chair of the IEEE Information Theory Society Hong Kong Chapter. He currently serves as an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS. He received the 2008 Princeton University Wu Prize for Excellence and the 2011 IEEE Communications Society Asia-Pacific Outstanding Young Researcher Award. He was a selected participant at the U.S. National Academy of Engineering China–America Frontiers of Engineering Symposium in 2013.

Jianping Wang (M'03) received the B.Sc. and M.Sc. degrees from Nankai University, Tianjin, China, in 1996 and 1999, respectively, and the Ph.D. degree from The University of Texas at Dallas, Richardson, TX, USA, in 2003.

She is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. Her research interests include dependable networking, optical networking, and service-oriented wireless sensor/ad hoc networking.