# Enhancing the Anonymity in Information Diffusion Based on Obfuscated Coded Data

Jin Wang, *Member, IEEE*, Kejie Lu, *Senior Member, IEEE*, Jianping Wang, *Member, IEEE*, Chuan Wu, *Member, IEEE*, and Naijie Gu

**Abstract**—Linear network coding (LNC) is a promising approach to facilitate anonymity in information diffusion because each packet is generated by linearly combining multiple incoming packets. Since the coefficients used in the linear combination would reveal the correlation between incoming and outgoing packets at a node, most existing studies on anonymous LNC design focus on encrypting these coefficients. Despite the importance of these studies, the correlation of coded content can still be analyzed and the potential of un-encrypted LNC has not been fully exploited. In this paper, we tackle these issues and we propose a novel ALNCode scheme that can enhance anonymity by generating outgoing packets that are correlated to incoming coded packets of multiple flows. With solid theoretical analysis, we first prove the probability that incoming coded packets from different flows are correlated. Then, we prove that, if such correlation exists, we can design deterministic LNC to obfuscate the correlation of packets. With the same condition, we also prove the probability that a randomly generated coded packet is correlated to coded packets in other flows. Besides the theoretical study, we conduct extensive numerical experiments to understand the impacts of various coding parameters and the performance of ALNCode in real scenarios.

**Index Terms**—Anonymity, information diffusion, network coding, secure linear network coding, deterministic linear network coding, random linear network coding, traffic analysis

## 1 Introduction

In recent years, privacy and anonymity have become increasingly important for information diffusion and propagation in various network scenarios, e.g., social networks, content delivery networks, etc. [1], [2], [3], [4], [5], [6]. In a network, when an attacker can only observe a snapshot of the spread of a content at a certain time, information diffusion schemes have been designed to obfuscate the real source by multiple pseudo sources [7], [8].

In this paper, we consider a stronger attack model, in which an attacker can continuously monitor the network flows within a time period. In this case, to enable anonymous communication, a critical issue is how to preserve *flow untraceability*, which aims to hide the routing path, source, and receiver from malicious attackers with wiretapping and traffic analysis capabilities.


- *J. Wang is with the Department of Computer Science and Technology, Soochow University, Suzhou 215000, China, and also with the Department of Computer Science, City University of Hong Kong, Hong Kong. E-mail: wjin1985@suda.edu.cn.*
- *K. Lu is with the Department of Computer Science and Engineering, University of Puerto Rico at Mayagüez, Mayagüez, Puerto Rico 00682. E-mail: kejie.lu@upr.edu.*
- *J. Wang is with the Department of Computer Science, City University of Hong Kong, Hong Kong. E-mail: jianwang@cityu.edu.hk.*
- *C. Wu is with the Department of Computer Science, University of Hong Kong, Hong Kong. E-mail: cwu@cs.hku.hk.*
- *N. Gu is with the Department of Computer Science, University of Science and Technology of China, Hefei 230022, China. E-mail: gunj@ustc.edu.cn.*

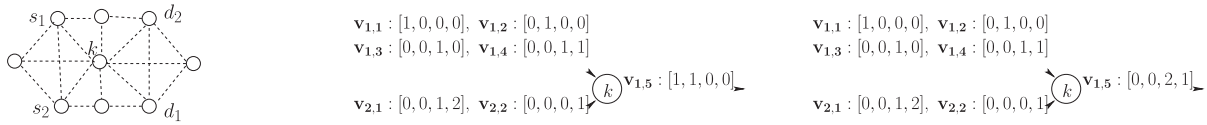

*Manuscript received 7 July 2018; revised 10 Nov. 2018; accepted 11 Dec. 2018. Date of publication 21 Dec. 2018; date of current version 3 Dec. 2019. (Corresponding author: Kejie Lu.)*
*Recommended for acceptance by X. Cao.*
*Digital Object Identifier no. 10.1109/TNSE.2018.2888848*

To provide flow untraceability against traffic analysis attacks, the basic requirement is to protect the routing information using secure routing protocols [9], [10], [11]. In addition to that, traditional approaches [12], [13] have to: (1) hide the content correlation among packets with per-hop encryption, (2) hide the size correlation of packets by padding packets with random symbols, and (3) hide the time correlation among packets of the same flow by mixing the order of packet transmissions from different flows at intermediate nodes. Clearly, these schemes are computationally expensive.

To achieve the same objectives with much better efficiency, a promising technology is *linear network coding* (LNC) [14], [15]. In LNC, a group of original packets in a flow, known as a *generation*, are used to generate coded packets at the source node. Each coded packet has a *Global Encoding Vector* (GEV), which consists of the coding coefficients to produce the coded packet from the set of original packets. At an intermediate node, each outgoing packet of a flow is generated by linearly combining the incoming packets of the same flow, where we define a *Coded Data Vector* (CDV) as a vector that includes both GEV and coded payload.

Therefore, LNC naturally avoids copying packets in the same flow, which may conceal the content correlation (to be investigated in this paper) without using computationally expensive encryption. Moreover, with LNC, coded packets can have an equal size and are buffered at intermediate nodes to generate new coded packets, naturally preventing correlating packet sizes and arrival time patterns. Nevertheless, given the encoding mechanisms of LNC, linear dependency among GEVs of coded packets may reveal information of the

2327-4697 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.




(a) A network with three unicast sessions. (b) A GEV generated without considering annonymity. (c) An *obfuscated* GEV generated.

Fig. 1. *ALNCode* design: An example.

flow path, if a wiretapper analyzes the correlation between incoming GEVs and outgoing ones.

We now use an example in Fig. 1 to demonstrate this problem. In Fig. 1, there are 2 unicast flows in the network, denoted as $f_i$ ($i \in \{1,2\}$) with source $s_i$ and destination $d_i$. We consider that each flow can utilize multiple paths and we also assume that all packets are generated by LNC on the finite field $\mathbb{F}_3$, with 4 packets in a generation. At a certain time epoch, we assume that node $k$ buffered a total of 6 packets, in which there are four coded packets from flow $f_1$ with GEVs $\mathbf{v}_{1,j}, j \in \{1,2,3,4\}$ and two coded packets from flow $f_2$ with GEVs $\mathbf{v}_{2,1}$ and $\mathbf{v}_{2,2}$. For each flow, the intermediate node $k$ only buffers coded packets with linearly independent GEVs and generates new outgoing coded packets by linearly combining these buffered coded packets.

In Fig. 1b, node $k$ can generate a new coded packet for flow $f_1$ by adding two received packets with linearly independent GEVs $\mathbf{v}_{1,1}$ and $\mathbf{v}_{1,2}$, and derive the new GEV as $\mathbf{v}_{1,5} = \mathbf{v}_{1,1} + \mathbf{v}_{1,2}$. In this example, from the attacker's point of view, since the intermediate node $k$ only buffers coded packets with linearly independent GEVs for each flow and the size of generation is 4, the new GEV, i.e., $\mathbf{v}_{1,5}$, is generated by no more than four linearly independent GEVs. Since $\mathbf{v}_{1,5}$ can only be generated by adding $\mathbf{v}_{1,1}$ and $\mathbf{v}_{1,2}$, the attacker can tell these three packets belong to the same flow.

To hide the correlation among GEVs, most existing studies suggest to encrypt GEVs [4], [5], [6]. In [4], [5], the authors proposed different schemes to first share a secret key between the source and the destination, then apply a *homomorphic encryption function* that allows intermediate nodes to produce new encrypted GEVs without knowing the secret key, and finally let the destination decrypt received GEVs with the pre-shared secret key. On the other hand, based on the ideas of shift cipher, Zhang et al. proposed to reorder the content in a coded packet at the source node such that the GEV is permutated in coded payload [6]. Although the existing anonymous LNC schemes can hide the correlation among GEVs, we note that part of each coded packet is shown in plaintext, e.g., the payload in [4] or the reordered GEV and payload in [6]. Clearly, an adversary can analyze the linear correlation using the whole (or any part of the) coded packets, i.e., CDVs. Therefore, in this paper, we will investigate the design of LNC to defend such traffic analysis attacks, including GEV analysis attack from an attacker with limited computation capability, and CDV analysis attack from an attacker with higher computation capability.

Specifically, although the linear correlations of incoming and outgoing vectors can be utilized by the attacker to infer the flow path, we can also utilize such linear correlations to generate obfuscated vectors to make the outgoing vectors have linear correlations with not only the incoming vectors from the same flow but also those from other flows, which can efficiently provide the flow anonymity without using encryption. Based on this idea, we will propose a novel anonymous LNC scheme, *ALNCode*, that can efficiently achieve flow untraceability in a communication network with multiple unicast flows.

We now illustrate our idea by using an example in Fig. 1c. Different to the case in Fig. 1b, node $k$ now generates a new GEV[1] by $\mathbf{v}_{1,5} = \mathbf{v}_{1,3} + \mathbf{v}_{1,4}$. The new GEV is now *obfuscated*, because $\mathbf{v}_{1,5} = 2\mathbf{v}_{2,1} = 2\mathbf{v}_{1,3} + \mathbf{v}_{2,2} = 2\mathbf{v}_{1,4} + 2\mathbf{v}_{2,2}$. As a result, $\mathbf{v}_{1,5}$ is not only correlated with $\{\mathbf{v}_{1,3}, \mathbf{v}_{1,4}\}$, but also $\{\mathbf{v}_{2,1}\}, \{\mathbf{v}_{1,3}, \mathbf{v}_{2,2}\}, \{\mathbf{v}_{1,4}, \mathbf{v}_{2,2}\}$. It means that the newly generated GEV for flow $f_1$ is correlated with GEVs in flow $f_2$. Therefore, to any traffic analysis attacker that tries to correlate the incoming and outgoing GEVs, it would not be able to tell accurately which packets belong to the same flow.

Next, we first summarize the main contributions of this paper and then explain the new contributions by comparing this paper with its conference version [1].

- ▷ We propose a novel anonymous LNC scheme, *ALNCode*, based on the idea to generate obfuscated *coded data*, *i.e*, GEVs or CDVs, which can provide anonymity in networks with multiple unicast flows.
- ▷ We theoretically prove the probability that incoming coded data from different flows are correlated. We also conduct extensive numerical experiments to evaluate the impact of various LNC parameters.
- ▷ We prove that, if there exists correlation between incoming coded data in different flows, then we can generate obfuscated coded data using the *ALNCode*.
- ▷ Given the correlation requirement for incoming coded data, we design an efficient deterministic LNC scheme such that outgoing coded data are guaranteed to obfuscate their correlation with the incoming coded data in different flows.
- ▷ Given the same conditions, we also give theoretical analysis to show the potential of using the standard random LNC to thwart traffic analysis attacks.
- ▷ We conduct solid security analysis for the *ALNCode* against GEV (or CDV) analysis attacks.

Compared with our prior work [1], the new contributions include: (1) We extend our ideas for defending against the GEV analysis attack to the case that the attacker has higher computation capability and thus can analyze the whole CDVs. To this end, we consider GEVs and CDVs as vectors in general and conduct theoretical analysis and numerical experiments to investigate the probability that there exists an obfuscated vector in Section 3. (2) In addition to designing deterministic LNC, we also theoretically analyze the lower bound of the probability that a standard random LNC can

1. Here we still use the GEV as example but the same idea can be applied to CDV as well.

produce an obfuscated GEV or CDV in Section 3.4. (3) We provide new theoretical analysis and simulation results to evaluate the performance of the proposed schemes against traffic analysis attack in Sections 4.1 and 4.2, for which we introduce a new concept, i.e., *anonymity level*, in Section 2. (4) We add discussions on the impact of parameters selection in Section 4.3 and discussions on the implementation of the *ALNCode* scheme in Section 4.4. (5) We discuss the relationships of our work with more related work from different aspects in Section 5, which covers the state-of-the-art.

The rest of the paper is organized as follows. We formally present the models of linear network coding and traffic analysis attacks in Section 2. In Section 3, we first prove the probability that incoming coded packets from different flows are correlated. Then, we prove that, if such correlation exists, we can design deterministic LNC such that the outgoing coded packets of a flow are guaranteed to be correlated to incoming coded packets in other flows, which can obfuscate the correlation of packets in a flow. With the same condition, we also prove the probability that a randomly generated coded packet is correlated to coded packets in other flows. Section 4 discusses how our mechanism can efficiently defend against both GEV analysis attack and CDV analysis attack. We discuss related work in Section 5 and conclude the paper in Section 6.

## 2 Anonymous Communication Model with LNC

In this section, we present the network, LNC model, and the traffic analysis attacks studied in this paper.

### 2.1 Network and Linear Network Coding Models

We consider a communication network with multiple unicast flows between multiple pairs of *source* and *destination* nodes. We assume that the network topology for transmission is fixed, and we assume that an LNC scheme (e.g., the scheme in [14]) is already used in the network to deliver multiple unicast flows, so most functions in existing LNC schemes, including routing, will be used. Specifically, each flow has a unique flow number and may go through multiple simple paths, where a simple path consists of a sequence of links with no duplicated link. For each unicast flow, the existing LNC scheme can determine all the simple paths and the number of CDVs transmitted on each path. Usually, edge-disjoint is not required when the LNC selects paths. The paths of different flows may intersect at common intermediate nodes (e.g., node $k$ in Fig. 1a).

