

Circular-shift Linear Network Codes with Arbitrary Odd Block Lengths

Qifu Tyler Sun[†], Hanqi Tang[†], Zongpeng Li[‡], Xiaolong Yang[†], and Keping Long[†]

[†]Department of Communication Engineering, University of Science and Technology Beijing, China

[‡] School of Computer Science, Wuhan University, China

Abstract—Circular-shift linear network coding (LNC) is a special type of vector LNC with low encoding and decoding complexities, with local encoding kernels chosen from cyclic permutation matrices. When L is a prime with primitive root 2, it was recently shown that a scalar linear solution over $\text{GF}(2^{L-1})$ can induce an L -dimensional circular-shift linear solution at rate $(L-1)/L$. In this work, we prove that for an arbitrary odd L , every scalar linear solution over $\text{GF}(2^{m_L})$, where m_L refers to the multiplicative order of 2 modulo L , can induce an L -dimensional circular-shift linear solution at a certain rate. Based on the generalized connection, we further prove that for every multicast network, as long as m_L is larger than a threshold, there exists an L -dimensional circular-shift linear solution at rate $\phi(L)/L$, where $\phi(L)$ means the Euler's totient function of L . An efficient algorithm for constructing such a solution is also designed. Finally, we prove that every multicast network is asymptotically circular-shift linearly solvable.

I. INTRODUCTION

A multicast network is a finite directed acyclic multigraph, with a unique source node s and a set T of receivers. In a multicast network, the source s generates ω binary sequences of length L' , and every edge transmits a binary sequence of length L . A linear network coding (LNC) scheme computes an outgoing binary sequence from a non-source node v as a linear function of the incoming binary sequences to v . It qualifies as an L -dimensional linear solution at rate L'/L if every receiver can recover the ω source binary sequences of length L' from its incoming binary sequences of length L .

On a binary sequence, a circular-shift operation can be not only software implemented with a negligible computational complexity compared with bit-wise additions, but amenable to implementation through atomic hardware operations as well. In order to reduce the encoding complexity of LNC, Ref. [1]-[4] studied LNC schemes with circular-shifts as the linear operations on a binary sequence. Specifically, when L is a large enough prime minus 1, a low-complexity linear solution at rate 1 was designed in [1] for a special class of multicast networks known as Combination Networks. The LNC schemes studied in [2] are called rotation-and-add linear codes, and are applicable to an arbitrary multicast network. The ones studied in [3] are called BASIC functional-repair regenerating codes. BASIC codes are discussed in the context of a distributed storage system, which are essentially equivalent to a multicast network. When $L > |T|$ is a prime with primitive root 2, *i.e.*,

the multiplicative order of 2 modulo L is $L-1$, the existence of an L -dimensional rotation-and-add linear solution at rate $(L-1)/L$ and an L -dimensional BASIC functional-repair regenerating code at rate $(L-1)/L$ have been respectively shown in [2] and [3], from the approach of cyclic convolutional coding.

More recently, circular-shift LNC was formulated in [4] in the context of a general network and from the perspective of *vector* LNC. Compared with the conventional scalar LNC approach (See, e.g., [5][6]), which models binary sequences as elements in $\text{GF}(2^L)$, vector LNC (See, e.g., [7][8]) models binary sequences as vectors in $\text{GF}(2)^L$. The coding operations performed at intermediate nodes by scalar LNC and by vector LNC are linear functions over $\text{GF}(2^L)$ and over the ring of $L \times L$ binary matrices, respectively. Under the framework of vector LNC, the linear coding operation of circular-shifts on a binary sequence $[m_1 \ m_2 \ \dots \ m_L]$ by $1 \leq j \leq L-1$ positions to the right can simply be expressed as $[m_1 \ m_2 \ \dots \ m_L] \mathbf{C}_L^j = [m_{L-j+1} \ \dots \ m_L \ m_1 \ \dots \ m_{L-j}]$, where \mathbf{C}_L refers to the following $L \times L$ cyclic permutation matrix over $\text{GF}(2)$

$$\mathbf{C}_L = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix}.$$

A key advantage of such formulation utilized in [4] is that when L is odd, the cyclic permutation matrix \mathbf{C}_L^j can be diagonalized as

$$\mathbf{C}_L^j = \mathbf{V}_L \cdot \mathbf{\Lambda}_\alpha^j \cdot \mathbf{V}_L^{-1} \quad \forall j \geq 0 \quad (1)$$

where α is a primitive L^{th} root of unity (over $\text{GF}(2)$), \mathbf{V}_L is the $L \times L$ Vandermonde matrix generated by $1, \alpha, \dots, \alpha^{L-1}$ over $\text{GF}(2)(\alpha)$, the minimal field containing $\text{GF}(2)$ and α , and $\mathbf{\Lambda}_\alpha$ is the $L \times L$ diagonal matrix with diagonal entries equal to $1, \alpha, \dots, \alpha^{L-1}$. Prior to [4], such a diagonalization manipulation on \mathbf{C}_L has also been adopted in the rank analysis of quasi-cyclic LDPC codes [9][10] as well as certain quasi-cyclic stabilizer quantum LDPC codes [11].

When L is a prime with primitive 2, based on (1), it was revealed in [4] that every scalar linear solution over $\text{GF}(2^{L-1})$ induces an L -dimensional circular-shift linear solution at rate $(L-1)/L$. Thus, a rotation-and-add linear solution considered

Q. T. Sun (qfsun@ustb.edu.cn) is the corresponding author.

in [2] and a BASIC functional-repair regenerating code considered in [3] can be efficiently constructed via the efficient construction of a scalar linear solution.

In order to make the design of circular-shift LNC more flexible, in the present paper, we investigate an intrinsic connection between scalar LNC and circular-shift LNC for an arbitrary odd block length L , in the context of multicast networks. The main contributions and the organization of this paper are summarized as follows:

- After reviewing some preliminary knowledge of LNC in Section II, we introduce a method in Section III to obtain an L -dimensional circular-shift linear code from an arbitrary scalar linear code over $\text{GF}(2^{m_L})$, where m_L refers to the multiplicative order of 2 modulo L . Based on a rank analysis between the scalar linear code and the induced circular-shift linear code, we further turn the circular-shift linear code into an L -dimensional circular-shift linear solution at a certain rate.
- Under the general framework, in Section IV, we first prove the existence of an L -dimensional circular-shift linear solution at rate $\phi(L)/L$, as long as m_L is large enough, where one of the specific sufficient bounds is the number of receivers. An efficient algorithm to construct such a solution, via a flow path approach, is also proposed.
- Stemming from this, in Section V, we provide a positive answer to an open conjecture in [4]: every multicast network is asymptotically circular-shift linearly solvable.

II. PRELIMINARIES

We consider a multicast network in the present paper. The network notations and assumptions herein are the same as the ones in [12]. The data unit transmitted on every edge $e \in E$ is an L -dimensional row vector \mathbf{m}_e of binary data symbols. For every receiver $t \in T$, based on the $|\text{In}(t)|$ received data units, the goal is to recover the ω source data units generated by s . Without loss of generality, assume $|\text{Out}(s)| = |\text{In}(t)| = \omega$, and there is not any edge from s to a receiver. A topological order is assumed on E led by edges in $\text{Out}(s)$.

An L -dimensional vector linear code ($\mathbf{K}_{d,e}$) (over $\text{GF}(2)$ and at rate 1) is an assignment of a local encoding kernel $\mathbf{K}_{d,e}$, which is an $L \times L$ matrix over $\text{GF}(2)$, to every pair (d,e) of edges such that $\mathbf{K}_{d,e}$ is the zero matrix $\mathbf{0}$ when (d,e) is not an adjacent pair. For every edge e emanating from a non-source node v , the data unit vector $\mathbf{m}_e = \sum_{d \in \text{In}(v)} \mathbf{m}_d \mathbf{K}_{d,e}$. Every vector linear code uniquely determines a global encoding kernel \mathbf{F}_e , which is an $\omega L \times L$ matrix over $\text{GF}(2)$, for every edge e . A vector linear code is a *vector linear solution* if for every receiver $t \in T$, the column-wise juxtaposition $[\mathbf{F}_e]_{e \in \text{In}(t)}$ has full rank ωL . A 1-dimensional vector linear code is a scalar linear code, in which case we shall use the scalar symbol $k_{d,e}$ and the vector symbol \mathbf{f}_e to denote the local and global encoding kernels, respectively.

As formulated in [4], an L -dimensional circular-shift linear code of degree δ is an L -dimensional vector linear code with local encoding kernels selected from

$$\mathcal{C}_\delta = \left\{ \sum_{j=0}^{L-1} a_j \mathbf{C}_L^j : a_j \in \{0,1\}, \sum_{j=0}^{L-1} a_j \leq \delta \right\}, \quad (2)$$

that is, from matrices that can be written as summation of at most δ cyclic permutation matrices. There exist multicast networks which do not have an L -dimensional circular-shift linear solution of degree δ for any L and δ [4]. When L is a prime with primitive root 2, an L -dimensional circular-shift linear solution at rate $(L-1)/L$ can be readily obtained from a scalar linear solution over $\text{GF}(2^{L-1})$ subject to some local encoding kernel constraints, where an L -dimensional (fractional) linear code at rate L'/L is a variation of an L -dimensional vector linear code with the following differences: the ω data units $\mathbf{m}'_1, \dots, \mathbf{m}'_\omega$ generated at s are L' -dimensional row vectors over $\text{GF}(2)$, and each of the L binary data symbols in \mathbf{m}_e , $e \in \text{Out}(s)$, is a $\text{GF}(2)$ -linear combination of the ones in $\mathbf{m}'_1, \dots, \mathbf{m}'_\omega$, i.e., $[\mathbf{m}_e]_{e \in \text{Out}(s)} = [\mathbf{m}'_i]_{1 \leq i \leq \omega} \mathbf{G}_s$ for some $\omega L' \times \omega L$ source encoding matrix \mathbf{G}_s over $\text{GF}(2)$.