With LNC, a source node partitions the data flow into *data blocks* of the fixed size $H$, and every $h$ consecutive data blocks in the flow form a *generation*. LNC is performed among data blocks in the same generation of a flow. A coded packet transmitted on each link consists of the coded data block and the GEV representing the coding coefficients to produce the coded data block from the original data blocks.

*Source encoding*: Given original blocks $\{\mathbf{m}_1, \ldots, \mathbf{m}_h\}$ in generation $j$ of flow $i$, the source node selects $h$ linearly independent GEVs, $\{\mathbf{v}_1, \ldots, \mathbf{v}_h\}$, over vector space $\mathbb{F}_q^h$, and generates $h$ coded data blocks $\{\mathbf{m}'_1, \ldots, \mathbf{m}'_h\}$ using these GEVs. The $h$ coded data blocks are generated as follows, shown together with the GEVs:

$$\left[\mathbf{v}_n \mid \mathbf{m}'_n\right] = \left[\mathbf{v}_n \mid \sum_{l=1}^{h} v_{n,l}\mathbf{m}_n\right], \tag{1}$$

where $1 \leq n \leq h$ and $v_{n,l}$ is the $l_{th}$ element of vector $\mathbf{v}_n$.

*Intermediate Node Encoding*. Each intermediate node buffers coded packets received for a generation of a flow for $T$ time slots, and produces new coded packets for this generation from the buffered packets. Suppose the node has received $r$ coded data blocks $\{\mathbf{m}'_1, \ldots, \mathbf{m}'_r\}$ for generation $j$ of flow $i$ during time $T$, corresponding to $r$ GEVs $\{\mathbf{v}'_1, \ldots, \mathbf{v}'_r\}$. To generate a new coded packet, it produces a local encoding vector $\mathbf{c} = [c_1, \ldots, c_r]$ from vector space $\mathbb{F}_q^r$, and then generates the new coded data block $\mathbf{m}''$ together with a new GEV $\mathbf{v}''$ as:

$$[\mathbf{v}'' \mid \mathbf{m}''] = \left[\sum_{l=1}^{r} c_l\mathbf{v}'_l \mid \sum_{l=1}^{r} c_l\mathbf{m}'_l\right]. \tag{2}$$

*Destination Decoding*. After receiving $h$ coded data blocks $\{\mathbf{m}''_1, \ldots, \mathbf{m}''_h\}$ from generation $j$ of flow $i$ with linearly independent GEVs $\{\mathbf{v}''_1, \ldots, \mathbf{v}''_h\}$, a destination node recovers the original data blocks $\{\mathbf{m}_1, \ldots, \mathbf{m}_h\}$ by inverting the matrix composed by the GEVs:

$$\begin{bmatrix}\mathbf{m}_1\\ \vdots\\ \mathbf{m}_h\end{bmatrix} = \begin{bmatrix}\mathbf{v}''_1\\ \vdots\\ \mathbf{v}''_h\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{m}''_1\\ \vdots\\ \mathbf{m}''_h\end{bmatrix}. \tag{3}$$

In this paper, we use intra-session LNC, in which the outgoing vectors for a flow are generated by linearly combining the incoming vectors from the same flow. The design objective is different to the objectives in existing inter-flow LNC [16]. Specifically, our design aims to protect the anonymity of information diffusion by generating outgoing vectors for a flow that have linear dependency with incoming vectors from other flows.

In a practical communication network, each coded data block to be delivered is tagged with its routing information, flow number, generation number, and the GEV, which together is referred to as a *coded packet* and all coded packets in the network have an equal size [14], [15]. We also assume that a secure and anonymous routing protocol [9], [10], [11] is in place (similar to the assumptions made in [4] and [6]). With such a protocol, the routing information, flow and generation numbers attached to each coded packet are protected. On the other hand, GEVs and CDVs are not encrypted.

### 2.2 The Attack Model

We consider a passive wiretapping attacker with traffic analysis abilities from outside of the network. We assume that it can continuously monitor the network state within a time period. Specifically, it can observe all the packets along all the links in the network and analyze them, attempting to identify sources, destinations, and paths of the flows [4], [12], [13].

For the attacker, routing, flow, and generation information for each coded packet sniffed is hidden (by the secure and anonymous routing protocol), but GEVs and coded data blocks, i.e., the payload of the coded packet, are open. The coded data blocks of each coded packet and its corresponding GEV is referred as *coded data vector*. In this paper, we consider

TABLE 1
Notations

| Symbol | Definition |
|---|---|
| Symbol in **bold** font | Vectors, matrixes and linear spans |
| **Symbol** $^{\mathrm{T}}$ | transpose of a matrix or a vector |
| $\mathbb{F}_q$ | a finite field of size $q$, over which the LNC is defined. |
| $h$ | the number of data blocks in each generation of a flow |
| $H$ | the size of each data block |
| t | the number of elements in each vector. For GEVs, $\mathrm{t} = h$. For CDVs, $\mathrm{t} = h + H$. |
| $\mathbf{A}$ | the set of GEVs (or CDVs) received by node $k$ from generation $j$ of flow $i$ in the past $T$ time slots |
| $\mathbf{B}$ | the set of GEVs (or CDVs) received by node $k$ from flows other than $i$ in the past $T$ time slots |
| $R$ | $\dim(L(\mathbf{A} \cup \mathbf{B}))$ |
| $f_1$, $f_2$ | $f_1 = \lvert\mathbf{A}\rvert$; $f_2 = \lvert\mathbf{B}\rvert$ |
| $F$ | the total number of GEVs (or CDVs) received by node $k$ from all the flows in the past $T$ time slots |
| $L(\cdot)$ | linear span of a set of vectors. For a matrix $\mathbf{Y}$, $L(\mathbf{Y})$ is the row vector space of $\mathbf{Y}$ |
| $\mathrm{rank}(\mathbf{Y})$ | the rank of a matrix $\mathbf{Y}$ |
| $r_i$ | $r_1 = \dim(L(\mathbf{A}))$, $r_2 = \dim(L(\mathbf{B}))$ |
| $P(A)$ | the probability that condition A is satisfied |
| $P(A\|B)$ | the probability that condition A is satisfied when condition B is satisfied |
| $\overline{\mathbf{C}}$ | the matrix formed by nonzero vectors in the set of vectors $\mathbf{C}$ as its rows |
| $\mathbf{N}_{i,j,k}$ | the basis of vector space $L(\mathbf{A}) \cap L(\mathbf{B})$ |
| $N$ | $N = \lvert\mathbf{N}_{i,j,k}\rvert = \dim(L(\mathbf{A}) \cap L(\mathbf{B}))$ |
| $\mathbf{\Theta}_{i,j,k}$ | the obfuscated basis of $L(\mathbf{A})$ which is the basis of $L(\mathbf{A})$ extended from $\mathbf{N}_{i,j,k}$ |

two kinds of attackers: the first kind attacker has limited computation capability which can only analyze the GEVs of each coded blocks, the second kind attacker has high computation capability which can fully analyze the CDVs.

In this paper, we will design an LNC scheme to enhanced anonymity against traffic analysis and flow tracing. Specifically, the flow untraceability objective studied in this paper is to hide the linear correlations of incoming and outgoing GEVs (or CDVs) for each flow at each intermediate node, i.e., each newly generated outgoing GEV (or CDV) is linearly dependent with multiple incoming GEVs (or CDVs) from other sources.

To measure the anonymity, we define the *anonymity level* for each outgoing GEV (or CDV) as the number of incoming GEVs (or CDVs) linearly correlated with the outgoing GEV (or CDV), i.e., the number of incoming GEVs (or CDVs) should be traced back from one outgoing GEV (or CDV). In the following sections, we will show that the increased number of incoming GEVs (or CDVs) linearly correlated with the outgoing GEV (or CDV) not only increases the computational complexity of traffic analysis attack, but also decreases the probability that the attacker traces the flow back (forth) to the multiple sources (destinations), and consequently the attacker cannot identify the real source (destination).

To facilitate further discussions, we summarize important notations in the paper for ease of reference in Table I.

## 3 *ALNCode*: A Novel Anonymous Linear Network Coding against Traffic Analysis Attacks

We now present our Anonymous Linear Network Coding (*ALNCode*) mechanism to provide flow untraceability. We first give theoretical analysis to acquire the sufficient and necessary condition that the correlation of packets in a flow can be obfuscated. Then, we prove that, if the sufficient and necessary condition is satisfied, we can design deterministic LNC such that the outgoing coded packets of a flow are guaranteed to be correlated to incoming coded packets in other flows. With the same condition, we also prove the probability that a randomly generated coded packet is correlated to coded packets in other flows. These theoretical results show that the proposed *ALNCode* can achieve anonymous communication even without encrypting the packet.

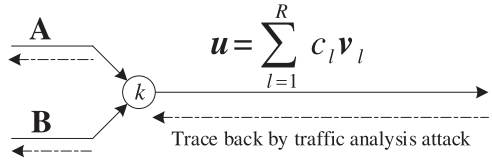



Fig. 2. Generate an obfuscated vector $\boldsymbol{u}$ for vector set $\mathbf{A}$ at node $k$, where $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_R\}$ is a maximum independent set of $\mathbf{A} \cup \mathbf{B}$, $\{\boldsymbol{v}_g, \boldsymbol{v}_{g+1}, \ldots, \boldsymbol{v}_R\} \subseteq \mathbf{B}$, $1 \le g \le R$, $\boldsymbol{u} = \sum_{l=1}^{R} c_l \boldsymbol{v}_l$ and $[c_g, c_{g+1}, \ldots, c_R]$ is a nonzero vector.

### 3.1 Theoretical Analysis

In this subsection, we first prove the sufficient and necessary condition that outgoing coded packets of a flow can be generated to be correlated to incoming coded packets in other flows, which can obfuscate the correlation of packets in a flow. We then prove the probability that such condition is satisfied. We also conduct theoretical analysis and extensive numerical experiments to understand the impacts of LNC parameters, such as the finite field of LNC, the number of original data blocks, etc.

#### 3.1.1 Definitions

The key idea in *ALNCode* is to produce *obfuscated* GEVs (or CDVs) at intermediate nodes, which are linearly correlated not only with received GEVs (or CDVs) from the same flow, but also those from other flows. Before we give the basic idea of *ALNCode* scheme, we first show the definition of the *obfuscated vector*. Suppose that $\mathbf{A}$ and $\mathbf{B}$ are two sets of vectors, we have the following definition.

**Definition 1.** *$\boldsymbol{u}$ is an* obfuscated vector *for vector space $L(\mathbf{A})$, w.r.t $\mathbf{B}$, iff $\boldsymbol{u} \in L(\mathbf{A})$ and there exists a maximal linearly independent set of $\mathbf{A} \cup \mathbf{B}$, denoted as $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_R\}$, in which $\{\boldsymbol{v}_g, \boldsymbol{v}_{g+1}, \ldots, \boldsymbol{v}_R\} \subseteq \mathbf{B}$, $1 \le g \le R$, $\boldsymbol{u} = \sum_{l=1}^{R} c_l \boldsymbol{v}_l$ and $[c_g, c_{g+1}, \ldots, c_R]$ is a nonzero vector.*

From the above definition, the obfuscated vector $\boldsymbol{u}$ of vector space $L(\mathbf{A})$ not only can be generated by the vectors in $\mathbf{A}$, but also has linear correlations with linearly independent vectors in set $\mathbf{B}$. Therefore, if an attacker analyzes the linear correlations between obfuscated vector $\boldsymbol{u}$ of vector space $L(\mathbf{A})$ and the set of vectors $\mathbf{A} \cup \mathbf{B}$, it cannot tell which set of vectors (i.e., $\mathbf{A}$) are used to generate vector $\boldsymbol{u}$. Fig. 2 shows the idea.

Accordingly, suppose that $\mathbf{A}$ is the set of GEVs (or CDVs) received at intermediate node $k$ from generation $j$ of flow $i$ and $\mathbf{B}$ is the set of GEVs (or CDVs) received at $k$ from flows other than $i$, we have the formal definition of the *obfuscated GEV (or CDV)*.

**Definition 2.** *$\boldsymbol{u}_{i,j}$ is an* obfuscated GEV (or CDV) *for generation $j$ of flow $i$, iff $\boldsymbol{u}_{i,j} \in L(\mathbf{A})$ and there exists a maximal linearly independent set of $\mathbf{A} \cup \mathbf{B}$, denoted as $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_R\}$, in which $\{\boldsymbol{v}_g, \boldsymbol{v}_{g+1}, \ldots, \boldsymbol{v}_R\} \subseteq \mathbf{B}$, $1 \leq g \leq R$, $\boldsymbol{u}_{i,j} = \sum_{l=1}^{R} c_l \mathbf{v}_l$ and $[c_g, c_{g+1}, \ldots, c_R]$ is a nonzero vector.*

In the above definition, $\boldsymbol{u}_{i,j} \in L(\mathbf{A})$ means the new outgoing GEV (or CDV) $\boldsymbol{u}_{i,j}$ can be generated by incoming GEVs (or CDVs) from the same generation and flow, which is the requirement of network coding. Moreover, the second part condition means the generated GEV (or CDV) $\boldsymbol{u}_{i,j}$ also has linear correlations with GEVs (or CDVs) received from other flows, which is the requirement of confusing the GEV (or CDV) analysis attacker. In this way, if an attacker attempts to trace back the source of the coded packet with GEV (or CDV) $\boldsymbol{u}_{i,j}$, it would fail to identify which flow the packet actually belongs to.

### 3.1.2 The Existence of Obfuscated Vector

We next prove the sufficient and necessary condition that an obfuscated vector does exist.

**Theorem 1.** *Given two sets of vectors $\mathbf{A}$ and $\mathbf{B}$, an obfuscated vector $\boldsymbol{u}$ exists for $L(\mathbf{A})$, iff $dim(L(\mathbf{A}) \cap L(\mathbf{B})) \neq 0$.*

**Proof.** The proof is shown in *Appendix* A., which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TNSE.2018.2888848 □

### 3.1.3 The Intersection Probability

In this subsection, we will prove the probability that the sufficient and necessary condition is satisfied.

In practice, the contents of most unicast flows are randomized at the source nodes by using compression and encryption schemes defined in protocols, such as *Hypertext Transfer Protocol Secure* (HTTPS) and *Secure Shell* (SSH). For example, a recent study by Google[2] shows that the percentage of HTTPS traffic is increasing significantly in the past three years and now more than 60 percent Chrome traffic are using HTTPS.

When LNC is used, the source node usually chooses every coefficient in GEV randomly from a finite field $\mathbb{F}_q$. Consequently, the coded vectors (i.e., GEVs and CDVs) sent from the source node of a flow can be considered as a sequence of vectors that are randomly and independently selected from vector space $\mathbb{F}_q^{\mathfrak{t}}$, where $\mathfrak{t}$ is the number of elements in each vector and $\mathfrak{t} = h$ for GEV and $\mathfrak{t} = h + H$ for CDV, respectively. Similarly, when an intermediate node $k$ receives some vectors in a generation of a flow, it usually generates outgoing vectors by randomly choosing coefficients in $\mathbb{F}_q$ to linearly combine incoming vectors. So the outgoing vectors from node $k$ can be viewed as a sequence of vectors that are randomly and independently selected from the linear span of incoming vectors.

Based on these analyses, we now consider all the incoming vectors from generation $j$ of flow $i$ received by node $k$. Let the number of these incoming vectors be $f_1$. Since these incoming vectors are coming from one or more other nodes and the vectors generated by each of them may be selected in different linear spans, the linear span of these incoming vectors can be very complicated. To address this issue, we will consider two cases. In the first case, to simplify the analysis, we assume that node $k$ receives $f_1$ incoming vectors that are randomly and independently selected from vector space $\mathbb{F}_q^{\mathfrak{t}}$. In the second case, we can assume that node $k$ can find the linear span of $f_1$ incoming vectors and we let the dimension of the linear span be $r$.

Next, we consider the vectors received by node $k$ from all other flows. Let the number of these incoming vectors be $f_2$. Since these vectors belong to different flows, we will assume that node $k$ receives $f_2$ vectors that are randomly and independently selected from vector space $\mathbb{F}_q^{\mathfrak{t}}$.

We now define the *intersection probability* as the probability that the linear span of vectors in one flow has non-empty intersection with the linear span of vectors from all other flows. In the first case for incoming vectors in one flow, we state and prove Lemma 1 to show the probability that the dimension of the linear span of all incoming vectors equals to $r$. We then develop Theorem 2 for the lower bound of the intersection probability. For the second case, we state and prove Lemma 2 to show the lower bound of the intersection probability.

Let $\mathbf{A}$ be the set of incoming vectors from generation $j$ of flow $i$ received by node $k$ and $\mathbf{B}$ be the set of incoming vectors from all other flows received by node $k$. To simplify the notations, we also let $L_1$ and $L_2$ be the linear spans for $\mathbf{A}$ and $\mathbf{B}$, respectively. Let $\begin{bmatrix} m \\ r \end{bmatrix}_q$ be the Gaussian binomial coefficient, *i.e.*, $\begin{bmatrix} m \\ r \end{bmatrix}_q = \frac{(q^m-1)(q^{m-1}-1)\cdots(q^{m-r+1}-1)}{(q-1)(q^2-1)\cdots(q^r-1)}, 0 < r \leq m$. We set $\begin{bmatrix} m \\ 0 \end{bmatrix}_q = 1, \forall m > 0$. We first prove the following lemma.

**Lemma 1.** *For any $m \times n$ dimensional matrix $\mathbf{Y}$ whose elements are randomly selected from finite field $\mathbb{F}_q$, the probability that $\mathrm{rank}(\mathbf{Y}) = r, 0 \leq r \leq min(m, n)$ is given by:*

$$p_1(m, n, r, q) = \begin{bmatrix} m \\ r \end{bmatrix}_q \prod_{ql=n-r+1}^{n} (q^l - 1) q^{\frac{r(r-1)}{2} - mn}.$$

**Proof.** The proof is shown in *Appendix* B, available in the online supplemental material. □

From Lemma 1, we have $p_1(m, n, r, q) = p_1(n, m, r, q)$. Next, we develop a lower bound of intersection probability in the second case for incoming vectors in one flow.

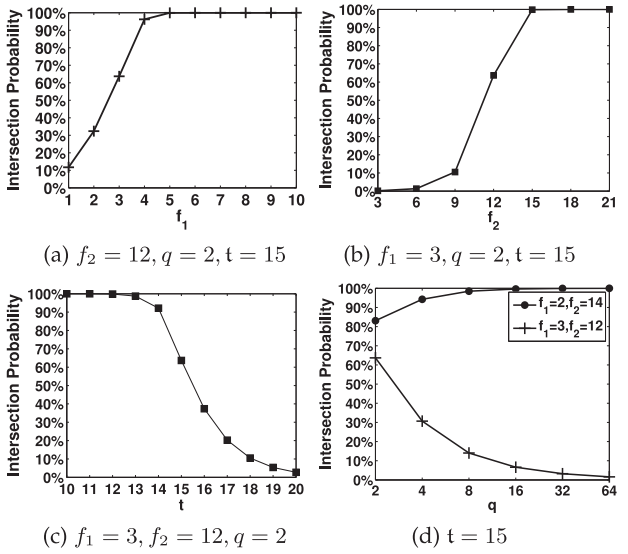
**Lemma 2.** *Given a vector space $L_1$ with $dim(L_1) = r(r \geq 0)$, if $L_2$ is a vector space spanned by $f_2$ vectors randomly selected from $\mathbb{F}_q^{\mathfrak{t}}$ in order, the probability that $dim(L_1 \cap L_2) \neq 0$ is:*

$$p_2(r, f_2, \mathfrak{t}, q) \geq 1 - \sum_{g=0}^{\min(f_2, \mathfrak{t}-r)} p_1(f_2, \mathfrak{t} - r, g, q) q^{(g-f_2)r}.$$

**Proof.** The proof is shown in *Appendix* C, available in the online supplemental material. □

Now we prove the lower bound of the intersection probability in the first case for incoming vectors in one flow.

2. https://www.zdnet.com/article/google-this-surge-in-chrome-https-traffic-shows-how-much-safer-you-now-are-online/




(a) $f_2 = 12, q = 2, \mathrm{t} = 15$ (b) $f_1 = 3, q = 2, \mathrm{t} = 15$

(c) $f_1 = 3, f_2 = 12, q = 2$ (d) $\mathrm{t} = 15$

Fig. 3. Lower bound of intersection probability under different influential parameters.

**Theorem 2.** *Suppose that $f_1$ and $f_2$ received vectors are randomly selected from $\mathbb{F}_q^{\mathrm{t}}$, respectively, for any $\mathrm{t}, q, f_1, f_2 \geq 0$, the probability $dim(L_1 \cap L_2) \neq 0$ satisfies:*

$$P(\dim(L_1 \cap L_2) \neq 0) \geq \sum_{r=0}^{\min(f_1,\mathrm{t})} \left( \begin{bmatrix} f_1 \\ r \end{bmatrix}_q \prod_{ql=\mathrm{t}-r+1}^{\mathrm{t}} (q^l - 1) q^{\frac{r(r-1)}{2} - f_1 \mathrm{t}} \right) \times \left( 1 - \sum_{g=0}^{\min(f_2,\mathrm{t}-r)} \left( \begin{bmatrix} f_2 \\ g \end{bmatrix}_q \prod_{ql=\mathrm{t}-r-g+1}^{\mathrm{t}-r} (q^l - 1) q^{\frac{g(g-1)}{2} - f_2 \mathrm{t} + gr} \right) \right).$$

**Proof.** The proof is shown in *Appendix* D, available in the online supplemental material. □

It shall be noted that, for any distribution of the received vectors, an intermediate node can always generate new coded vectors according to the proposed ALNCode (Algorithm 1) or random linear network coding with appropriate parameters (Theorem 4), because both the schemes are not based on the assumption of random distribution of received vectors.

### 3.1.4 The Influential Parameters

We next conduct theoretical analysis and extensive numerical experiments to understand the impacts of LNC parameters, such as the finite field of LNC, the number of original data blocks, etc.

To provide a better idea of the intersection probability with its deciding parameters, we show the lower bound derived in Theorem 2 at different values of $f_1, f_2, \mathrm{t}$, and $q$. Figs. 3a and 3b show that the probability increases with the increase of $f_1$ and $f_2$, respectively. The reason is straightforward: when $\mathrm{t}$ and $q$ are fixed, the more coded packets a node receives in the current flow and in other flows, the larger probability that the two vector spaces $L_1$ and $L_2$ have nonzero intersection.

**Corollary 1.** *Suppose that $\mathbf{Y}$ is an $m \times \mathrm{t}$ dimensional matrix in which the elements are randomly selected from $\mathbb{F}_q$, the probability that $\mathbf{Y}$ is full rank is:*

$$\begin{cases} \prod_{i=\mathrm{t}-m+1}^{\mathrm{t}} (1 - \frac{1}{q^i}), & \text{when } m \leq \mathrm{t}; \\ \prod_{i=m-\mathrm{t}+1}^{m} (1 - \frac{1}{q^i}), & \text{when } m > \mathrm{t}. \end{cases}$$

**Proof.** From Lemma 1, it holds obviously. □

From Corollary 1, we can conclude that the full rank probability of $m \times \mathrm{t}$-dimensional matrix $\mathbf{Y}$ increases as the increase of $q$. Moreover, the full rank probability also increases as the increase of $\mathrm{t}$ when $m \leq \mathrm{t}$, and increases as the decrease of $\mathrm{t}$ when $m > \mathrm{t}$.

An $m \times \mathrm{t}$ dimensional matrix can be formed by the received $m$ vectors as its rows, i.e., the $i$th row vector of the matrix is the $i$th received vector. Therefore, the $m \times \mathrm{t}$ dimensional matrix, in which each element is randomly selected from $\mathbb{F}_q$, and the $m$ vectors randomly selected from $\mathbb{F}_q^{\mathrm{t}}$ has one to one correspondence.

Fig. 3c shows that the probability decreases with the increase of $\mathrm{t}$, while Fig. 3d demonstrates different trends with the increase of $q$ in different cases. In particular, for $\forall f_1, f_2 > 0$, we show below with analysis that: when $f_1 + f_2 \leq \mathrm{t}$, this lower bound probability decreases with the increase of $q$ and $\mathrm{t}$; otherwise, it increases with the increase of $q$ and decreases with the increase of $\mathrm{t}$.

When $f_1 + f_2 \leq \mathrm{t}$, from Corollary 1, the probability that the $(f_1 + f_2) \times \mathrm{t}$ dimensional matrix formed by the received vectors (i.e., GEVs or CDVs) as its rows has full rank, is higher with larger $q$ and $\mathrm{t}$. When the $(f_1 + f_2) \times \mathrm{t}$ dimensional matrix has full rank, the $f_1 + f_2$ vectors are linearly independent (since $f_1 + f_2 \leq \mathrm{t}$), i.e., $\dim(L_1 \cap L_2) = 0$. Thus, the intersection probability decreases with the increase of $q$ and $\mathrm{t}$. Similar results are presented in [17], [18], which show the probability that the received GEVs are independent grows to 1, when $q$ and $\mathrm{t}$ grow to infinity if the number of received coded packets are no more than $\mathrm{t}$.