For brevity, an L -dimensional linear code at rate L'/L will be called an (L', L) linear code. An (L', L) linear code qualifies as a linear solution if for every receiver $t \in T$, the $\omega L' \times \omega L$ matrix $\mathbf{G}_s [\mathbf{F}_e]_{e \in \text{In}(t)}$ has full rank $\omega L'$. For an (L', L) linear solution, each receiver t has an $\omega L \times \omega L'$ decoding matrix \mathbf{D}_t over $\text{GF}(2)$ such that $\mathbf{G}_s [\mathbf{F}_e]_{e \in \text{In}(t)} \mathbf{D}_t = \mathbf{I}_{\omega L'}$, where $\mathbf{I}_{\omega L'}$ refers to the identity matrix of size $\omega L'$. Based on \mathbf{D}_t , the ω L' -dimensional source data units can be recovered at t via

$$[\mathbf{m}_e]_{e \in \text{In}(t)} \mathbf{D}_t = ([\mathbf{m}'_i]_{1 \leq i \leq \omega} \mathbf{G}_s [\mathbf{F}_e]_{e \in \text{In}(t)}) \mathbf{D}_t = [\mathbf{m}'_i]_{1 \leq i \leq \omega}.$$

As remarked in the previous section, a key reason for formulating circular-shift LNC from the perspective of vector LNC [4] is to utilize the diagonalization of cyclic permutation matrices for odd L in (1), which will also facilitate us to establish a more general connection between circular-shift LNC and scalar LNC in this work. In (1),

$$\mathbf{V}_L = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & \alpha & \dots & \alpha^{L-1} \\ \vdots & \vdots & \dots & \vdots \\ 1 & \alpha^{L-1} & \dots & \alpha^{(L-1)(L-1)} \end{bmatrix}, \quad (3)$$

$$\mathbf{\Lambda}_\alpha = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \alpha & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \alpha^{L-1} \end{bmatrix}, \quad (4)$$

$$\mathbf{V}_L^{-1} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & \alpha^{-1} & \dots & \alpha^{-(L-1)} \\ \vdots & \vdots & \dots & \vdots \\ 1 & \alpha^{-(L-1)} & \dots & \alpha^{-(L-1)(L-1)} \end{bmatrix}, \quad (5)$$

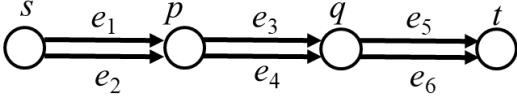


Fig. 1. A network consists of four nodes.

III. CIRCULAR-SHIFT LNC OVER ODD BLOCK LENGTHS

Hereafter in this paper, let L denote an odd integer, m_L denote the multiplicative order of 2 modulo L , and α be a primitive L^{th} root of unity over $\text{GF}(2)$. Then, the minimum field containing both $\text{GF}(2)$ and α is $\text{GF}(2^{m_L})$. When a scalar linear code over $\text{GF}(2^{m_L})$ is denoted by $(k_{d,e}(\alpha))$, it means that every local encoding kernel $k_{d,e}(\alpha)$ is the evaluation of a certain polynomial $k_{d,e}(x)$ over $\text{GF}(2)$ by setting x equal to α .

When L is a prime with primitive root 2, $m_L = L - 1$. In this special case, it has been revealed that in a general network, every scalar linear solution over $\text{GF}(2^{L-1})$ can induce an $(L - 1, L)$ circular-shift linear solution in a straightforward manner [4]. Actually, we next demonstrate that such construction of a circular-shift linear code also applies to the case that L is an odd integer.

On a multicast network, consider a scalar linear code $(k_{d,e}(\alpha))$ over $\text{GF}(2^{m_L})$. Define an L -dimensional circular-shift linear code $(\mathbf{K}_{d,e})$ by

$$\mathbf{K}_{d,e} = \begin{cases} \mathbf{0} & \text{if } k_{d,e}(x) = 0 \\ k_{d,e}(\mathbf{C}_L) & \text{otherwise} \end{cases} \quad (6)$$

where $k_{d,e}(\mathbf{C}_L)$ means an $L \times L$ matrix obtained via replacing x by \mathbf{C}_L in the polynomial $k_{d,e}(x)$. For an edge e , denote by $\mathbf{f}_e(\alpha)$ and \mathbf{F}_e its global encoding kernel determined by $(k_{d,e}(\alpha))$ and $(\mathbf{K}_{d,e})$, respectively. In addition, note that for each $0 \leq j \leq L - 1$, $(k_{d,e}(\alpha^j))$ also forms a scalar linear code over $\text{GF}(2^{m_L})$, where every $k_{d,e}(\alpha^j)$ is the evaluation of $k_{d,e}(x)$ by setting $x = \alpha^j$. Denote by $\mathbf{f}_e(\alpha^j)$ the global encoding kernel determined by $(k_{d,e}(\alpha^j))$ for edge e .

Theorem 1. For every receiver t ,

$$\text{rank}([\mathbf{F}_e]_{e \in \text{In}(t)}) = \sum_{j=0}^{L-1} \text{rank}([\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)}). \quad (7)$$

Proof. Please refer to Appendix-A. \square

Example. Consider the network depicted in Fig.1, which consists of a source node s , two relay nodes and the receiver t . Assume $L = 9$. In this case, $m_L = 6$ and α is a root of $x^6 + x^3 + 1$, which divides $x^9 + 1$. Consider the following scalar linear code $(k_{d,e}(\alpha))$ over $\text{GF}(2^6)$:

$$\begin{aligned} k_{e_1,e_3}(\alpha) &= k_{e_1,e_4}(\alpha) = k_{e_3,e_5}(\alpha) = 1 & k_{e_2,e_4}(\alpha) &= 1 + \alpha^3 \\ k_{e_2,e_3}(\alpha) &= k_{e_4,e_5}(\alpha) = k_{e_3,e_6}(\alpha) = 0 & k_{e_4,e_6}(\alpha) &= 1 + \alpha^6 \end{aligned}$$

Determined by $(k_{d,e}(\alpha))$, the global encoding kernels for incoming edges to t are

$$[\mathbf{f}_e(\alpha)]_{e \in \text{In}(t)} = \begin{bmatrix} 1 & 1 + \alpha^6 \\ 0 & (1 + \alpha^3)(1 + \alpha^6) \end{bmatrix} = \begin{bmatrix} 1 & 1 + \alpha^6 \\ 0 & 1 \end{bmatrix}.$$

When $k_{e_2,e_4}(\alpha) = 1 + \alpha^3$ and $k_{e_4,e_6}(\alpha) = 1 + \alpha^6$ are respectively thought of as the evaluation of defined polynomials $1 + x^3$ and $1 + x^6$, $k_{e_2,e_4}(\alpha^3) = 1 + \alpha^9 = 0$ and $k_{e_4,e_6}(\alpha^3) = 1 + \alpha^{18} = 0$. Thus, determined by $(k_{d,e}(\alpha^3))$, the global encoding kernels for incoming edges to t are $[\mathbf{f}_e(\alpha^3)]_{e \in \text{In}(t)} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$. Note that $\mathbf{f}_{e_6}(\alpha^3) = [0 \ 0]^T$ is calculated based on $k_{d,e}(\alpha^3)$ and it cannot be obtained from $\mathbf{f}_{e_6}(\alpha) = [1 + \alpha^6 \ 1]^T$ by simply replacing α with α^3 . One can further verify that $[\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ when $j \in \{0, 3, 6\}$, $[\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)} = \begin{bmatrix} 1 & 1 + \alpha^6 \\ 0 & 1 \end{bmatrix}$ when $j \in \{1, 4, 7\}$, and $[\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)} = \begin{bmatrix} 1 & 1 + \alpha^3 \\ 0 & 1 \end{bmatrix}$ when $j \in \{2, 8, 5\}$. Thus, $\sum_{j=0}^{L-1} \text{rank}([\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)}) = 15$.

Now consider the circular-shift linear code $(\mathbf{K}_{d,e})$ induced from $(k_{d,e}(\alpha))$ via (6). Determined by it, $[\mathbf{F}_e]_{e \in \text{In}(t)} = \begin{bmatrix} \mathbf{I}_9 & \mathbf{I}_9 + \mathbf{C}_9^6 \\ \mathbf{0} & (\mathbf{I}_9 + \mathbf{C}_9^3)(\mathbf{I}_9 + \mathbf{C}_9^6) \end{bmatrix}$, the rank of which equals 15 too.

Next, as $\alpha^6 + \alpha^3 + 1 = 0$, $k_{e_2,e_4}(\alpha)$ and $k_{e_4,e_6}(\alpha)$ can be respectively expressed as α^6 and α^3 and thought of as the evaluation of defined polynomials x^6 and x^3 . Under this new setting, $[\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ when $j \in \{0, 3, 6\}$, $[\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)} = \begin{bmatrix} 1 & \alpha^3 \\ 0 & 1 \end{bmatrix}$ when $j \in \{1, 4, 7\}$, and $[\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)} = \begin{bmatrix} 1 & 1 + \alpha^6 \\ 0 & 1 \end{bmatrix}$ when $j \in \{2, 8, 5\}$. Thus, $\sum_{j=0}^{L-1} \text{rank}([\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)}) = 18$. For the corresponding circular-shift linear code $(\mathbf{K}_{d,e})$ defined by (6), $[\mathbf{F}_e]_{e \in \text{In}(t)} = \begin{bmatrix} \mathbf{I}_9 & \mathbf{C}_9^3 \\ \mathbf{0} & \mathbf{I}_9 \end{bmatrix}$, the rank of which equals 18 too. \blacksquare

Theorem 1 establishes a connection between circular-shift LNC and scalar LNC for an arbitrary odd block length L . On one hand, it justifies that in the application of circular-shift LNC, the 1-bit redundancy during transmission is inevitable in the following sense. In order to make $[\mathbf{F}_e]_{e \in \text{In}(t)}$ full rank ωL for an L -dimensional circular-shift linear code, according to Eq. (7), the $\omega \times \omega$ matrix $[\mathbf{f}_e(1)]_{e \in \text{In}(t)}$ determined by the scalar linear code $(k_{d,e}(1))$ needs to be full rank ω . Since $(k_{d,e}(1))$ is defined over $\text{GF}(2)$, it directly endows low implementation complexity and there is no need to consider LNC at all.