When $f_1 + f_2 > \mathrm{t}$, let $\overline{\mathbf{V}}_3 = \begin{bmatrix} \overline{\mathbf{A}} \\ \overline{\mathbf{B}} \end{bmatrix}$. The analysis is shown below:

Given two groups of vectors $\mathbf{A}$ and $\mathbf{B}$ over a vector space $\mathbb{F}_q^{\mathrm{t}}$, $L_1$ and $L_2$ are subspaces of $\mathbb{F}_q^{\mathrm{t}}$. Then both their intersection $L_1 \cap L_2$ and their sum $L_1 + L_2$ are also subspaces of $\mathbb{F}_q^{\mathrm{t}}$. We have [19]:

$$\dim(L_1) + \dim(L_2) = \dim(L_1 + L_2) + \dim(L_1 \cap L_2), \quad (4)$$

where $\dim(L_1 + L_2) = \dim(L(\mathbf{A} \cup \mathbf{B}))$.

Given a set of vectors $\mathbf{V}$, let $\overline{\mathbf{V}}$ be the matrix formed by nonzero vectors in $\mathbf{V}$ as its rows. We have:

$$\dim(L_1) = \mathrm{rank}(\overline{\mathbf{A}}), \dim(L_2) = \mathrm{rank}(\overline{\mathbf{B}}) \text{ and } \dim(L(\mathbf{A} \cup \mathbf{B})) = \mathrm{rank}\left( \begin{bmatrix} \overline{\mathbf{A}} \\ \overline{\mathbf{B}} \end{bmatrix} \right).$$

Therefore, we have:

$$\begin{aligned} \dim(L_1 \cap L_2) &= \dim(L_1) + \dim(L_2) - \dim(L(\mathbf{A} \cup \mathbf{B})) \\ &= \mathrm{rank}(\overline{\mathbf{A}}) + \mathrm{rank}(\overline{\mathbf{B}}) - \mathrm{rank}(\overline{\mathbf{V}}_3). \end{aligned}$$

When the $f_1$ vectors received from generation $j$ of flow $i$ are linearly independent, and the $f_2$ vectors received from other flows are linearly independent, then $\mathrm{rank}(\overline{\mathbf{A}}) = \min(f_1, \mathrm{t})$, $\mathrm{rank}(\overline{\mathbf{B}}) = \min(f_2, \mathrm{t})$, and $\mathrm{rank}(\overline{\mathbf{V}}_3) \leq \mathrm{t}$. In this case, we have

$$\begin{aligned}\dim(L_1 \cap L_2) &= \text{rank}(\overline{\mathbf{A}}) + \text{rank}(\overline{\mathbf{B}}) - \text{rank}(\overline{\mathbf{V}}_3)\\ &\geq \min(f_1, \mathfrak{t}) + \min(f_2, \mathfrak{t}) - \mathfrak{t}\\ &> 0. \text{ (since } f_1 + f_2 > \mathfrak{t} \text{ and } f_1, f_2 > 0)\end{aligned}$$

Therefore, if the probability that $\mathbf{A}$ and $\mathbf{B}$ have full ranks increases, the probability that $\dim(L_1 \cap L_2) > 0$ increases. From Corollary 1, the former probabilities increase with the increase of $q$ and the decrease of $\mathfrak{t}$.

The above results shows that obfuscated vectors (GEVs or CDVs) exist with high probability when appropriate coding parameters are selected. Moreover, it can guide the practical selection of field size ($q$), the number of data blocks per generation ($h$), and the number of packets to buffer before recoding ($f_1$) received coded packets, given the routes of flows decided by the routing protocol (which determines $f_2$). In general, an intermediate node may buffer sufficient number of linearly independent coded packets in generation $j$ of flow $i$ to produce different newly coded packets. When dividing a flow into generations, a reasonably small $h$ should be chosen to guarantee a good intersection probability, as well as low decoding complexity. The finite field size $q$ can then be set accordingly: if many coded packets can be received at each node such that $f_1 + f_2 > \mathfrak{t}$, a relatively large $q$ can be used, but not too large considering the communication overhead (the ratio of GEV length and packet length) and decoding complexity (Gaussian elimination); if few coded packets can be received, we can simply select $q = 2$ for the best intersection probability.

### 3.2 The Basic Idea

In this subsection, we show the basic ideas to generate obfuscated vectors for vector space $L_1$ when $\dim(L_1 \cap L_2) \neq 0$.

Suppose $\mathbf{N}_{i,j,k} = \{\mathbf{n}_1, \ldots, \mathbf{n}_N\}$ denotes the basis of vector space $L_1 \cap L_2$, $\mathbf{N}_{i,j,k}$ can be extended to the basis of vector space $L_1$ (with methods described in Section 3.3), i.e., letting $r_1 = \dim(L_1)$, there exist $r_1 - N$ vectors, $\{\boldsymbol{\alpha}_{l_1}, \ldots, \boldsymbol{\alpha}_{l_{r_1-N}}\}$, in $A$, such that $\boldsymbol{\Theta}_{i,j,k} = \{\mathbf{n}_1, \ldots, \mathbf{n}_N, \boldsymbol{\alpha}_{l_1}, \ldots, \boldsymbol{\alpha}_{l_{r_1-N}}\}$ forms the basis of $L_1$. $\boldsymbol{\Theta}_{i,j,k}$ is referred to as the *obfuscated basis* of $L_1$. Let $\overline{\boldsymbol{\Theta}}_{i,j,k}$ be the matrix formed by vectors in $\boldsymbol{\Theta}_{i,j,k}$ as its rows, and $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_{r_1}]$ be a vector in $\mathbb{F}_q^{r_1}$. Set $\mathbf{v}_{i,j} = \boldsymbol{\rho}\overline{\boldsymbol{\Theta}}_{i,j,k}$. Next, we prove the sufficient condition under which $\mathbf{v}_{i,j}$ produced above *is* an obfuscated vector.

**Theorem 3.** *When $dim(L_1 \cap L_2) \neq 0$, the vector $\mathbf{v}_{i,j}$ is an obfuscated vector, if not all the first $N$ elements of $\boldsymbol{\rho}$ are zero.*

**Proof.** The proof is shown in *Appendix* E, available in the online supplemental material. □

### 3.3 A Deterministic Linear Network Coding Scheme

Based on Definition 2, Theorems 1 and 3, we now design a detailed LNC scheme, by which $r_1$ new coded packets with linearly independent obfuscated vectors (i.e., GEVs or CDVs) can be generated at each intermediate node $k$, after it receives $r_1$ linearly independent vectors (i.e., GEVs or CDVs) in the past $T$ time slots from generation $j$ of flow $i$, as long as $\dim(L_1 \cap L_2) \neq 0$ is satisfied.

According to the basic ideas illustrated in the previous section, we first summarize the steps to obtain the obfuscated $r_1$ new coded packets with linearly independent obfuscated vectors as follows:

- ▷ Obtain $L_1 \cap L_2$ from the received vectors at node $k$ from generation $j$ of flow $i$ and other flows.
- ▷ Derive $\mathbf{N}_{i,j,k}$, i.e., the basis of $L_1 \cap L_2$, and extend it to $\boldsymbol{\Theta}_{i,j,k}$, i.e., the basis of vector space $L_1$.
- ▷ Select $r_1$ linearly independent vectors $\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_{r_1}$, in which $\boldsymbol{\rho}_m \in \mathbb{F}_q^{r_1}$ with the first $N$ elements not all zero, $m \in \{1, \ldots, r_1\}$.
- ▷ Generate the $m$th obfuscated vector for generation $j$ of flow $i$: $\mathbf{v}_m = \boldsymbol{\rho}_m\overline{\boldsymbol{\Theta}}_{i,j,k}$.

We next detail these procedures, as well as how local encoding vectors are formed at $k$ to generate these new vectors from received vectors.

*1) Derive $\boldsymbol{\Theta}_{i,j,k}$.* Let $\boldsymbol{\Lambda} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{r_1}\}$ be the maximal linearly independent set of $\mathbf{A}$. $\boldsymbol{\Lambda}$ is the basis of $L_1$. Let $\overline{\boldsymbol{\Lambda}}$ be the matrix formed by vectors in $\boldsymbol{\Lambda}$ as its rows. Let $\boldsymbol{\Gamma} = \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{r_2}\}$ be the maximal linearly independent set of $\mathbf{B}$. $\boldsymbol{\Gamma}$ is the basis of $L_2$. We compute the basis of $L_1 \cap L_2$ and extend it to the basis of $L_1$ following a general method [20]:

i) Construct a matrix $\mathbf{X} = [\boldsymbol{\alpha}_1^T, \ldots, \boldsymbol{\alpha}_{r_1}^T, \boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_{r_2}^T]$ and then reduce $\mathbf{X}$ to its reduced row-echelon form $rref(\mathbf{X})$ by Gaussian elimination. Note that if a row of $rref(\mathbf{X})$ is nonzero, the first nonzero element of this row is refereed to as the *pivot* of the row. A *non-pivotal column* refers to a column no elements of which is a pivot.

ii) Let $N'$ be the number of non-pivotal columns of $rref(\mathbf{X})$. Then $N' = N = \dim(L_1 \cap L_2)$ [20]. We can obtain $N$ linear combinations $\sum_{l=1}^{r_1} a_{n,l}\boldsymbol{\alpha}_l, 1 \leq n \leq N$, where $a_{n,i}$ is the $i_{th}$ element of the $n_{th}$ non-pivotal column of $rref(\mathbf{X})$. These linear combinations form the basis of $L_1 \cap L_2$, i.e., $\mathbf{N}_{i,j,k}$.

iii) To derive $\boldsymbol{\Theta}_{i,j,k}$, a basis of $L_1$ which contains $\mathbf{N}_{i,j,k}$, we first construct a matrix

$$\boldsymbol{\Phi} = \left[\sum_{l=1}^{r_1} a_{1,l}\boldsymbol{\alpha}_l^T, \quad \ldots, \quad \sum_{l=1}^{r_1} a_{N,l}\boldsymbol{\alpha}_l^T, \quad \boldsymbol{\alpha}_1^T, \quad \ldots, \quad \boldsymbol{\alpha}_{r_1}^T\right].$$

We know the basis of the column space of $\boldsymbol{\Phi}$ can be derived as follows: reduce $\boldsymbol{\Phi}$ to its reduced row-echelon form $rref(\boldsymbol{\Phi})$, and then those column vectors in $\boldsymbol{\Phi}$, that correspond to the columns in $rref(\boldsymbol{\Phi})$ containing pivots, form the basis. The column space of $\boldsymbol{\Phi}$ is indeed $L(\mathbf{N}_{i,j,k} \cup \mathbf{A}) = L_1$, and thus we have derived a basis of $L_1$. In addition, since the set of vectors in $\mathbf{N}_{i,j,k}$ are linearly independent, all the column vectors in the $\mathbf{N}_{i,j,k}$ part of $\boldsymbol{\Phi}$ correspond to columns in $rref(\boldsymbol{\Phi})$ containing pivots. Thus, the basis of $L_1$ derived above is composed of all the vectors in $\mathbf{N}_{i,j,k}$, as well as $r_1 - N$ other vectors in $\mathbf{A}$, which we denote as $\{\boldsymbol{\alpha}_{L_1}, \ldots, \boldsymbol{\alpha}_{l_{r_1-N}}\}$. $\boldsymbol{\Theta}_{i,j,k}$, the basis of $L_1$ which contains $\mathbf{N}_{i,j,k}$, is thus derived as

$$\left\{\sum_{l=1}^{r_1} a_{1,l}\boldsymbol{\alpha}_l, \ldots, \sum_{l=1}^{r_1} a_{N,l}\boldsymbol{\alpha}_l, \boldsymbol{\alpha}_{l_1}, \ldots, \boldsymbol{\alpha}_{l_{r_1-N}}\right\}. \quad (5)$$

*2) Generate $r_1$ linearly independent obfuscated vectors.* The vectors in $\boldsymbol{\Theta}_{i,j,k}$ form the basis of $L_1$ and the first $N$ vectors in $\boldsymbol{\Theta}_{i,j,k}$ are the basis of $L_1 \cap L_2$. In general, to produce $r_1$ linearly independent obfuscated vectors, we left-multiply $\boldsymbol{\Theta}_{i,j,k}$ by a nonsingular matrix composed by $r_1$ linearly

independent vectors from $\mathbb{F}_q^{r_1}$, the first $N$ elements of each of which are not all zeros. We can select a nonsingular lower triangular matrix $\mathbf{C}_1$ as follows:

$$\mathbf{C}_1 = \begin{bmatrix} c_{1,1} & & & \\ c_{2,1} & c_{2,2} & 0 & \\ \vdots & \vdots & \ddots & \\ c_{r_1,1} & c_{r_1,2} & \cdots & c_{r_1,r_1} \end{bmatrix},$$

where each $c_{i',j'}, 1 \le j' \le i' \le r_1$ is randomly selected from $\{1 \cdots q-1\}$. Since $\dim(L_1 \cap L_2) \neq 0$ and the leading element of each row of $\mathbf{C}_1$ is nonzero, each row vector of $\mathbf{C}_1\overline{\mathbf{\Theta}}_{i,j,k}$ is an obfuscated vector; since matrix $\mathbf{C}_1$ has full rank, these $r_1$ obfuscated vectors are linearly independent.

*3) Construct local encoding vectors*

Recall the intermediate node encoding model described in Section 2.1: after receiving coded packets corresponding to $r_1$ linearly independent GEVs (or CDVs) $\mathbf{\Lambda} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{r_1}\}$, node $k$ selects $r_1$ coding coefficients from $\mathbb{F}_q^{r_1}$. Then according to these local encoding vectors, it does linear combinations of the $r_1$ received coded packets to produce $r_1$ coded packets with $r_1$ obfuscated GEVs (or CDVs). We can derive the local encoding vectors as follows.

Let $\mathbf{\Omega}$ denote the $r_1 \times r_1$ dimensional local encoding matrix, whose rows are the local encoding vectors. It should satisfy $\mathbf{\Omega}\overline{\mathbf{\Lambda}} = \mathbf{C}_1\overline{\mathbf{\Theta}}_{i,j,k}$. Since the matrix $\overline{\mathbf{\Theta}}_{i,j,k}$ is formed by the obfuscate basis as its rows, it can be represented as $\overline{\mathbf{\Theta}}_{i,j,k} = \mathbf{C}_2\overline{\mathbf{\Lambda}}$, where $\mathbf{C}_2$ is a $r_1 \times r_1$ dimensional matrix as follows:

$$\mathbf{C}_2 = \begin{bmatrix} a_{1,1} & \cdots & a_{1,r_1} \\ \vdots & \vdots & \vdots \\ a_{N,1} & \cdots & a_{N,r_1} \\ & \mathbf{I}_{r_1,l_1} & \\ & \vdots & \\ & \mathbf{I}_{r_1,l_{r_1-N}} & \end{bmatrix} \quad \text{(according to Eq. (5))},$$

where $\mathbf{I}_{r_1,l_n}$ is the $l_n$th row of a $r_1 \times r_1$ identity matrix.

Since $\mathbf{\Omega}\overline{\mathbf{\Lambda}} = \mathbf{C}_1\overline{\mathbf{\Theta}}_{i,j,k} = \mathbf{C}_1\mathbf{C}_2\overline{\mathbf{\Lambda}}$, we derive $\mathbf{\Omega} = \mathbf{C}_1\mathbf{C}_2$, i.e., each row of $\mathbf{\Omega}$ is a local encoding vector, which node $k$ should use to generate $r_1$ independent obfuscated GEVs (or CDVs) according to the intermediate node encoding model described in Section 2.1.

Algorithm 1 shows how to calculate the local encoding matrix. In this algorithm, there are a few important features: (1) when $\dim(L_1 \cap L_2) > 0$, the proposed algorithm gives the local encoding vector to generate each outgoing vector to make sure it is obfuscated, and (2) if an intermediate node receives $r_1$ linearly independent incoming packets from a flow, the proposed Algorithm 1 can generate $r_1$ linearly independent local encoding vectors for it. With these elegant designs, the proposed algorithm can guarantee that, when $\dim(L_1 \cap L_2) > 0$, if an intermediate node receives $r_1$ linearly independent incoming packets in a flow, it can generate $r_1$ linearly independent outgoing coded packets and all of them are obfuscated.

Next, we will show the computational complexity of generating $r_1$ linearly independent coded packets with obfuscated GEVs (or CDVs). Suppose the length of each vector in sets $\mathbf{A}$ and $\mathbf{B}$ is $\mathfrak{t}$. The total computational complexity is shown as follows: Let $f_1 = |\mathbf{A}|$ and $f_2 = |\mathbf{B}|$. Since the computational complexity of Gaussian elimination applied to an $m \times n$ dimensional matrix is $O(mn\min(m,n))$ and that of matrix multiplication between an $m \times n$ dimensional matrix and a $n \times l$ dimensional matrix is $O(mnl)$, the computational complexity of each line in Algorithm 1 can be derived as follows ( $N \le r_1 \le h$) :

1) line 1: $O(\mathfrak{t}f_1\min(\mathfrak{t}, f_1) + \mathfrak{t}f_2\min(\mathfrak{t}, f_2)) = O(\mathfrak{t}^2(f_1+f_2))$.
2) line 3: $O(\mathfrak{t}(r_1+r_2)\min(\mathfrak{t}, r_1+r_2)) = O(\mathfrak{t}^2(r_1+r_2))$.
3) line 4-6, 9-11: $O(Nr_1\mathfrak{t}) = O(hr_1\mathfrak{t})$.
4) line 8: $O(\mathfrak{t}(N+r_1)\min(\mathfrak{t}, N+r_1)) = O(\mathfrak{t}^2r_1)$.
5) line 13-16: $O(r_1^3) = O(h^2r_1)$.

**Algorithm 1.** Local Encoding Matrix Computing in *ALNCode*

1: Find the maximal linearly independent set of $\mathbf{A}$ and $\mathbf{B}$ by Gaussian elimination which are $\{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{r_1}\}$ and $\{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{r_2}\}$ respectively.
2: Construct a matrix $\mathbf{X} = [\boldsymbol{\alpha}_1^T, \ldots, \boldsymbol{\alpha}_{r_1}^T, \boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_{r_2}^T]$.
3: Compute reduced row-echelon form matrix $rref(\mathbf{X})$ by Gaussian elimination. Let the number of non-pivotal columns of $rref(\mathbf{X})$ be $N$.
4: **for** $n$ from 1 to $N$ **do**
5: $\boldsymbol{\theta}_n = \sum_{l=1}^{r_1} a_{n,l}\boldsymbol{\alpha}_l$ where $a_{n,i}$ is the $i_{th}$ element of the $n_{th}$ non-pivotal column of $rref(\mathbf{X})$. The $n_{th}$ row of $\mathbf{C}_2$ is set to $[a_{n,1}, \ldots, a_{n,r_1}]$.
6: **end for**
7: $\mathbf{N}_{i,j,k} = \bigcup_{n=1}^{N}\boldsymbol{\theta}_n$.
8: Find $r_1 - N$ vectors in $\mathbf{A}$, $\{\boldsymbol{\alpha}_{l_1}, \ldots, \boldsymbol{\alpha}_{l_{r_1-N}}\}$, such that GEVs in set $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N, \boldsymbol{\alpha}_{l_1}, \ldots, \boldsymbol{\alpha}_{l_{r_1-N}}\}$ are linearly independent.
9: **for** $n$ from 1 to $r_1 - N$ **do**
10: $\boldsymbol{\theta}_{N+n} = \boldsymbol{\alpha}_{l_n}$. The $(N+n)_{th}$ row of $\mathbf{C}_2$ is set to $\mathbf{I}_{r_1,l_n}$.
11: **end for**
12: The *obfuscated basis* of $L_1$ is $\mathbf{\Theta}_{i,j,k} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{r_1}\}$.
13: **for** $l$ from 1 to $r_1$ **do**
14: select $l$ numbers from $\{1, \ldots, q-1\}$ as the first $l$ elements of the $l_{th}$ row of matrix $\mathbf{C}_1$. The remaining $r_1 - l$ elements are set to 0.
15: **end for**
16: **return** local encoding matrix $\mathbf{\Omega} = \mathbf{C}_1\mathbf{C}_2$.

Since $r_1 \le f_1, r_2 \le f_2$ and $h \le \mathfrak{t}$, the total computational complexity to compute local encoding matrix $\mathbf{\Omega}$ is $O(\mathfrak{t}^2(f_1+f_2))$. To further calculate $r_1$ new coded packets, the total computational complexity is $O(\mathfrak{t}^2(f_1+f_2) + r_1^2H)$. To generate obfuscated GEVs, $\mathfrak{t} = h$; while to generate obfuscated CDVs, $\mathfrak{t} = h + H$. Note that to generate $r_1$ new coded packets without considering anonymity, the computational complexity of network coding is $O(h^2f_1 + r_1^2H)$.

In both expressions, the first part in the computational complexity (i.e., $O(h^2f_1)$ for traditional LNC and $O(\mathfrak{t}^2(f_1+f_2))$ for ALNCode) is introduced by Gaussian eliminations. The second part $O(r_1^2H)$ is introduced by matrix multiplication. By comparing the complexity of traditional LNC and ALNCode, we note that the ALNCode needs more computation in the first part but the asymptotic behavior could be the same as that of the traditional NC. Moreover, the computational complexity can be further reduced because both the Gaussian elimination and the matrix multiplication can be implemented by using parallel

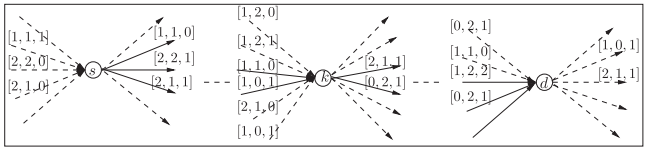



Fig. 4. *ALNCode* at different nodes.

computing or hardware on today's routers [21]. Therefore, we believe that the proposed ALNCode scheme can be efficiently implemented.

*At the Source and Destination*. Our previous discussions have been focusing on recoding at intermediate nodes to hide relationship of its incoming and outgoing packets. Since the source or destination node of one flow can be an intermediate node for other flows, we next show that, with a similar scheme, the source and destination of a flow can also hide themselves, as long as there are other flows going through them.

For the source node $s$ of flow $i$, if $s$ also receives other flows, it can also try to produce obfuscated vectors (i.e., GEVs or CDVs) for generation $j$ of flow $i$, which is linearly correlated with other flows. Generally, if $\dim(L_2) = r_2$, the source can generate $r_2$ linearly independent vectors from $L_2$, all of which are obfuscated vectors for generation $j$ of flow $i$. By analyzing the linear correlation of the obfuscated vectors and incoming vectors, the attacker cannot identify that the node is the source node of flow $i$ because the outgoing vectors have linear correlation with the incoming vectors, i.e., the behavior of node $s$ is the same as other intermediate nodes. Since the source node needs to generate $h$ linearly independent GEVs (or CDVs) for generation $j$ of flow $i$, we will show in the following sections that if the number of packets from other flows passing through $s$ is larger than $h$, $s$ has high probability to generate $h$ linearly independent obfuscated vectors.

At the destination node $d$ of flow $i$, if some other flows go through it, it can also generate obfuscated vectors for such flows, exploiting the received vectors from flow $i$. Therefore, in both cases, the anonymity of source and destination nodes are protected. Moreover, even if an attacker can distinguish that some nodes are sources or destinations, it cannot distinguish which node communicates with which other node with high probability (to be shown in Section 4).

Fig. 4 gives an example, where solid directed lines denote packets of generation $j$ of flow $i$ and dotted directed lines denote packets of other flows. Let $h = 3$ and LNC is performed over $\mathbb{F}_3$. At the source of flow $i$, $\mathbf{a} = [1, 1, 1], \mathbf{b} = [2, 2, 0], \mathbf{c} = [2, 1, 0]$ are incoming GEVs from other flows. The source can generate obfuscated GEVs such as $\mathbf{d} = 2\mathbf{b} = [1, 1, 0], \mathbf{e} = \mathbf{a} + 2\mathbf{b} = [2, 2, 1]$, and $\mathbf{f} = \mathbf{a} + \mathbf{b} + \mathbf{c} = [2, 1, 1]$. At the destination, $\mathbf{o} = [1, 2, 2], \mathbf{p} = [0, 2, 1]$ are incoming GEVs from generation $j$ of flow $i$ and $\mathbf{m} = [0, 2, 1], \mathbf{n} = [1, 1, 0]$ are from generation $j'$ of flow $i'$. The destination can generate obfuscated GEVs for flow $i'$, such as $\mathbf{g} = \mathbf{m} + \mathbf{n} = \mathbf{o} + 2\mathbf{p} = [1, 0, 1], \mathbf{h} = \mathbf{m} + 2\mathbf{n} = 2\mathbf{o} = [2, 1, 1]$.

### 3.4 Random Linear Network Coding Scheme

In the previous subsection, we have discussed how to construct a secure linear code in a deterministic manner to implement *ALNCode*, by which the outgoing vectors (i.e., GEVs or CDVs) produced are guaranteed to obfuscate their correlation with the corresponding incoming vector, under mild conditions. In this section, we investigate the potential of using the random LNC to thwart traffic analysis attacks and analyze the probability that a random LNC can produce an obfuscated vector (i.e., GEV or CDV).

In practice, random LNC has been widely utilized to realize LNC [17], [18], [22]. We now investigate the behavior of random LNC, when it is applied to hide the correlation of incoming vectors and outgoing vectors. The main difference of deterministic LNC and random LNC is that deterministic LNC first computes the obfuscated basis, then generates the local encoding matrix in a deterministic manner, and finally generates coded packets with obfuscated vectors, while random LNC randomly selects a local encoding matrix to generate new coded packets. Using random LNC without computing the obfuscated basis to generate the local encoding matrix, the computational complexity to generate $r_1$ new coded packets is $O(r_1 f_1 H)$, which is lower than the proposed deterministic LNC scheme in Section 3.3.

However, without computing the obfuscated basis for each generation $j$ of flow $i$ before generating new vectors (i.e., GEVs or CDVs) and corresponding coded packets, the random LNC scheme cannot guarantee that each newly generated vector can obfuscate the attacker when $\dim(L_1 \cap L_2) \neq 0$. With regard to the probability that a randomly generated vector is an obfuscated vector, we have the following theorem.