On the other hand, it asserts that every scalar linear solution is possible to induce an (L', L) circular-shift linear solution at a certain rate L'/L .

For instance, as proved in [4], when L is a prime with primitive root 2, if an arbitrary scalar linear code $(k_{d,e}(\alpha))$ over $\text{GF}(2^{L-1})$ qualifies as a linear solution, then the scalar linear code $(k_{d,e}(\alpha^j))$ is a linear solution for all $1 \leq j \leq L - 1$ too. This property of $(k_{d,e}(\alpha))$, together with Theorem 1, guarantees that the L -dimensional circular-shift linear code $(\mathbf{K}_{d,e})$ defined by (6) satisfies $\text{rank}([\mathbf{F}_e]_{e \in \text{In}(t)}) \geq \omega(L - 1)$ for every receiver t . Thus, after appropriately designing a source encoding matrix \mathbf{G}_s , we can obtain an $(L - 1, L)$ circular-shift linear solution $(\mathbf{K}_{d,e}(\alpha))$.

Stemming from this idea, we next deal with the case that the block length L is an arbitrary odd integer. First we observe

the following property on a scalar linear code $(k_{d,e}(\alpha))$ over $\text{GF}(2^{m_L})$.

Lemma 2. If $(k_{d,e}(\alpha))$ is a scalar linear solution, then for every $j \geq 0$, the scalar linear code $(k_{d,e}(\alpha^{2^j}))$ qualifies as a linear solution too.

Proof. Consider a receiver t and a nonnegative integer j . It can be shown that the mapping $\sigma_j : \text{GF}(2^{m_L}) \rightarrow \text{GF}(2^{m_L})$ defined by $\sigma_j(\beta) = \beta^{2^j}$ is an automorphism of $\text{GF}(2^{m_L})$ that fixes the elements in $\text{GF}(2)$ (See, e.g., Theorem 2.21 in [13]). Thus, the full rank of $[\mathbf{f}_e(\alpha^{2^j})]_{e \in \text{In}(t)}$ can be readily implied by the full rank of $[\mathbf{f}_e(\alpha)]_{e \in \text{In}(t)}$ since $\det([\mathbf{f}_e(\alpha^{2^j})]_{e \in \text{In}(t)}) = \det([\mathbf{f}_e(\alpha)]_{e \in \text{In}(t)})^{2^j} \neq 0$. \square

Let \mathcal{J} be the set of integers between 0 and $L-1$ such that the scalar linear code $(k_{d,e}(\alpha^j))$ over $\text{GF}(2^{m_L})$ is a linear solution. As a consequence of Lemma 2, \mathcal{J} is closed under multiplication by 2 (modulo L). Let $J = |\mathcal{J}|$. Denote by $\tilde{\mathbf{I}}_{\mathcal{J}}$ the $J \times L$ matrix obtained from \mathbf{I}_L by deleting the $(j+1)^{\text{st}}$ row whenever $j \notin \mathcal{J}$, and by $\tilde{\mathbf{V}}$ the $J \times J$ matrix obtained from $\tilde{\mathbf{I}}_{\mathcal{J}} \mathbf{V}_L$ by restricting to the first J columns. For instance,

when $L = 15$ and $\mathcal{J} = \{1, 2, 4, 8\}$, $\tilde{\mathbf{V}} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^4 \\ 1 & \alpha^2 & \alpha^4 & \alpha^8 \\ 1 & \alpha^4 & \alpha^8 & \alpha \\ 1 & \alpha^8 & \alpha & \alpha^2 \end{bmatrix}$,

where $\alpha \in \text{GF}(2^4)$ is a primitive 15^{th} root of unity. As $\tilde{\mathbf{V}}$ can be regarded as a $J \times J$ Vandermonde matrix generated by $\alpha^j, j \in \mathcal{J}$, it is invertible. Define \mathbf{G} and \mathbf{G}_s , respectively, to be the $J \times L$ and $J\omega \times L\omega$ matrix

$$\mathbf{G} = \tilde{\mathbf{V}}^{-1} \tilde{\mathbf{I}}_{\mathcal{J}} \mathbf{V}_L^{-1}, \quad \mathbf{G}_s = \mathbf{I}_\omega \otimes \mathbf{G}. \quad (8)$$

Lemma 3. Every entry in \mathbf{G} , and hence in \mathbf{G}_s belongs to $\text{GF}(2)$.

Proof. Please refer to Appendix-B. \square

Justified by the above lemma, \mathbf{G}_s is defined over $\text{GF}(2)$, so it is a candidate for the source encoding matrix. The next theorem further proves that \mathbf{G}_s is indeed a desired one.

Theorem 4. Equipped with the source encoding matrix $\mathbf{G}_s = \mathbf{I}_\omega \otimes \mathbf{G}$, the circular-shift linear code $(\mathbf{K}_{d,e})$ constructed by (6) qualifies as a (J, L) linear solution.

Proof. This is continuation of the proof of Theorem 1, with the additional \mathbf{G}_s taken into account. We shall show that for every receiver t ,

$$\text{rank}(\mathbf{G}_s [\mathbf{F}_e]_{e \in \text{In}(t)}) = \sum_{j \in \mathcal{J}} \text{rank}([\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)}), \quad (9)$$

which yields $\text{rank}(\mathbf{G}_s [\mathbf{F}_e]_{e \in \text{In}(t)}) = \omega J$, so that the code $(\mathbf{K}_{d,e})$ is a (J, L) circular-shift linear solution by definition. The proof of (9) is provided in Appendix-C. \square

Remark. The source encoding matrix \mathbf{G}_s defined in (8) is not the only one to make the code $(\mathbf{K}_{d,e})$ a (J, L) linear solution, but it is a carefully designed one such that it applies to $(\mathbf{K}_{d,e})$ constructed from an arbitrary scalar linear code $(k_{d,e}(\alpha))$. One may wonder whether $\mathbf{I}_\omega \otimes \tilde{\mathbf{I}}_{\mathcal{J}}$ can also be used as a source

encoding matrix, because when $\mathcal{J} = \{1, 2, \dots, L-1\}$, it becomes exactly the one adopted in [4] for the constructed $(L-1, L)$ circular-shift linear solution. We remark here that $\mathbf{I}_\omega \otimes \tilde{\mathbf{I}}_{\mathcal{J}}$ is insufficient to be a source encoding matrix in a more general case, as illustrated in the next example.

Example. Assume $\omega = 2$, $L = 7$, and $[\mathbf{f}_e(x)]_{e \in \text{In}(t)} = \begin{bmatrix} 1 & & & & & & \\ 0 & 1+x+x^2+x^4 & & & & & \end{bmatrix}$ for some receiver t . In this case, when the primitive 7^{th} root of unity $\alpha \in \text{GF}(2^3)$ is selected subject to $\alpha + \alpha^2 + \alpha^4 = 1 + \alpha^3 + \alpha^5 + \alpha^6 = 0$, we have

$$\text{rank}([\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)}) = \begin{cases} 1 & \text{when } j \in \{0, 3, 5, 6\} \\ 2 & \text{when } j \in \{1, 2, 4\} \end{cases}$$

Correspondingly, set $\mathcal{J} = \{1, 2, 4\}$ and $\tilde{\mathbf{I}}_{\mathcal{J}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$. By (8), $\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$, and it can be checked that

$$\text{rank}((\mathbf{I}_2 \otimes \mathbf{G})[\mathbf{f}_e(\mathbf{C}_7)]_{e \in \text{In}(t)}) = 6.$$

In contrast,

$$\text{rank}((\mathbf{I}_2 \otimes \tilde{\mathbf{I}}_{\mathcal{J}})[\mathbf{f}_e(\mathbf{C}_7)]_{e \in \text{In}(t)}) = 5,$$

so it is impossible for receiver t to recover all 6 source binary data symbols if $\mathbf{I}_2 \otimes \tilde{\mathbf{I}}_{\mathcal{J}}$ is set as the source encoding matrix.

IV. EXPLICIT CONSTRUCTION OF A CIRCULAR-SHIFT LINEAR SOLUTION

A. Existence of a Circular-shift Linear Solution

In the previous section, we have introduced a general method to construct a (fractional) circular-shift linear solution from a scalar linear solution, but there is no explicit characterization on the code rate and the degree of the constructed code. In this section, we proceed to introduce the construction of a $(\phi(L), L)$ circular-shift linear solution of degree δ , where $\phi(L)$ means the Euler's totient function of an odd integer L . For this goal, we need recall the concept of cyclotomic polynomials. Write

$$R = \{1 \leq r \leq L-1 : \gcd(r, L) = 1\}. \quad (10)$$

The L^{th} cyclotomic polynomial over $\text{GF}(2)$ is

$$Q_L(x) = \prod_{r \in R} (x - \alpha^r).$$

When L is a prime with primitive root 2, $Q_L(x)$ itself is an irreducible polynomial over $\text{GF}(2)$. For a general odd L , the following lemma will be useful.

Lemma 5. For an odd integer L , $\phi(L)$ is divisible by m_L . The cyclotomic polynomial $Q_L(x)$ factors into $\frac{\phi(L)}{m_L}$ irreducible polynomials $f_1(x), f_2(x), \dots, f_{\phi(L)/m_L}(x)$ over $\text{GF}(2)$ of the same degree m_L .