**Theorem 4.** *For an intermediate node $k$, if $\dim(L_1) = r_1$ and $dim(L_1 \cap L_2) = N$, with the random LNC scheme, the probability that $k$ can generate an obfuscated new vector for generation $j$ of flow $i$ is no less than $1 - \frac{q^{r_1 - N} - 1}{q^{r_1} - 1}$.*

**Proof.** The proof is shown in *Appendix* F, available in the online supplemental material. □

Note that, for a given node, the values of $q, r_1$ are the same in both the case of generating obfuscated GEV and the case of generating obfuscated CDV. However, the value of $N$ in the case of generating obfuscated CDV is no more than the case of generating obfuscated GEV.

The above theorem shows that the lower bound of the probability that a newly vector generated by node $k$ for generation $j$ of flow $i$ increases with the increase of $N$, e.g., the dimension of the vector space $L_1 \cap L_2$. We have the following corollary.

**Corollary 2.** *Given $N, r_1$ and $0 < N < r_1$, the lower bound of the probability that a newly generated vector is an obfuscated vector increases with the increase of $q$.*

**Proof.** The proof is shown in *Appendix* G, available in the online supplemental material. □

According to Theorem 4, we can derive another lower bound, which is only affected by the size of finite field.

**Corollary 3.** *When $dim(L_1 \cap L_2) \neq 0$, i.e., $N > 0$, the probability that a newly generated vector is an obfuscated vector is larger than $1 - \frac{1}{q}$.*

**Proof.** The proof is shown in *Appendix* H, available in the online supplemental material. □

Theorem 4, Corollarys 2 and 3 reveal that (1) when $\dim(L_1 \cap L_2) > 0$, an obfuscated vector can be generated

with certain probability when we use the random LNC (Theorem 4), and (2) the probability will become higher when we select a larger finite field for the random LNC (Corollarys 2 and 3). Theorem 4 also reveals the impacts of other parameters to the probability, which are further studied in the numerical experiments. Overall, our analysis and experiments show that the legacy random LNC with appropriate parameters can be adopted to provide anonymity against traffic analysis attack with sufficiently high probability.

Finally, when $\dim(L_1 \cap L_2) > 0$, the deterministic LNC designed in Section 3.3 can guarantee that all the generated vectors are obfuscated. To this end, deterministic LNC is better than random LNC because random LNC may generate vectors that are not obfuscated. On the other hand, the computation complexity of random LNC is much smaller than that of the deterministic LNC.

## 4 Analysis on Anonymity Against Attacks

In this section, we discuss how the proposed *ALNCode* can practically provide anonymity against different traffic analysis attacks. We first show that the traffic analysis attacker can acquire the linear correlations of incoming vectors and an outgoing vector by exploiting efficient approaches. Then, we show that even if the attacker can obtain these correlations, the linear correlations among the vectors of the same flow can be hidden by selecting appropriate parameters to generate outgoing vectors which are linear correlated with a large number of incoming vectors from other flows. Finally, we illustrate the impacts of parameters on the system performance.

### 4.1 Approaches for Analyzing One Outgoing Coded Packet

In this subsection, we propose two approaches that can be used by the attacker to analyze the linear correlations of the incoming vectors and one outgoing vector.

Let the set of incoming vectors be $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_F\}$ and the outgoing vector be $\mathbf{v}$. Since the flow number and the generation number of these vectors are hidden, the attacker can only analyze the linear correlations between $\mathbf{v}$ and vectors in $\mathbf{V}$ to distinguish which set of vectors is from the same flow with $\mathbf{v}$.

To generate an outgoing vector, the intermediate node linearly combines a set of linearly independent incoming vectors of the same flow. From the attacker's point of view, it wants to distinguish these outgoing vectors and then traces them back to the source. However, the attacker cannot uniquely determine these incoming vectors when vector $\mathbf{v}$ can be generated by different linearly independent sets of incoming vectors. Therefore, by analyzing the linear correlations between $\mathbf{v}$ and $\mathbf{V}$, it only can know which set of incoming vectors can be used to generate vector $\mathbf{v}$.

We first show an approach for obtaining the exact linear correlations between an outgoing vector and all incoming vectors at an intermediate node.

If incoming vector $\mathbf{v}_i$ is used to generate the outgoing vector $\mathbf{v}$, then there exists a maximal linearly independent set of $\mathbf{V}$, denoted as $\mathbf{V}_I$, such that $\mathbf{v}_i \in \mathbf{V}_I$ and the coefficient of $\mathbf{v}_i$ to generate $\mathbf{v}$ is a nonzero number. Therefore, the attacker can first find all maximal linearly independent sets of $\mathbf{V}$ and then uniquely determine the linear correlations between $\mathbf{v}$ and each maximal linearly independent set of incoming vectors. For $F$ incoming vectors, there are at most $\binom{F}{R}$ different maximal linearly independent sets, in which $R = \dim(L(\{\mathbf{v}_1, \ldots, \mathbf{v}_F\}))$.

Although the above approach can obtain the linear correlations between $\mathbf{v}$ and each maximal linearly independent set of $\mathbf{V}$, it has exponentially computational complexity. Next, we show an approximation approach, based on which the attacker can easily determine the incoming vectors that have no linear correlations with $\mathbf{v}$ and find the set of incoming vectors that can be used to generate vector $\mathbf{v}$.

If an incoming vector $\mathbf{v}_i$ has linear correlation with outgoing vector $\mathbf{v}$, then we have $\mathbf{v} = \sum_{l=1}^{F} \alpha_l \mathbf{v}_l$ and $\alpha_i \neq 0$. It means $\mathbf{v}_i = \frac{1}{\alpha_i}(\sum_{l \in \{1,\ldots,F\}-\{i\}} \alpha_l \mathbf{v}_l - \mathbf{v})$. Therefore, $\mathbf{v}_i$ is a linear combination of $\{\mathbf{v}_1, \ldots, \mathbf{v}_F, \mathbf{v}\} - \{\mathbf{v}_i\}$. Therefore, according to this necessary condition, the attacker first generates a matrix $\mathcal{M} = [\mathbf{v}_1^T, \ldots, \mathbf{v}_F^T, \mathbf{v}^T]^T$; then for each $i \in \{1, \ldots, F\}$, generates a new matrix $\mathcal{M}^i$ by replacing the $i$th row vector in $\mathcal{M}$, i.e., $\mathbf{v}_i$, by a zero vector; at last compares the ranks of matrix $\mathcal{M}$ and matrix $\mathcal{M}^i$. If $\mathrm{rank}(\mathcal{M}) \neq \mathrm{rank}(\mathcal{M}^i)$, then the $i$th incoming vector, i.e., $\mathbf{v}_i$, does not have linear correlation with outgoing vector $\mathbf{v}$. After removing all the incoming vectors that have no linear correlations with $\mathbf{v}$ the attacker can find the set of incoming vectors that may be used to generate vector $\mathbf{v}$. Since Gaussian elimination method is used for each index $i$ in $\{1, \ldots, F\}$, the total computational complexity of this analysis approach is $O(F^2 \mathtt{t} \min(F+1, \mathtt{t}))$.

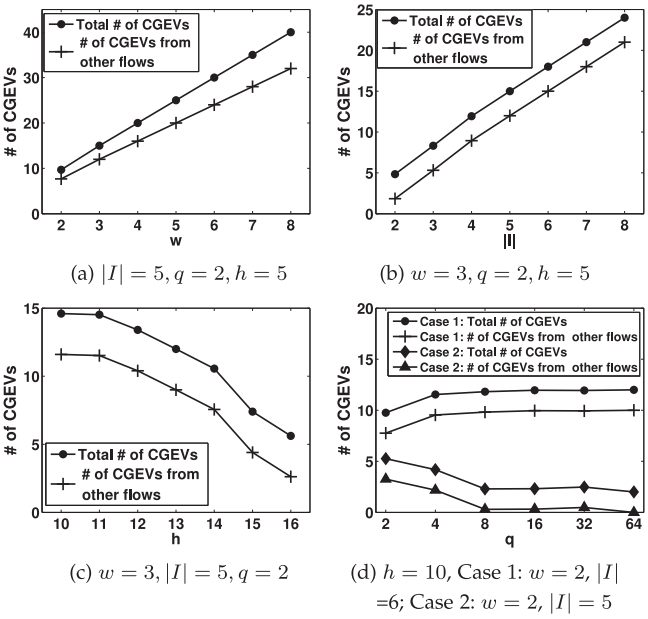
We note that there may exist more efficient analysis approaches to find these linear correlations. However, regardless which approach the attacker will use, the proposed *ALNCode* can hide the linear correlations by selecting appropriate parameters to generate outgoing vectors which are linear correlated with a large number of incoming vectors from other flows.

### 4.2 Linear Dependence with Multiple Incoming Coded Packets

In this subsection, we first give the theoretical analysis to show the lower bound of the probability that an outgoing vector (i.e., GEV or CDV) has linear correlations with multiple subset of incoming vectors. Then, we show the average number of incoming vectors linearly correlated with the outgoing vector generated by the proposed *ALNCode*, i.e., the anonymity level.

One outgoing vector can be produced uniquely from the incoming vectors, i.e., the subset of incoming vectors linearly correlated with the outgoing vector can be uniquely found, iff the incoming vectors are linearly independent. However, we will show that the probability that the set of incoming vectors is linearly dependent, is not only affected by the length of data block ($H$), but also affected by the size of generation ($h$), the size of finite field ($q$) and the number of coded packets received by the intermediate node ($F$). Moreover, when appropriate parameters are selected, the generated vector have linear correlations with multiple subsets of incoming vectors with high probability even if $H$ goes to infinite.

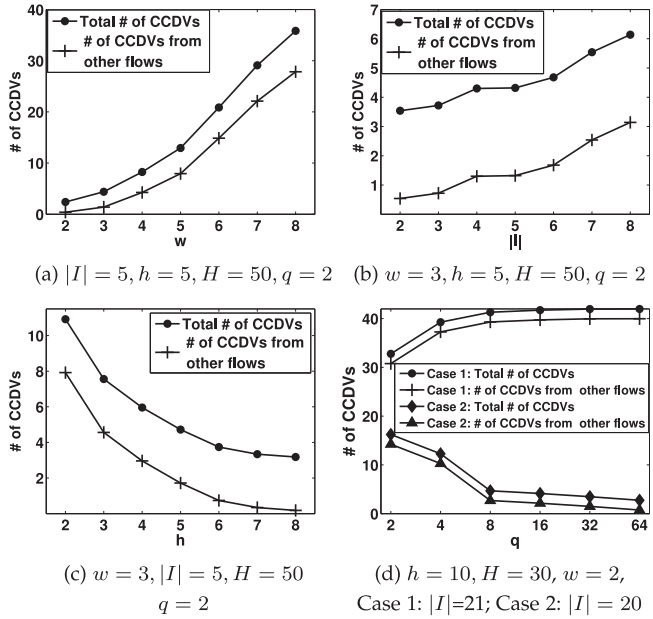
When assuming that all the incoming vectors (i.e., GEVs or CDVs) are randomly selected from vector space $\mathbb{F}_q^{\mathtt{t}}$, different incoming vectors, indeed, are linearly independent with high probability when $\mathtt{t}$ is large. Although original




(a) $|I| = 5, q = 2, h = 5$ (b) $w = 3, q = 2, h = 5$

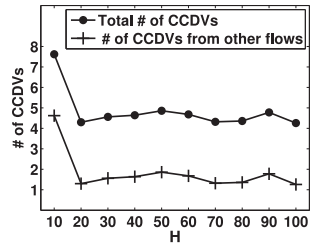
(c) $w = 3, |I| = 5, q = 2$ (d) $h = 10$, Case 1: $w = 2$, $|I|$ =6; Case 2: $w = 2$, $|I| = 5$

Fig. 5. Total number of CGEVs and the number of CGEVs from other flows under different influential parameters.




(a) $|I| = 5, h = 5, H = 50, q = 2$ (b) $w = 3, h = 5, H = 50, q = 2$

(c) $w = 3, |I| = 5, H = 50$ $q = 2$ (d) $h = 10, H = 30, w = 2,$ Case 1: $|I|$=21; Case 2: $|I| = 20$




(e) $w = 3, |I| = 5, h = 5, q = 2$

Fig. 6. Total number of CCDVs and the number of CCDVs from other flows under different influential parameters.

vectors generated at source node for each generation of a flow can be regarded as being randomly selected from vector space $\mathbb{F}_q^t$, but, in fact, all the vectors for a certain generation received by an intermediate node are not generated from the whole vector space $\mathbb{F}_q^t$, but from the same set of $h$ original vectors, i.e., a fixed $h$-dimensional vector space, which increases the probability that the set of incoming vectors is linearly dependent. Assume that an intermediate node receives $F$ incoming vectors, which are from a set of generations, denoted as $I$. We denote $w_i$ vectors received from generation $i$ as $\mathbf{M}'_i = \{\mathbf{m}'_{i,1}, \ldots, \mathbf{m}'_{i,w_i}\}$ and $\sum_{i \in I} w_i = F$. We give the lower bound of the probability that $F$ incoming vectors are linearly dependent as follows.

**Theorem 5.** *The probability that $F$ incoming GEVs (or CDVs) are linearly dependent, is no less than $1 - \prod_{i \in I} \prod_{j=h-w_i+1}^{h} (1 - \frac{1}{q^j})$ when $w_i \leq h, \forall i \in I$. Otherwise, the probability equals to 1.*

**Proof.** The proof is shown in *Appendix* I, available in the online supplemental material. □

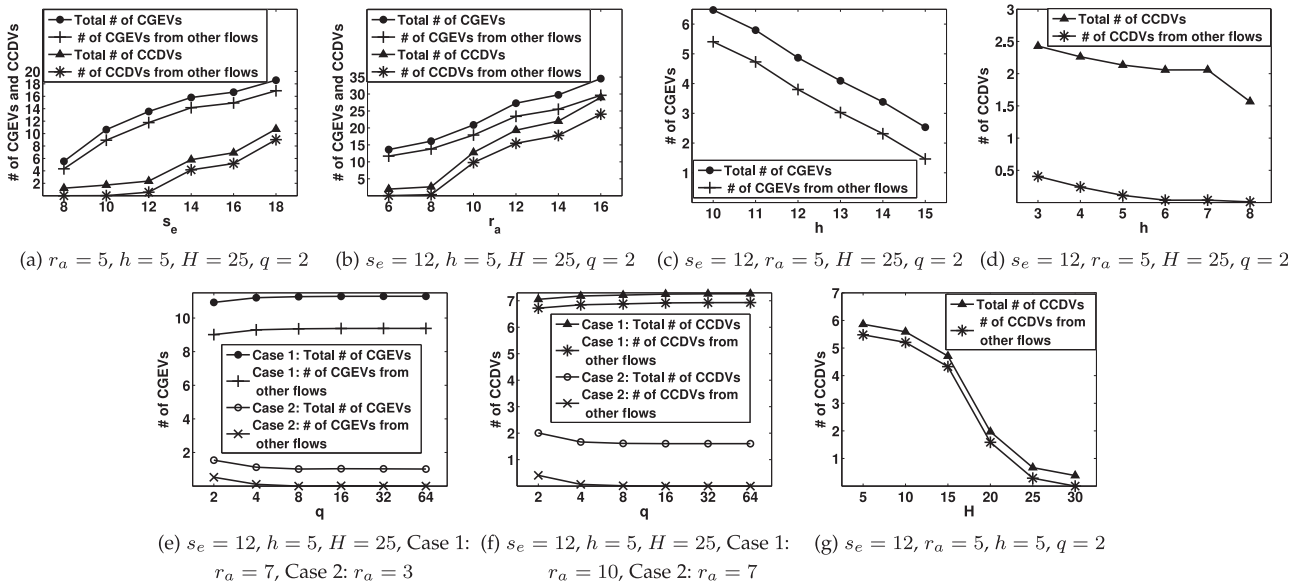
From the above theorem, the lower bound is only affected by $h$, $q$ and $F$. It means that when these parameters are fixed, the lower bound of the probability is fixed even when the length of the data block, $H$, goes to infinite. Therefore, when appropriate parameters are selected, the generated vectors have linear correlations with multiple subsets of incoming vectors with high probability even if the length of the data block goes to infinite (also to be shown in Fig. 6d).

If only the set of incoming vectors are linearly dependent, each outgoing vector has linear correlations with multiple subsets of incoming vectors, which not only increases the computational complexity of traffic analysis attack, but also decreases the probability that the attacker traces the flow back (forth) to the multiple sources (destinations), and consequently the attacker cannot identify the real source (destination).

Next, we first conduct two sets of simulations for generating obfuscated GEVs and CDVs, respectively, at a node to show the average number of incoming vectors linearly correlated with the generated vectors by the proposed *ALNCode* at different values of $w$, $|I|$, $h$, $H$ and $q$, in which $w$ denotes average number of incoming vectors from each generation and all the incoming vectors are from $|I|$ generations of different flows. We generate an outgoing vector (i.e., GEV or CDV) for different analysis attacks (i.e., GEV or CDV analysis attack). An incoming GEV is referred as *CGEV* if it is linearly correlated with the outgoing GEV. An incoming CDV is referred as *CCDV* for an outgoing CDV if it is linearly correlated with the outgoing CDV. We also test the ALNCode scheme in a 10-nodes ring network topology with multiple traffic flows and different transmission rates.

### 4.2.1 CGEVs and CCDVs on a Single Node

Fig. 5 shows the anonymity levels of the outgoing GEV for the GEV analysis attack. Figs. 5a and 5b show that the anonymity level of the outgoing GEV increases with the increase of $w$ and $|I|$, respectively. The reason is straightforward: when $h$ and $q$ are fixed, the more GEVs a node receives in the current flow and in other flows, the larger number of incoming GEVs have correlations with the generated outgoing GEV. On the other hand, Fig. 5c shows that the anonymity level of the outgoing GEV decreases with the increase of $h$. The reason is that when the number of GEVs received is fixed, the increase of length of each GEV will increase the probability that the incoming GEVs are linearly independent. Fig. 5d demonstrates different trends with the increase




(a) $r_a = 5, h = 5, H = 25, q = 2$ (b) $s_e = 12, h = 5, H = 25, q = 2$ (c) $s_e = 12, r_a = 5, H = 25, q = 2$ (d) $s_e = 12, r_a = 5, H = 25, q = 2$

(e) $s_e = 12, h = 5, H = 25$, Case 1: $r_a = 7$, Case 2: $r_a = 3$ (f) $s_e = 12, h = 5, H = 25$, Case 1: $r_a = 10$, Case 2: $r_a = 7$ (g) $s_e = 12, r_a = 5, h = 5, q = 2$

Fig. 7. Average number of CGEVs and CCDVs in a ring topology under different influential parameters.

TABLE 2
The impacts of coding parameters on the performance of ALNCode and random LNC

| | Encoding complexity for generate $r_1$ new coded packets | Decoding complexity | Communication overhead | Probability to generate an obfuscated GEV or obfuscated CDV when $L_1 \cap L_2 \neq \emptyset$ |
|---|---|---|---|---|
| ALNCode | for obfuscated GEV, $O(h^2(f_1+f_2)+r_1^2H)$; for obfuscated CDV, $O((h+H)^2(f_1+f_2)+r_1^2H)$ | $O(h^2H)$ | $\frac{h}{h+H}$ | $= 1$ (Theorem 3) |
| Traditional Random LNC | $O(h^2f_1+r_1^2H)$ | $O(h^2H)$ | $\frac{h}{h+H}$ | $\geq 1-\frac{q^{r_1-N}-1}{q^{r_1}-1}$ (Theorem 4) |

of $q$ in different cases. In particular, when $w|I| \leq h$, the anonymity level decreases with the increase of $q$; when $w|I| > h$, it increases with the increase of $q$. It has similar trends and reasons with the performance of the intersection probability shown in Fig. 3d. We note that the length of the data block, i.e., $H$, has no impact on the anonymity levels of the outgoing GEV for the GEV analysis attack. However, it has impact on the anonymity levels of the outgoing CDV for the CDV analysis attack.

We show the anonymity levels of the outgoing CDV for the CDV analysis attack in Fig. 6. From figures Figs. 6a, 6b, 6c, 6d, performances of the anonymity levels of the outgoing CDV have similar trends with the performance of the linearly dependent probability shown in Fig. 5. From Fig. 6e, when parameters $w$, $|I|$, $q$ and $h$ are fixed, the anonymity levels will be almost fixed even when the length of the data block goes to infinite, which also reflects the lower bound proved in Theorem 5. Therefore, when appropriate parameters are selected, the generated GEVs (or CDVs) have linear correlations with multiple subsets of incoming GEVs (or CDVs) with high probability even if $H$ goes to infinite.

#### 4.2.2 CGEVs and CCDVs in a Ring Network

We next show the simulation results of the ALNCode scheme in a 10-nodes ring network topology[3] with multiple traffic flows and different transmission rates. Specifically, there are ten nodes in the ring network. The number of traffic flows is denoted as $s_e$. For each traffic flow, we randomly select two nodes as its source and destination. Moreover, the transmission rate of each traffic flow is denoted as $r_a$. Under each combination of parameters, for each node and each traffic flow passing through it, we randomly select an outgoing coded packet for the flow and compute the number of CGEVs and CCDVs, based on which we obtain the average number of CGEVs and CCDVs per node and flow, i.e., the anonymity levels, for each combination of parameters. The simulation results are shown in Figs. 7a, 7b, 7c, 7d, 7e, 7f, 7g.

From these results, both the number of CGEVs and the number of CCDVs increase with the increase of session number $s_e$ and the transmission rate $r_a$, because the number of coded data packets passing through each node grows larger. When parameters $s_e$ and $r_a$ are fixed, the performances of the anonymity levels have similar trends with the performance of CGEVs and CCDVs shown in Figs. 5 and 6.

### 4.3 The Impacts of Parameters on the System Performance

Based on the theoretical analysis and simulation results, in this subsection, we discuss the impacts of parameters on the system performance including computation complexity, overhead, etc. Specifically, we first show Table 2 to explicitly explain the impacts of various parameters in the ALNCode scheme and the traditional LNC scheme without anonymity consideration. The last column of Table 2 shows the probability that a new outgoing vector generated by

3. We have tested the ALNCode in different topologies. Due to limited space, we present only the results obtained in a ring topology.

node $k$ is an obfuscated GEV or obfuscated CDV, which has a positive correlation with the anonymity level.

For the tradeoffs between anonymity, system performance, complexity and overheads, we have the following conclusions.

The decrease of $h$ will (1) increase the anonymity level (shown in Figs. 5c, 6c, 7c and 7d); (2) decrease the encoding and decoding complexity (shown in Table 2); (3) decrease the communication overheads (shown in Table 2); and (4) decrease network throughput [23].

The decrease of $q$ will (1) decrease the anonymity level when many coded data packets can be received at each node (shown in Figs. 5d, 6d, 7e and 7f); (2) increase the anonymity level when few coded data packets can be received at each node (shown in Figs. 5d, 6d, 7e and 7f); (3) decrease the decoding complexity (shown in Table 2); (4) decrease the communication overheads (shown in Table 2); and (5) decrease the network throughput because it decreases the probability of the independence between the coded data packets received by destination [23].

### 4.4 Discussions on the Implementation of the ALNCode Scheme

To implement ALNCode, existing schemes for implementing LNC can be used, including encoding, decoding, routing, etc. To provide anonymity by using ALNCode, one action needed is to apply Algorithm 1 on each node so as to generate local encoding vectors; another additional operation is to perform flow monitoring. With traffic information, the *ALNCode* parameters can be optimized, including $q$, $h$, etc.

Moreover, if there are only few flows in the network, using *ALNCode* alone is vulnerable because the number of coded packets ($f_1$ and $f_2$) received by each node is small. In this case, to improve the anonymity level, the proposed *ALNCode* shall be combined with some existing techniques, such as dummy traffic [11], [13], [24].

For each node, when $\dim(L_1 \cap L_2) = 0$, the node uses the conventional LNC scheme to generate new coded vectors. When $\dim(L_1 \cap L_2) > 0$, it uses the proposed *ALNCode* scheme to generate new obfuscated coded vectors, the analysis and design show that it does not compromise the decodability. In particular, if an intermediate node receives $r$ linearly independent incoming packets from a flow, it can generate $r$ obfuscated linearly independent outgoing packets. It means that the span space of these newly generated vectors is also $L_1$. Therefore, the decodability of *ALNCode* is the same as that of the conventional deterministic LNC scheme. On the other hand, the proposed Algorithm 1 requires more computation, which is mainly due to the process about finding the basis of incoming coded vectors. Nevertheless, since Gaussian elimination is used to find the basis, the computing task can be done progressively. For instance, when the first two coded packets are received, an elimination operation can be performed. In other words, by the time that the last packet in a generation is received, the elimination process is almost done. In this manner, the proposed scheme will not significantly affect the throughput and latency.

For the traditional random LNC scheme, our analysis shows that it can generate an obfuscated vector with sufficiently high probability (Theorem 4). In this case, the traditional random LNC scheme is not changed, but some parameters (such as the finite field) may be updated, which may increase the computation overhead.

About the anonymity level, in this paper, we define the anonymity level as the number of incoming vectors that are linearly correlated with an outgoing vector. First, the anonymity level is not considered in the design of the *ALNCode*. Second, the anonymity level reflects the complexity and accuracy of traffic analysis, because the higher anonymity level, the more vectors should be traced back and analyzed, which leads to higher computational complexity and lower accuracy of traffic analysis. Third, in this paper, we have conducted extensive simulation experiments to demonstrate the anonymity level (i.e., the average number of CGEV and CCDV) in different scenarios when the proposed *ALNCode* is used. We will consider the anonymity level in the design of *ALNCode* to further improve the performance of anonymity in our future work.