Proof. See, for example, Theorem 2.47 in [13]. \square

For $1 \leq \delta \leq L-1$, denote by

$$\mathcal{K}_\delta^{(x)} = \left\{ \sum_{j=0}^{L-1} a_j x^j : a_j \in \{0, 1\}, \sum_{j=0}^{L-1} a_j \leq \delta \right\} \quad (11)$$

the set of polynomials over GF(2) of degree at most $L-1$ and with at most δ nonzero terms. Analogously, for $0 \leq i \leq L-1$, write

$$\begin{aligned} \mathcal{K}_\delta^{(\alpha^i)} &= \left\{ k(\alpha^i) : k(x) \in \mathcal{K}_\delta^{(x)} \right\} \\ &= \left\{ \sum_{j=0}^{L-1} a_j \alpha^{ij} : a_j \in \{0, 1\}, \sum_{j=0}^{L-1} a_j \leq \delta \right\}, \end{aligned}$$

that is, every element in $\mathcal{K}_\delta^{(\alpha^i)}$ corresponds to evaluation of a polynomial in $\mathcal{K}_\delta^{(x)}$ by setting $x = \alpha^i$. Note that for a general odd L , it is possible to have two distinct polynomials $k_1(x), k_2(x) \in \mathcal{K}_\delta^{(x)}$ subject to $k_1(\alpha^i) = k_2(\alpha^i)$. Hence, $\mathcal{K}_\delta^{(\alpha^i)}$ is a *multiset* instead of a set. Moreover, when $\delta = m_L$, all elements in GF(2^{m_L}) are contained in $\mathcal{K}_\delta^{(\alpha)}$, because $\{1, \alpha, \alpha^2, \dots, \alpha^{m_L-1}\}$ forms a polynomial basis of GF(2^{m_L}). Denote by K_δ the number of distinct elements in $\mathcal{K}_\delta^{(\alpha)}$.

Based on Lemma 5, we can obtain the following lemma, which plays a key role to prove the existence of a circular-shift linear solution of degree δ on a multicast network for block length L subject to a constraint.

Lemma 6. Let L be an odd integer, and $g(x_1, x_2, \dots, x_n)$ a non-zero multivariate polynomial of degree at most D in every x_j over GF(2^{m_L}). When $\frac{m_L K_\delta}{\phi(L)} > D$, there exist $k_1(x), k_2(x), \dots, k_n(x) \in \mathcal{K}_\delta^{(x)}$ such that the evaluation

$$g(k_1(\alpha^r), k_2(\alpha^r), \dots, k_n(\alpha^r)) \neq 0$$

holds for all $r \in R$.

Proof. Please refer to Appendix-D. \square

Theorem 7. Consider a multicast network with the set T of receivers and an odd integer L with the associated set $\mathcal{K}_\delta^{(x)}$ defined in (11). When $\frac{m_L K_\delta}{\phi(L)} > |T|$, there exists a $(\phi(L), L)$ circular-shift linear solution of degree δ .

Proof. We need to show that when $\frac{m_L K_\delta}{\phi(L)} > |T|$, there exists an assignment of $k_{d,e}(x) \in \mathcal{K}_\delta^{(x)}$ to every adjacent pair (d, e) , such that for all $r \in R$, the scalar linear code $(k_{d,e}(\alpha^r))$ over GF(2^{m_L}) is a linear solution. This is because the circular-shift linear code $(\mathbf{K}_{d,e})$ constructed from such $(k_{d,e}(\alpha))$ by (6) is of degree δ , and is equipped with the source encoding matrix $\mathbf{G}_s = \mathbf{I}_\omega \otimes (\tilde{\mathbf{V}}^{-1} \tilde{\mathbf{I}}_{\mathcal{J}} \mathbf{V}_L^{-1})$, where \mathcal{J} set to be R , it is an $(|R|, L)$ circular-shift linear solution according to Theorem 4, with $|R| = \phi(L)$.

Assign every adjacent pair (d, e) an indeterminate $x_{d,e}$. Under the classical framework of LNC in [6] and the observation in [15], every multicast network can be associated with a polynomial, denoted by $F(*)$, over GF(2^{m_L}) in indeterminates $\{x_{d,e} : \text{adjacent pair } (d, e)\}$ such that

- the degree of $F(*)$ in every $x_{d,e}$ is at most $|T|$.
- a scalar linear code $(k_{d,e})$ is a linear solution if and only if the evaluation of $F(*)$ by setting $x_{d,e} = k_{d,e}$ is a nonzero value in GF(2^{m_L}).

When $\frac{m_L K_\delta}{\phi(L)} > |T|$, by Lemma 6, there exists an assignment of $k_{d,e}(x) \in \mathcal{K}_\delta^{(x)}$ for every adjacent pair (d, e) such that the

evaluation of $F(*)$ by setting $x_{d,e} = k_{d,e}(\alpha^r)$ is nonzero for all $r \in R$. \square

B. Efficient Construction

Given a subset F of a finite field GF(q) with $|F| \geq |T|$, a well-known efficient algorithm was proposed in [16], by a flow path approach, to construct a scalar linear solution $(k_{d,e})$ over GF(q) with all $k_{d,e} \in F$. In this subsection, we shall demonstrate that by the flow path approach, a $(\phi(L), L)$ circular-shift linear solution can also be efficiently construct.

Adopt the same notation as in the previous subsection and assume L is an odd integer. Justified by Theorem 4, in order to efficiently constructed a $(\phi(L), L)$ circular-shift linear solution of degree δ , it suffices to efficiently assign $k_{d,e}(x) \in \mathcal{K}_\delta^{(x)}$ for every adjacent pair (d, e) such that for each $r \in R$, the scalar linear code $(k_{d,e}(\alpha^r))$ over GF(2^{m_L}) is a linear solution. The algorithm below can efficiently construct such desired $k_{d,e}(x)$.

Algorithm 8. Assume $\frac{m_L K_\delta}{\phi(L)} > |T|$. As initialization,

- For each receiver t , associate an arbitrary collection \wp_t of ω edge-disjoint paths starting from $\text{Out}(s)$ and ending at $\text{In}(t)$, and set $I_t = \text{Out}(s)$;
- Set $[\mathbf{f}_e(x)]_{e \in \text{Out}(s)} = \mathbf{I}_\omega$;
- denote by C_j , $1 \leq j \leq \frac{\phi(L)}{m_L}$, the cyclotomic cosets modulo L , and by r_j an arbitrary entry in C_j .

For every non-source node v , according to a topological order, perform the following procedures for every $e \in \text{Out}(v)$ to assign $k_{d_i,e}(x) \in \mathcal{K}_\delta^{(x)}$ and compute $\mathbf{f}_e(x)$:

- 1) For each $d \in \text{In}(v)$, denote by T_d the set of such receivers t that the adjacent pair (d, e) is on some path in \wp_t . If $|T_d| = 0$, then set $k_{d,e}(x) = 0$.
- 2) Let $\{d_1, \dots, d_l\}$ denote the subset of $\text{In}(v)$ with $|T_{d_i}| > 0$ for all $1 \leq i \leq l$. Note that a receiver t can only appear in at most one set T_{d_i} , $1 \leq i \leq l$.
- 3) If $l = 0$, then end the current iteration for e .
- 4) Otherwise, for all $t \in T_{d_i}$, $1 \leq i \leq l$, and all $1 \leq j \leq \frac{\phi(L)}{m_L}$, compute the ω -dimensional (column) vector $\mathbf{w}_{t,j}$ over GF(2^{m_L}) subject to

$$\begin{aligned} \mathbf{f}_{d_i}(\alpha^{r_j})^T \mathbf{w}_{t,j} &= 1, \\ \mathbf{f}_{d'}(\alpha^{r_j})^T \mathbf{w}_{t,j} &= 0, \quad \forall d' \in I_t \setminus \{d_i\}, \end{aligned} \quad (12)$$

Note that as to be proved in Proposition 9 in the sequel, when the iteration for an edge completes,

$$\text{rank}([\mathbf{f}_{e'}(\alpha^{r_j})]_{e' \in I_t}) = \omega, \quad \forall t \in T, 1 \leq j \leq \phi(L)/m_L. \quad (13)$$

Thus, such $\mathbf{w}_{t,j}$ subject to (12) can always be computed. Then, replace I_t by $I_t \cup \{e\} \setminus \{d_i\}$.

- 5) For $i = 1$, set $k_{d_i,e}(x) = 1$ and define $\mathbf{f}(x) = \mathbf{f}_{d_i}(x)$. For $2 \leq i \leq l$, iteratively assign $k_{d_i,e}(x) \in \mathcal{K}_\delta^{(x)}$ and update $\mathbf{f}(x)$ in the following way so as to keep

$$\mathbf{f}(\alpha^{r_j})^T \mathbf{w}_{t,j} \neq 0 \quad \forall t \in \bigcup_{1 \leq i' \leq i} T_{d_{i'}}, 1 \leq j \leq \frac{\phi(L)}{m_L} \quad (14)$$

after every iteration i .

- If $\mathbf{f}(\alpha^{rj})^T \mathbf{w}_{t,j} \neq 0$ for all $t \in T_{d_i}$ and $1 \leq j \leq \frac{\phi(L)}{m_L}$, then set $k_{d_i,e}(x) = 0$, keep $\mathbf{f}(x)$ unchanged, and end the current iteration on i .
- Otherwise, for $1 \leq j \leq \frac{\phi(L)}{m_L}$, define \mathcal{A}_j as

$$\mathcal{A}_j = \left\{ -\frac{\mathbf{f}_{d_i}(\alpha^{rj})^T \mathbf{w}_{t,j}}{\mathbf{f}(\alpha^{rj})^T \mathbf{w}_{t,j}} : \text{either } t \in T_{d_i} \text{ and } \mathbf{f}(\alpha^{rj})^T \mathbf{w}_{t,j} \neq 0 \text{ or } t \in \bigcup_{1 \leq i' < i} T_{d_{i'}} \right\}$$

Note that under the inductive assumption (14) up to iteration $i-1$, whose correctness will also be justified in Proposition 9, such \mathcal{A}_j is well defined.