## 5 Related Works

LNC has been widely explored in recent years, which has been proved to achieve the maximum throughput bound of a network [14]. If the local encoding vectors can be randomly selected by each intermediate node, the LNC scheme is referred to as *random LNC* [17], [18]. Random LNC makes LNC more practical. Otherwise, if the local encoding vector must be selected to achieve some properties, the LNC scheme is referred to as *deterministic LNC* [25], [26], [27]. In our work, we studied both deterministic and random LNC schemes.

In addition to achieving the maximum throughput of a network, the information security also can be provided by LNC in a content distribution network against active modification attacks [28] and passive wiretapping attacks [25], [26], [27], [29]. With respect to defense against wiretapping attacks, the main focus has been on exploring the capability of LNC to provide confidentiality of the packet content [25], [26], [27].

Although many works have been done on LNC design to provide confidentiality, few efforts have been devoted to utilizing LNC on communication anonymity. Among all attack models against anonymity, traffic analysis attack is a major one in traditional networks [11], [12], [13], [30], [31]. There mainly exist three representative approaches on defending against traffic analysis attack in traditional networks: *the Crowds approach*, *the onion routing approach*, and *the Mix approach*.

*Crowds* [30] provides a centralized service to randomly select participants of a network into a group (the "crowd"), which includes the source. Each packet is routed through the crowd before it is sent to the destination node, such that the attacker cannot tell which node in the crowd is the original source. In the onion routing approach [12], [31], the source establishes a path to the destination through a number of nodes called *onion routers*, and encrypts the routing information and packet repeatedly with public keys of the onion routers, in order to prevent any attacker from learning the path information. With the Mix approach [11], [13], instead of forwarding each packet as it arrives, an intermediate node, i.e., the *Mix* node, waits for a random period of time and then forwards packets it received in mixed order, so as to hide the time correlation among packets of the same flow. These existing approaches either require centralized

services, which is not scalable, or demands encryption of whole packets, which is computationally expensive. Moreover, these approaches cannot be directly implemented in the network with LNC because of the coding operations on each intermediate node.

Among the few proposals which utilize LNC for anonymous communication, we have discussed the works by Fan et al. [4] and Zhang et al. [6] in Section 1. Although the existing anonymous LNC schemes can hide the correlation among GEVs, an attacker can compromise the flow untraceability by evaluating the correlation of incoming and outgoing CDVs. In this paper, we give a novel idea to hide the correlation among GEVs (or CDVs) of the same flow by fully utilizing the properties of the LNC itself.

In [32], [33], the authors proposed two Joint FoUntain coding and Network coding (FUN) schemes to boost information spreading over multi-hop lossy networks. The coding schemes in FUN can significantly increase the throughput of information spreading by optimally combining fountain coding, intra-session network coding, and cross-next-hop network coding. Since linearly combining coded vectors is an essential function in FUN schemes, we believe that our scheme can be extended to improve the anonymity in FUN-based networks.

## 6 Conclusion

In this paper, we have systematically investigated the potentials of using linear network coding to provide flow untraceability against traffic analysis attack that is based on the correlation of incoming and outgoing coded packets. Specifically, we proposed a novel LNC mechanism, *ALNCode*, to protect anonymity of source, destination, and paths of each flow with a simple but novel idea: the correlation of incoming and outgoing coded packets in one flow can be hidden by generating coded packets that are linearly correlated with packets of other incoming flows. To implement the *ALNCode*, we designed a deterministic LNC scheme and investigated how the *ALNCode* can help a standard random LNC scheme to thwart traffic analysis attack. In our study, we developed comprehensive theoretical analysis on the existence of obfuscated coding vectors (GEVs or CDVs), and we conducted extensive simulation experiments to evaluate the behaviors of *ALNCode* in various networks. Theoretical and simulation results demonstrate that the *ALNCode* can effectively defend against traffic analysis attacks even if the coded packets are not encrypted.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 61672370 and 61502328), the General Research Fund from Hong Kong Research Grant Council (No. 120612), the Shanghai Oriental Scholar Program, Natural Science Foundation of the Higher Education Institutions of Jiangsu Province (No. 16KJB520040), Science Technology and Innovation Committee of Shenzhen Municipality (No. JCYJ20170818095109386), NSFC-Guangdong Joint Fund under project U1501254 and National Science Foundation under grant CNS-1730325. Part of this work has been published in IEEE INFOCOM 2011 [1]. This version contains at least 50 percent new materials.

## References

[1] J. Wang, J. Wang, C. Wu, K. Lu, and N. Gu, "Anonymous communication with network coding against traffic analysis attack," in *Proc. 30th. Comput. Commun.*, Apr. 2011, pp. 1008–1016.

[2] T. Li, T. Jung, Z. Qiu, H. Li, L. Cao, and Y. Wang, "Scalable privacy-preserving participant selection for mobile crowdsensing systems: Participant grouping and secure group bidding," *IEEE Trans. Netw. Sci. Eng.*, 2018, https://ieeexplore.ieee.org/document/8254369

[3] J. Wang, K. Lu, J. Wang, J. Zhu, and C. Qiao, "ULNC: An untraceable linear network coding mechanism for mobile devices in wireless mesh networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7621–7633, Sep. 2016.

[4] Y. Fan, Y. Jiang, H. Zhu, J. Chen, and X. Shen, "Network coding based privacy preservation against traffic analysis in multi-hop wireless networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 3, pp. 834–843, Mar. 2011.

[5] A. F. Atya, T. ElBatt, and M. Youssef, "On the flow anonymity problem in network coding," in *Proc. 9th Int. Wireless Commun. Mobile Computing. Conf.*, Jul. 2013, pp. 225–230.

[6] P. Zhang, C. Lin, Y. Jiang, Y. Fan, and X. Shen, "A lightweight encryption scheme for network-coded mobile ad hoc networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2211–2221, Sep. 2014.

[7] G. Fanti, P. Kairouz, S. Oh, K. Ramchandran, and P. Viswanath, "Hiding the rumor source," *IEEE Trans. Inf. Theory*, vol. 63, pp. 6679–6713, Oct. 2017.

[8] S. Spencer and R. Srikant,"On the impossibility of localizing multiple rumor sources in a line graph," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 43, pp. 66–68, 2015.

[9] X. Lin, R. Lu, Z. Huafei, P.-H. Ho, X. Shen, and Z. Cao, "ASRPAKE: An anonymous secure routing protocol with authenticated key exchange for wireless ad hoc networks," in *Proc. IEEE Int. Conf. Commun.*, 2007, pp. 1247–1253.

[10] T. Isdal, M. Piatek, A. Krishnamurthy, and T. Anderson, "Privacy-preserving p2p data sharing with oneswarm," in *Proc. ACM SIGCOMM 2010*, Sep. 2009, pp. 111–122.

[11] M. Rennhard, "Introducing MorphMix: Peer-to-peer based anonymous internet usage with collusion detection," in *Proc. Workshop Privacy Electron. Soc.*, 2002, pp. 91–102.

[12] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE J. Select. Areas Commun.*, vol. 16, no. 4, pp. 482–494, May 1998.

[13] G. Danezis, R. Dingledine, D. Hopwood, and N. Mathewson, "Mixminion: Design of a type III anonymous remailer protocol," in *Proc. Workshop Privacy Electronic. Soc.*, 2003, pp. 2–15.

[14] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[15] S. Lakshminarayana and A. Eryilmaz, "Multirate multicasting with intralayer network coding," *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1256–1269, 2013.

[16] K. Sachin, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh network," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 401–412, Oct. 2008.

[17] T. Ho, M. Medard, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[18] J. M., F. C., and D. S., "Subspace properties of randomized network coding," in *Proc. of IEEE Inf. Theory Workshop Inf. Theory Wireless Netw.*, 2007, pp. 1–5.

[19] K. M. Hoffman and R. Kunze, *Linear Algebra*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, Apr. 1971.

[20] K.-W. Yang, "A basis for the intersection of subspaces," *Math. Mag.*, vol. 70, no. 4, Oct. 1997, Art. no. 297.

[21] M. Zhang, H. Li, F. Chen, H. Hou, H. An, W. Wang, and J. Huang, "A general co/decoder of network coding in HDL," in *Proc. Int. Symp. Networking Coding*, 2011, pp. 1–5.

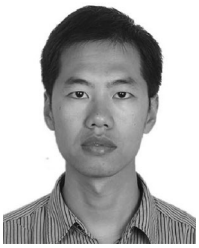
[22] L. Lima, M. Medard, and J. Barros, "Random linear network coding: A free cipher?" in *Proc. IEEE Int. Symp. Inf. Theory*, 2007, pp. 546–550.

[23] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Annu. Allerton Conf. Commun. Control Comput.*, vol. 41, 2003, pp. 40–49.

[24] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing for anonymous and private internet connections," *Commun. ACM*, vol. 42, no. 2, pp. 39–41, 1999.

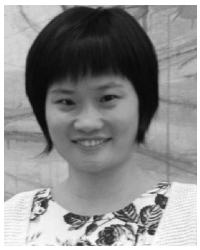
[25] N. Cai and R. Yeung, "Secure network coding," in *Proc. IEEE Int. Symp. Inf. Theory*, 2002, Art. no. 323.

[26] K. Bhattad and K. R. Narayanan, "Weakly secure network coding," in *Proc. 1st Workshop Netw. Coding Theory Appl.*, 2005, pp. 1–6.
[27] J. Wang, J. Wang, K. Lu, B. Xiao, and N. Gu, "Modeling and optimal design of linear network coding for secure unicast with multiple streams," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 10, pp. 2025–2035, Oct. 2013.
[28] A. Newell, J. Dong, and C. Nita-Rotaru, "On the practicality of cryptographic defences against pollution attacks in wireless network coding," *ACM Comput. Surveys*, vol. 45, no. 3, pp. 39:1–39:26, 2013.
[29] P. H. Che, M. Chen, T. Ho, S. Jaggi, and M. Langberg, "Routing for security in networks with adversarial nodes," in *Proc. Int. Symp. Netw. Coding*, 2013, pp. 1–6.
[30] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for web transactions," *ACM Trans. Inf. Syst. Secur.*, vol. 1, no. 1, pp. 66–92, 1998.
[31] R. Snader and N. Borisov, "Improving security and performance in the tor network through tunable path selection," *IEEE Trans. Dependable Secure Comput.*, vol. 8, no. 5, pp. 728–741, Sep. 2011.
[32] H. Zhang, K. Sun, Q. Huang, Y. Wen, and D. Wu, "FUN coding: Desging and analysis," *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 3340–3353, Dec. 2016.
[33] Q. Huang, K. Sun, X. Li, and D. Wu, "Just FUN: A joint fountain coding and network coding approach to loss-tolerant information spreading," in *Proc. ACM MobiHoc*, 2014, pp. 1–6.
[34] Z. Wan, *Geometry of Classical Groups over Finite Fields*, 2nd ed. Beijing, P.R. China: Science Press, 2002.



**Jin Wang** (M'12) received the BS degree from Ocean University of China, in 2006, and the PhD degree in computer science jointly awarded by City University of Hong Kong and University of Science and Technology of China, in 2011. He is currently an associate professor with the Department of Computer Science and Technology, Soochow University, Suzhou, China. His research interests include network coding, network security, content-centric networks, and edge computing. He is a member of the IEEE.

**Kejie Lu** (S'01-M'04-SM'07) received the BSc and MSc degrees from Beijing University of Posts and Telecommunications, Beijing, China, in 1994 and 1997, respectively, and the PhD degree in electrical engineering from the University of Texas at Dallas, Richardson, Texas, in 2003. In July 2005, he joined the University of Puerto Rico at Mayagüez, Mayagüez, Puerto Rico, where he is currently a professor with the Department of Computer Science and Engineering. His research interests include computer and communication networks, cyber-physical system, and network-based computing. He is a senior member of the IEEE.



**Jianping Wang** received the BS and MS degrees in computer science from Nankai University, Tianjin, China, in 1996 and 1999, respectively, and the PhD degree in computer science from the University of Texas at Dallas, in 2003. She is currently a professor with the Department of Computer Science, City University of Hong Kong. Her research interests include optical networks, wireless networks, and cloud computing. She is a member of the IEEE.

**Chuan Wu** (M'08) received the BSc and MSc degrees in computer science and technology from Tsinghua University, Beijing, China, in 2000 and 2002, respectively, and the PhD degree in electrical and computer engineering from the University of Toronto, Canada, in 2008. She is currently an associate professor with the Department of Computer Science, University of Hong Kong, Hong Kong. Her research interests include cloud computing, software defined networking, online and mobile social networks. She is a member of the IEEE.

**Naijie Gu** received the BS degree in computational mathematics from the University of Science and Technology of China (USTC), Hefei, China, in 1983 and the MS and PhD degrees in computer science from USTC, in 1989 and 2004, respectively. He is currently a professor with the Department of Computer Science, University of Science and Technology of China. His research interests include parallel and distributed computing and multicast technology on IP layer.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.