- Set $k_{d_i,e}(x)$ to be a polynomial in $\mathcal{K}_\delta^{(x)}$ such that for all $1 \leq j \leq \frac{\phi(L)}{m_L}$,

$$k_{d_i,e}(\alpha^{rj}) \neq 0, \quad k_{d_i,e}(\alpha^{rj})^{-1} \notin \mathcal{A}_j. \quad (15)$$

As to be justified in Proposition 9, such $k_{d_i,e}(x)$ can always be selected.

- Reset $\mathbf{f}(x)$ to be $\mathbf{f}(x) + k_{d_i,e}(x)\mathbf{f}_{d_i}(x)$.

6) Set $\mathbf{f}_e(x)$ as $\mathbf{f}(x)$. The iteration for edge e completes.

After completion of the above procedures, $I_t = \text{In}(t)$ for all $t \in T$, and $k_{d,e}(x) \in \mathcal{K}_\delta^{(x)}$ has been set for every adjacent pair (d, e) . ■

Proposition 9. In Algorithm 8, after every iteration of Step 5), $k_{d_i,e}(x)$ can always be selected from $\mathcal{K}_\delta^{(x)}$ subject to (15), and (14) always holds. In addition, when the iteration for an arbitrary edge completes, Eq. (13) always holds. Consequently, based on the constructed $(k_{d,e}(x)) \in \mathcal{K}_\delta^{(x)}$ by Algorithm 8, $(k_{d,e}(\alpha^r))$ forms a scalar linear solution over $\text{GF}(2^{m_L})$ for all $r \in R$.

Proof. Please refer to Appendix-E. □

We next theoretically analyze the computational complexity of Algorithm 8. In the initialization step, for each receiver $t \in T$, it takes $\mathcal{O}(|E|\omega)$ operations to establish \wp_t by the augmenting path approach. After initialization, Algorithm 8 traverses every edge exactly once. In every iteration to deal with an edge, by the standard Gauss elimination approach, every $\mathbf{w}_{t,j}$ in Step 4) can be computed by $\mathcal{O}(\omega^3)$ operations and there are at most $\frac{\phi(L)}{m_L}|T|$ possible $\mathbf{w}_{t,j}$ to be computed. In Step 5), it requires at most $|T|$ iterations and in each iteration: i) it takes $\mathcal{O}(|T|\omega)$ operations to compute the values in every \mathcal{A}_j , $1 \leq j \leq \frac{\phi(L)}{m_L}$; ii) it takes at most $\mathcal{O}(\frac{\phi(L)^2}{m_L^2}|T|)$ operations to select $k_{d_i,e}(x)$ from $\mathcal{K}_\delta^{(x)}$ prescribed by (15), where computing the evaluation of $k_{d_i,e}(x)$ at $x = \alpha^{rj}$ can be avoided by setting a mapping table from $\mathcal{K}_\delta^{(x)}$ to $\mathcal{K}_\delta^{(\alpha^j)}$ in advance. In summary, the computational complexity of the algorithm is $\mathcal{O}(\frac{\phi(L)}{m_L}|E||T|(\omega^3 + \omega|T| + \frac{\phi(L)}{m_L}|T|))$.

We remark that in Algorithm 8, the use of vectors $\mathbf{w}_{t,j}$ was motivated by [16]. Its goal is to facilitate the formulation of \mathcal{A}_j and selection of $k_{d_i,e}(x)$ subject to (15), so as to keep the invariant (13) after the process for every edge. To simplify the algorithm presentation, we assume that every $\mathbf{w}_{t,j}$ is computed independently in different iterations, with

computational complexity $\mathcal{O}(\omega^3)$. If we adopt a similar idea to the one proposed in [16] (refer to Fig. 4 therein), it is possible to compute $\mathbf{w}_{t,j}$ subject to (12) in a recursive manner, so that the corresponding computational complexity can be reduced from $\mathcal{O}(\omega^3)$ to $\mathcal{O}(\omega^2)$, and the total computational complexity of the algorithm will reduce to $\mathcal{O}(\frac{\phi(L)}{m_L}|E||T|(\omega^2 + \omega|T| + \frac{\phi(L)}{m_L}|T|))$. As such variation is not the focus of the present paper, we shall not elaborate it in details.

We end this section by listing some design instances of a $(\phi(L), L)$ circular-shift linear solution of degree δ .

- Assume L is prime with primitive root 2 and $1 \leq \delta \leq \frac{L-1}{2}$, so that $m_L = \phi(L) = L - 1$ and all elements in $\mathcal{K}_\delta^{(\alpha)}$ are distinct. Thus, $K_\delta = \sum_{j=0}^{\delta} \binom{L-1}{j}$. When $K_\delta > |T|$, an $(L - 1, L)$ circular-shift linear solution of degree δ can be efficiently constructed. This is the case considered in [4].
- Assume $\delta = 1$, so that $K_\delta = L + 1$. When $\frac{L+1}{L-1}m_L > |T|$, a $(\phi(L), L)$ circular-shift linear solution of degree δ can be efficiently constructed.
- Assume $\delta = m_L$, so that all elements in $\text{GF}(2^{m_L})$ are contained in $\mathcal{K}_\delta^{(\alpha)}$ and $K_\delta = 2^{m_L}$. When $\frac{m_p}{\phi(L)}2^{m_L} > |T|$, a $(\phi(L), L)$ circular-shift linear solution of degree δ can be efficiently constructed.
- Assume $L = p^l$, where p is an odd prime, so that $\phi(L) = p^l - p^{l-1}$ and $m_L = m_p p^{l-1}$. When $\frac{m_p}{p-1}K_\delta > |T|$, a $(p^l - p^{l-1}, p^l)$ circular-shift linear solution of degree δ can be efficiently constructed.

Note that since K_1 is always larger than $\phi(L)$, when $m_L \geq |T|$, a $(\phi(L), L)$ circular-shift linear solution of degree 1 can be efficiently constructed.

V. ASYMPTOTICAL LINEAR SOLVABILITY OF CIRCULAR-SHIFT LNC

As circular-shift LNC has been proven insufficient to achieve the exact multicast capacity of some multicast networks [4], whether every multicast network is *asymptotically circular-shift linearly solvable*, that is, for any $\epsilon > 0$, it has an (L', L) circular-shift linear solution with $L'/L > 1 - \epsilon$, becomes the fundamental problem for theoretical study of circular-shift LNC. For the case that L is a prime with primitive root 2, the efficient construction of an $(L - 1, L)$ circular-shift linear solution has been discussed [4]. However, whether there are infinitely many primes with primitive root 2 is still unknown (See, e.g., [17]), so whether every multicast network is asymptotically circular-shift linearly solvable remains open. As an application of Theorem 7, which applies to a general odd block length L , we are able to give an affirmative answer to this open problem.

Theorem 10. Every multicast network is asymptotically circular-shift linearly solvable.

Proof. Consider an arbitrary multicast network with the set T of receivers. Let ϵ be an arbitrary positive value, and write $M = \max\{\lceil \frac{1}{\epsilon} \rceil, |T|\}$. For an arbitrary positive integer m ,

denote by P_m the set of primes modulo which the multiplicative order of 2 is equal to m . As p divides $2^m - 1$ for each $p \in P_m$, P_m contains finitely many primes, and thus so does $\bigcup_{m < M} P_m$. As there are infinitely many primes, there must exist a prime L so that its multiplicative order $m_L \geq M$. For such L with $\phi(L) = L - 1$, according to Theorem 7, there exists an $(L - 1, L)$ circular-shift linear solution. In addition, as $L > m_L > \frac{1}{\epsilon}$, $\frac{L-1}{L} > 1 - \epsilon$. \square

APPENDIX

A. Proof of Theorem 1

According to the classical framework in [6], the global encoding kernels $\mathbf{f}_e(\alpha)$ incoming to t can be expressed as

$$[\mathbf{f}_e(\alpha)]_{e \in \text{In}(t)} = \mathbf{A}(\alpha) (\mathbf{I}_{|E|-\omega} + \mathbf{K}(\alpha) + \mathbf{K}(\alpha)^2 + \dots) \mathbf{B}(1).$$

Here $\mathbf{A}(\alpha)$ and $\mathbf{K}(\alpha)$ respectively stand for $[k_{d,e}(\alpha)]_{d \in \text{Out}(s), e \notin \text{Out}(s)}$ and $[k_{d,e}(\alpha)]_{d,e \notin \text{Out}(s)}$ for brevity, and $\mathbf{B}(1)$ is an $(|E| - \omega) \times |\text{In}(t)|$ index matrix of which the unique nonzero entry 1 in every column corresponds to an edge in $\text{In}(t)$. Via replacing α in $\mathbf{A}(\alpha)$, $\mathbf{K}(\alpha)$ by the cyclic permutation matrix \mathbf{C}_L , and replacing 1 in $\mathbf{B}(1)$ by \mathbf{I}_L , we have

$$[\mathbf{F}_e]_{e \in \text{In}(t)} = \mathbf{A}(\mathbf{C}_L) (\mathbf{I}_{(|E|-\omega)L} + \mathbf{K}(\mathbf{C}_L) + \mathbf{K}(\mathbf{C}_L)^2 + \dots) \mathbf{B}(\mathbf{I}_L).$$

Based on (1),

$$\begin{aligned} \mathbf{A}(\mathbf{C}_L) &= (\mathbf{I}_\omega \otimes \mathbf{V}_L) \cdot \mathbf{A}(\Lambda_\alpha) \cdot (\mathbf{I}_{|E|-\omega} \otimes \mathbf{V}_L^{-1}), \\ \mathbf{K}(\mathbf{C}_L)^j &= \mathbf{K}(\mathbf{V}_L \Lambda_\alpha \mathbf{V}_L^{-1})^j \\ &= (\mathbf{I}_{|E|-\omega} \otimes \mathbf{V}_L) \cdot \mathbf{K}(\Lambda_\alpha)^j \cdot (\mathbf{I}_{|E|-\omega} \otimes \mathbf{V}_L^{-1}), \\ \mathbf{B}_t(\mathbf{I}_L) &= (\mathbf{I}_{|E|-\omega} \otimes \mathbf{V}_L) \cdot \mathbf{B}_t(\mathbf{I}_L) \cdot (\mathbf{I}_\omega \otimes \mathbf{V}_L^{-1}), \end{aligned}$$

where \otimes represents the Kronecker product. Thus,

$$[\mathbf{F}_e]_{e \in \text{In}(t)} = (\mathbf{I}_\omega \otimes \mathbf{V}_L) \cdot \mathbf{M} \cdot (\mathbf{I}_\omega \otimes \mathbf{V}_L^{-1}), \quad (16)$$

where

$$\mathbf{M} = \mathbf{A}(\Lambda_\alpha) (\mathbf{I}_{(|E|-\omega)L} + \mathbf{K}(\Lambda_\alpha) + \mathbf{K}(\Lambda_\alpha)^2 + \dots) \mathbf{B}(\mathbf{I}_L).$$

As $(\mathbf{I}_\omega \otimes \mathbf{V}_L)$ and $(\mathbf{I}_\omega \otimes \mathbf{V}_L)^{-1}$ are full rank ωL ,

$$\text{rank}([\mathbf{F}_e]_{e \in \text{In}(t)}) = \text{rank}(\mathbf{M}).$$

Since all of $\mathbf{A}(\Lambda_\alpha)$, $\mathbf{K}(\Lambda_\alpha)^j$, and $\mathbf{B}(\mathbf{I}_L)$ can be regarded as a block matrix with every block entry to be an $L \times L$ diagonal matrix, so is \mathbf{M} . Thus, we can rearrange the rows and columns in \mathbf{M} to form a new matrix $\tilde{\mathbf{M}}$ as follows. Let \mathbf{P} denote the $\omega L \times \omega L$ permutation matrix that can be written in the block form $\begin{bmatrix} \mathbf{J}_{1,1} & \dots & \mathbf{J}_{1,\omega} \\ \vdots & \ddots & \vdots \\ \mathbf{J}_{L,1} & \dots & \mathbf{J}_{L,\omega} \end{bmatrix}$, where every block $\mathbf{J}_{i,j}$, $1 \leq i \leq L$, $1 \leq j \leq \omega$, is an $\omega \times L$ matrix with the only nonzero entry 1 located at row j and column i . Set $\tilde{\mathbf{M}} = \mathbf{PMP}^T$. It can be checked that

$$\tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{M}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{L-1} \end{bmatrix}, \quad (17)$$

where

$$\mathbf{M}_j = \mathbf{A}(\alpha^j) (\mathbf{I}_{|E|-\omega} + \mathbf{K}(\alpha^j) + \mathbf{K}(\alpha^j)^2 + \dots) \mathbf{B}(1). \quad (18)$$

Under the expression in (18), it turns out that

$$\mathbf{M}_j = [\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)}. \quad (19)$$

So we have

$$\begin{aligned} \text{rank}([\mathbf{F}_e]_{e \in \text{In}(t)}) &= \text{rank}(\mathbf{M}) \\ &= \text{rank}(\tilde{\mathbf{M}}) \\ &= \sum_{j=0}^{L-1} \text{rank}([\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)}). \end{aligned}$$

■

B. Proof of Lemma 3

Denote by $\tilde{\mathbf{I}}_J$ the $J \times L$ matrix obtained from \mathbf{I}_L by restricting to the first J rows. Thus,

$$\tilde{\mathbf{V}} = \tilde{\mathbf{I}}_J \mathbf{V}_L \tilde{\mathbf{I}}_J^T. \quad (20)$$

Based on (3) and (5), it can be easily seen that \mathbf{V}_L^{-1} is a column permutation of \mathbf{V}_L . Thus, in order to show that in order to show every entry in $\mathbf{G} = \tilde{\mathbf{V}}^{-1} \tilde{\mathbf{I}}_J \mathbf{V}_L^{-1}$ is in $\text{GF}(2)$, it is equivalent to show that every entry in $\tilde{\mathbf{V}}^{-1} \tilde{\mathbf{I}}_J \mathbf{V}_L$ is in $\text{GF}(2)$.

As a complement of $\tilde{\mathbf{I}}_J$, denote by $\tilde{\mathbf{I}}_{\mathcal{J}^c}$ the $(L - J) \times L$ matrix obtained from \mathbf{I}_L by deleting the $(j+1)^{\text{st}}$ row whenever $j \in \mathcal{J}$. Since $\tilde{\mathbf{I}}_J^T \tilde{\mathbf{I}}_J + \tilde{\mathbf{I}}_{\mathcal{J}^c}^T \tilde{\mathbf{I}}_{\mathcal{J}^c} = \mathbf{I}_L$,

$$\mathbf{V}_L^{-1} (\tilde{\mathbf{I}}_J^T \tilde{\mathbf{I}}_J + \tilde{\mathbf{I}}_{\mathcal{J}^c}^T \tilde{\mathbf{I}}_{\mathcal{J}^c}) \mathbf{V}_L = \mathbf{I}_L.$$

In addition, since $\tilde{\mathbf{I}}_J \tilde{\mathbf{I}}_J^T = \mathbf{I}_J$,

$$\begin{aligned} &\tilde{\mathbf{I}}_J \mathbf{V}_L^{-1} \tilde{\mathbf{I}}_J^T \tilde{\mathbf{I}}_J \mathbf{V}_L \tilde{\mathbf{I}}_J^T + \tilde{\mathbf{I}}_J \mathbf{V}_L^{-1} \tilde{\mathbf{I}}_{\mathcal{J}^c}^T \tilde{\mathbf{I}}_{\mathcal{J}^c} \mathbf{V}_L \tilde{\mathbf{I}}_J^T \\ &= \tilde{\mathbf{I}}_J (\mathbf{V}_L^{-1} (\tilde{\mathbf{I}}_J^T \tilde{\mathbf{I}}_J + \tilde{\mathbf{I}}_{\mathcal{J}^c}^T \tilde{\mathbf{I}}_{\mathcal{J}^c}) \mathbf{V}_L) \tilde{\mathbf{I}}_J^T \\ &= \mathbf{I}_J. \end{aligned} \quad (21)$$

For simplicity, write

$$\mathbf{U}_1 = \tilde{\mathbf{I}}_J \mathbf{V}_L^{-1} \tilde{\mathbf{I}}_J^T, \quad \mathbf{U}_2 = \tilde{\mathbf{I}}_J \mathbf{V}_L^{-1} \tilde{\mathbf{I}}_{\mathcal{J}^c}^T \tilde{\mathbf{I}}_{\mathcal{J}^c} \mathbf{V}_L \tilde{\mathbf{I}}_J^T.$$

Together with (20), Eq. (21) can be written as $\mathbf{U}_1 \tilde{\mathbf{V}} + \mathbf{U}_2 = \mathbf{I}_J$. Because \mathbf{U}_1^T and $\tilde{\mathbf{V}}$ can be respectively regarded as a Vandermonde matrix generated by α^{-j} , $j \in \mathcal{J}$ and by α^j , $j \in \mathcal{J}$, they are invertible, and so is $\mathbf{I}_J + \mathbf{U}_2$. Thus,

$$\tilde{\mathbf{V}}^{-1} = (\mathbf{I}_J + \mathbf{U}_2)^{-1} \mathbf{U}_1,$$

and

$$\tilde{\mathbf{V}}^{-1} \tilde{\mathbf{I}}_J \mathbf{V}_L = (\mathbf{I}_J + \mathbf{U}_2)^{-1} (\tilde{\mathbf{I}}_J \mathbf{V}_L^{-1} \tilde{\mathbf{I}}_J^T) \tilde{\mathbf{I}}_J \mathbf{V}_L.$$

As the inverse of a matrix over $\text{GF}(2)$ is also over $\text{GF}(2)$, it turns out that in order to show that $\tilde{\mathbf{V}}^{-1} \tilde{\mathbf{I}}_J \mathbf{V}_L$ is a matrix over $\text{GF}(2)$, it suffices to show that both $\mathbf{V}_L^{-1} \tilde{\mathbf{I}}_J^T \tilde{\mathbf{I}}_J \mathbf{V}_L$ and $\mathbf{V}_L^{-1} \tilde{\mathbf{I}}_{\mathcal{J}^c}^T \tilde{\mathbf{I}}_{\mathcal{J}^c} \mathbf{V}_L (= \tilde{\mathbf{I}}_J^T \mathbf{U}_2 \tilde{\mathbf{I}}_J)$ are over $\text{GF}(2)$.

Among integers $0, 1, \dots, L - 1$, label the ones in \mathcal{J} as j_1, \dots, j_J , and the ones not in \mathcal{J} as j_{J+1}, \dots, j_L , both in an ascending order. We have

$$\mathbf{V}_L^{-1} \tilde{\mathbf{I}}_J^T = \begin{bmatrix} 1 \\ \alpha^{-j_1} \\ \alpha^{-2j_1} \\ \vdots \\ \alpha^{-(L-1)j_1} \end{bmatrix}_{1 \leq l \leq J}, \quad \tilde{\mathbf{I}}_J \mathbf{V}_L = \begin{bmatrix} 1 \\ \alpha^{j_1} \\ \alpha^{2j_1} \\ \vdots \\ \alpha^{(L-1)j_1} \end{bmatrix}_{1 \leq l \leq J}^T.$$

Thus, for all $1 \leq l_1, l_2 \leq L$, the $(l_1, l_2)^{th}$ entry in $\mathbf{V}_L^{-1} \tilde{\mathbf{I}}_{\mathcal{J}}^T \tilde{\mathbf{I}}_{\mathcal{J}} \mathbf{V}_L$ can be written as

$$\sum_{l=1}^J \alpha^{-l_1 j l} \alpha^{l_2 j l} = \sum_{l=1}^J \alpha^{(l_2 - l_1) j l}.$$

Because $\mathcal{J} = \{j_1, \dots, j_J\}$ is closed under multiplication by 2 modulo L , we have

$$\left(\sum_{l=1}^J \alpha^{(l_2 - l_1) j l} \right)^2 = \sum_{l=1}^J \alpha^{2(l_2 - l_1) j l} = \sum_{l=1}^J \alpha^{(l_2 - l_1) j l},$$

and so $\sum_{l=1}^J \alpha^{(l_2 - l_1) j l} \in \text{GF}(2)$. Similarly, for all $1 \leq l_1, l_2 \leq L$, the $(l_1, l_2)^{th}$ entry in $\mathbf{V}_L^{-1} \tilde{\mathbf{I}}_{\mathcal{J}^c}^T \tilde{\mathbf{I}}_{\mathcal{J}^c} \mathbf{V}_L$ is $\sum_{l=J+1}^L \alpha^{(l_2 - l_1) j l}$. Since both $\{0, 1, \dots, L\}$ and \mathcal{J} are closed under multiplication by 2 modulo L , $\{j_{J+1}, j_{J+2}, \dots, j_L\} = \{0, 1, \dots, L\} \setminus \mathcal{J}$ is also closed under multiplication by 2 modulo L . Thus, $\left(\sum_{l=J+1}^L \alpha^{(l_2 - l_1) j l} \right)^2 = \sum_{l=J+1}^L \alpha^{(l_2 - l_1) j l}$ and so $\sum_{l=J+1}^L \alpha^{(l_2 - l_1) j l} \in \text{GF}(2)$. ■

C. Proof of Theorem 4

It remains to prove (9). Follow the same argument as in the proof of Theorem 1 (refer to Appendix-A) till Eq. (19). For a receiver t , by (16), (17) and (19), we have

$$[\mathbf{F}_e]_{e \in \text{In}(t)} = (\mathbf{I}_\omega \otimes \mathbf{V}_L) \mathbf{P}^T \tilde{\mathbf{M}} \mathbf{P} (\mathbf{I}_\omega \otimes \mathbf{V}_L^{-1}),$$

where $\tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{M}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{L-1} \end{bmatrix}$, and $\mathbf{M}_j = [\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)}$.

By (8), $\mathbf{G}_s = \mathbf{I}_\omega \otimes (\tilde{\mathbf{V}}^{-1} \tilde{\mathbf{I}}_{\mathcal{J}} \mathbf{V}_L^{-1})$. Similar to the definition of the permutation matrix \mathbf{P} , define \mathbf{Q} as the $\omega J \times \omega J$ permutation matrix that can be written in the block form $\begin{bmatrix} \mathbf{J}_{1,1} & \dots & \mathbf{J}_{1,\omega} \\ \vdots & \ddots & \vdots \\ \mathbf{J}_{J,1} & \dots & \mathbf{J}_{J,\omega} \end{bmatrix}$, where every block $\mathbf{J}_{i,j}$, $1 \leq i \leq J$, $1 \leq j \leq \omega$, is an $\omega \times J$ matrix with the only nonzero entry 1 located at row j and column i . As $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_{\omega J}$,

$$\mathbf{G}_s = \mathbf{I}_\omega \otimes (\tilde{\mathbf{V}}^{-1} \tilde{\mathbf{I}}_{\mathcal{J}} \mathbf{V}_L^{-1}) = (\mathbf{I}_\omega \otimes \tilde{\mathbf{V}}^{-1}) \mathbf{Q}^T \mathbf{Q} (\mathbf{I}_\omega \otimes (\tilde{\mathbf{I}}_{\mathcal{J}} \mathbf{V}_L^{-1})).$$

Since the square matrix $(\mathbf{I}_\omega \otimes \tilde{\mathbf{V}}^{-1}) \mathbf{Q}^T$ is full rank ωJ and the square matrix $\mathbf{P} (\mathbf{I}_\omega \otimes \mathbf{V}_L^{-1})$ is full rank ωL ,

$$\begin{aligned} & \text{rank}(\mathbf{G}_s [\mathbf{F}_e]_{e \in \text{In}(t)}) \\ &= \text{rank} \left(\mathbf{Q} (\mathbf{I}_\omega \otimes (\tilde{\mathbf{I}}_{\mathcal{J}} \mathbf{V}_L^{-1})) (\mathbf{I}_\omega \otimes \mathbf{V}_L) \mathbf{P}^T \tilde{\mathbf{M}} \right) \\ &= \text{rank} \left(\mathbf{Q} (\mathbf{I}_\omega \otimes \tilde{\mathbf{I}}_{\mathcal{J}}) \mathbf{P}^T \tilde{\mathbf{M}} \right) \\ &= \text{rank} \left((\tilde{\mathbf{I}}_{\mathcal{J}} \otimes \mathbf{I}_\omega) \tilde{\mathbf{M}} \right) \\ &= \text{rank} \left(\sum_{j \in \mathcal{J}} [\mathbf{f}_e(\alpha^j)]_{e \in \text{In}(t)} \right). \end{aligned}$$

D. Proof of Lemma 6

According to Lemma 5, the cyclotomic polynomial $Q_L(x) = f_1(x) \cdots f_{\phi(L)/m_L}(x)$, where $f_j(x)$, $1 \leq j \leq d$, is an irreducible polynomial over $\text{GF}(2)$ of degree m_L . Thus, for every $f_j(x)$, the exponents of the m_L roots, expressed as powers of α , constitute a cyclotomic coset $\{r, 2r, \dots, 2^{m_L-1}r\}$

modulo L for some $r \in R$, and R can be partitioned into $\frac{\phi(L)}{m_L}$ distinct cyclotomic cosets modulo L .

Denote by C_j , $1 \leq j \leq \frac{\phi(L)}{m_L}$, the cyclotomic cosets modulo L such that $f_j(x) = \prod_{r \in C_j} (x - \alpha^r)$, and by r_j an arbitrary entry in C_j . If there exist $k_1(x), k_2(x), \dots, k_n(x) \in \mathcal{K}_\delta^{(x)}$ subject to $g(k_1(\alpha^{r_j}), \dots, k_n(\alpha^{r_j})) \neq 0$ for some $1 \leq j \leq d$, then for every $r' \in C_j$, which can be written as $2^l r_j$ modulo L for some $1 \leq l \leq m_L$,

$$\begin{aligned} g(k_1(\alpha^{r'}), \dots, k_n(\alpha^{r'})) &= g(k_1(\alpha^{2^l r_j}), \dots, k_n(\alpha^{2^l r_j})) \\ &= g(k_1(\alpha^{r_j})^{2^l}, \dots, k_n(\alpha^{r_j})^{2^l}) \\ &= g(k_1(\alpha^{r_j}), \dots, k_n(\alpha^{r_j}))^{2^l} \neq 0. \end{aligned}$$

In addition, as $R = \bigcup_{1 \leq j \leq \frac{\phi(L)}{m_L}} C_j$, in order to show the lemma, it suffices to show the existence of $k_1(x), k_2(x), \dots, k_n(x) \in \mathcal{K}_\delta^{(x)}$ such that

$$\prod_{1 \leq j \leq \frac{\phi(L)}{m_L}} g(k_1(\alpha^{r_j}), k_2(\alpha^{r_j}), \dots, k_n(\alpha^{r_j})) \neq 0. \quad (22)$$

Note that for each $r \in R$, as r is coprime with L , we have $\{\alpha, \alpha^2, \dots, \alpha^L\} = \{\alpha^r, \alpha^{2r}, \dots, \alpha^{Lr}\}$. Hence, the mapping $\psi : \mathcal{K}_\delta^{(\alpha)} \rightarrow \mathcal{K}_\delta^{(\alpha^r)}$ defined by $\psi(k(\alpha)) = k(\alpha^r)$ is a bijection, and $\mathcal{K}_\delta^{(\alpha)} = \mathcal{K}_\delta^{(\alpha^r)}$. Consequently, there exist $k_1(\alpha^r), k_2(\alpha^r), \dots, k_n(\alpha^r) \in \mathcal{K}_\delta^{(\alpha^r)}$ with $g(k_1(\alpha^r), k_2(\alpha^r), \dots, k_n(\alpha^r)) = 0$ if and only if there exists $k'_1(\alpha), k'_2(\alpha), \dots, k'_n(\alpha) \in \mathcal{K}_\delta^{(\alpha)}$ with $g(k'_1(\alpha), k'_2(\alpha), \dots, k'_n(\alpha)) = 0$.

As $g(x_1, \dots, x_n)$ has degree at most D in every x_j , by the Schwartz-Zippel lemma (See, e.g., [14]), it has at most DK_δ^{n-1} roots over $\mathcal{K}_\delta^{(\alpha)}$, where $\mathcal{K}_\delta^{(\alpha^i)}$ refers to the set containing all different elements in $\mathcal{K}_\delta^{(\alpha^i)}$ for $0 \leq i \leq L-1$. Since $\mathcal{K}_\delta^{(\alpha^{r_j})} = \mathcal{K}_\delta^{(\alpha)}$ and $|\mathcal{K}_\delta^{(\alpha^{r_j})}| = K_\delta$ for all $1 \leq j \leq \frac{\phi(L)}{m_L}$, by taking a union bound, we conclude that there are at most $\frac{\phi(L)}{m_L} DK_\delta^{n-1}$ possible choices of $k_1(\alpha), k_2(\alpha), \dots, k_n(\alpha) \in \mathcal{K}_\delta^{(\alpha)}$, where $k_i(\alpha)$ is the evaluation of some $k_i(x) \in \mathcal{K}_\delta^{(x)}$ by setting $x = \alpha$, such that

$$\prod_{1 \leq j \leq \frac{\phi(L)}{m_L}} g(k_1(\alpha^{r_j}), k_2(\alpha^{r_j}), \dots, k_n(\alpha^{r_j})) = 0.$$

Consequently, when $K_\delta > \frac{\phi(L)}{m_L} D$, i.e., $\frac{m_L K_\delta}{\phi(L)} > D$, there must exist $k_1(x), k_2(x), \dots, k_n(x) \in \mathcal{K}_\delta^{(x)}$ satisfying $g(k_1(\alpha^{r_j}), k_2(\alpha^{r_j}), \dots, k_n(\alpha^{r_j})) \neq 0$ for all $1 \leq j \leq \frac{\phi(L)}{m_L}$, i.e., Eq. (22) obeys. ■

E. Proof of Proposition 9

For $1 \leq j \leq \frac{\phi(L)}{m_L}$, by definition, $k_{d,e}(\alpha^{r_j}) \in \mathcal{K}_\delta^{(\alpha^{r_j})}$. Denote by $\mathcal{K}_\delta^{(\alpha^{r_j})}$ the set containing all different elements in $\mathcal{K}_\delta^{(\alpha^{r_j})}$. Assume that the algorithm is dealing with edge $e \in \text{Out}(v)$ for some non-source node v and iteration $1 \leq i \leq l$ in Step 5). Due to the definition of \mathcal{A}_j and the exclusion of the case that $\mathbf{f}(\alpha^{r_j})^T \mathbf{w}_{t,j} \neq 0$ for all $1 \leq j \leq \frac{\phi(L)}{m_L}$ and $t \in \bigcup_{1 \leq i' \leq i} T_{d,i'}$, $|\mathcal{A}_j| \leq |T|$ and $\frac{\phi(L)}{m_L} |\mathcal{A}_j| \leq \frac{\phi(L)}{m_L} |T| - 1$.

Hence, there are at most $|T|$ nonzero values in $\mathcal{K}_\delta^{(\alpha^{r_j})}$ whose multiplicative inverses belong to \mathcal{A}_j . It has been argued in the proof of Lemma 6 that $\mathcal{K}_\delta^{(\alpha)} = \mathcal{K}_\delta^{(\alpha^{r_j})}$. Under the union bound, there are at most $\frac{\phi(L)}{m_L}|T|$ elements in the set

$$\{k(\alpha) : k(x) \in \mathcal{K}_\delta^{(x)}, \text{ either } k(\alpha^{r_j}) = 0 \text{ or } k(\alpha^{r_j})^{-1} \in \mathcal{A}_j \\ \text{for some } 1 \leq j \leq \phi(L)/m_L\}.$$

As it is assumed at the beginning of Algorithm 8 that $\frac{m_L}{\phi(L)}K_\delta > |T|$, i.e., $K_\delta (= |\mathcal{K}_\delta^{(\alpha)}|) > \frac{\phi(L)}{m_L}|T|$, there must exist $k_{d_i,e}(x) \in \mathcal{K}_\delta^{(x)}$ such that (15) holds for all $1 \leq j \leq \frac{\phi(L)}{m_L}$.

We next show that condition (14) always holds after iteration i in Step 5). When $i = 1$, it obviously holds because of $\mathbf{f}(x) = \mathbf{f}_{d_i}(x)$ and (12). Inductively, assume (14) holds up to iteration $i - 1$, where $1 < i \leq l$ and $\mathbf{f}(x)$ is obtained after iteration $i - 1$. After iteration i , consider an arbitrary $1 \leq j \leq \frac{\phi(L)}{m_L}$. Since $k_{d_i,e}(\alpha^{r_j}) \neq 0$ and $k_{d_i,e}(\alpha^{r_j})^{-1} \notin \mathcal{A}_j$ by condition (15), for either the case $t \in T_{d_i}$ and $\mathbf{f}(\alpha^{r_j})^T \mathbf{w}_{t,j} \neq 0$ or the case $t \in \bigcup_{1 \leq i' < i} T_{d_{i'}}$,

$$\begin{aligned} & (k_{d_i,e}(\alpha^{r_j})^{-1} \mathbf{f}(\alpha^{r_j}) + \mathbf{f}_{d_i}(\alpha^{r_j}))^T \mathbf{w}_{t,j} \\ &= k_{d_i,e}(\alpha^{r_j})^{-1} \mathbf{f}(\alpha^{r_j})^T \mathbf{w}_{t,j} + \mathbf{f}_{d_i}(\alpha^{r_j})^T \mathbf{w}_{t,j} \neq 0. \end{aligned}$$

In addition, for the case $t \in T_{d_i}$ and $\mathbf{f}(\alpha^{r_j})^T \mathbf{w}_{t,j} = 0$, condition (12) implies

$$(\mathbf{f}(\alpha^{r_j}) + k_{d_i,e}(\alpha^{r_j}) \mathbf{f}_{d_i}(\alpha^{r_j}))^T \mathbf{w}_{t,j} = k_{d_i,e}(\alpha^{r_j}) \neq 0.$$

We have thus verified that when $\mathbf{f}(x)$ is replaced by $\mathbf{f}(x) + k_{d_i,e}(x) \mathbf{f}_{d_i}(x)$ after iteration i , (14) always holds. Lemma 5 in [16] then applies here to assert that $\mathbf{f}(\alpha^{r_j})$ is linearly independent of $\mathbf{f}_{d'}(\alpha^{r_j})$, $d' \in I_t \setminus \{e\}$.

We can now conclude that after $\mathbf{f}_e(x)$ is set as $\mathbf{f}(x)$ in Step 6), for every receiver $t \in \bigcup_{1 \leq i \leq l} T_{d_i}$, $\mathbf{f}_e(\alpha^{r_j})$ is linearly independent of $\{\mathbf{f}_{e'}(\alpha^{r_j}) : e' \in I_t \setminus \{e\}\}$, i.e., $\text{rank}([\mathbf{f}_{e'}(\alpha^{r_j})]_{e' \in I_t}) = \omega$, and hence (13) holds.

Since the algorithm terminates with $I_t = \text{In}(t)$ for all $t \in T$, $(k_{d,e}(\alpha^{r_j}))$ forms a scalar linear solution over $\text{GF}(2^{m_L})$ for all $1 \leq j \leq \frac{\phi(L)}{m_L}$. By Lemma 2, $(k_{d,e}(\alpha^{r_j}))$ forming a scalar linear solution implies that $(k_{d,e}(\alpha^r))$ forms a scalar linear solution for all $r \in C_j$, where C_j represents the cyclotomic coset modulo L containing r_j . As $R = \bigcup_{1 \leq j \leq \frac{\phi(L)}{m_L}} C_j$, $(k_{d,e}(\alpha^r))$ forms a scalar linear solution for all $r \in R$. ■

REFERENCES

- [1] M. Xiao, M. Médard, and T. Aulin, "A binary coding approach for combination networks and general erasure networks," *IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2007.
- [2] A. Keshavarz-Haddad and M. A. Khojastepour, "Rotate-and-add coding: A novel algebraic network coding scheme," *IEEE ITW*, Ireland, 2010.
- [3] H. Hou, K. W. Shum, M. Chen and H. Li, "BASIC codes: low-complexity regenerating codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3053-3069, Jun. 2016.
- [4] H. Tang, Q. T. Sun, Z. Li, X. Yang, and K. Long, "Circular-shift linear network coding," *IEEE Trans. Inform. Theory*, to be published. [Online]. Available: <https://ieeexplore.ieee.org/document/8353910/>, doi: 10.1109/TIT.2018.2832624.
- [5] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, Feb. 2003.

- [6] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, No. 5, Oct. 2003.
- [7] M. Médard, M. Effros, D. Karger, and T. Ho, "On coding for non-multicast networks," *Annual ALLERTON Conference*, 2003.
- [8] J. B. Ebrahimi and C. Fragouli, "Algebraic algorithm for vector network coding" *IEEE Trans. Inf. Theory*, vol. 57, no. 2, Feb. 2011.
- [9] L. Zhang, Q. Huang, S. Lin, K. Abdel-Ghaffar, and I. F. Blake, "Quasi-cyclic LDPC codes: an algebraic construction, rank analysis, and codes on latin squares," *IEEE Trans. Commun.*, vol. 58, no. 11, 2010.
- [10] Q. Diao, Q. Huang, S. Lin and K. Abdel-Ghaffar, "Cyclic and quasi-cyclic LDPC codes on constrained parity-check matrices and their trapping sets," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, 2012.
- [11] Y. Xie, J. Yuan, and Q. T. Sun, "Protograph based quantum LDPC codes from quadratic residue sets," *IEEE Trans. Commun.*, to be published. [Online]. Available: <https://ieeexplore.ieee.org/document/8340056/>, doi:10.1109/TCOMM.2018.2827945.
- [12] Q. T. Sun, X. Yang, K. Long, X. Yin, and Z. Li, "On vector linear solvability of multicast networks," *IEEE Trans. Comm.*, Dec. 2016.
- [13] R. Lidl and H. Niederreiter, *Finite Fields*, 2nd ed. Cambridge University Press, 1997.
- [14] S. Jukna, *Extremal Combinatorics: With Applications in Computer Science*, 2nd ed. Berlin, Germany: Springer-Verlag, 2011.
- [15] T. Ho, D. Karger, M. Médard, R. Koetter, "Network coding from a network flow perspective," *IEEE Int. Symp. Inf. Theory*, Yokohama, Japan, Jun. 2003.
- [16] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, Jun. 2005.
- [17] N. J. A. Sloane, "Primes with primitive root 2," *The On-Line Encyclopedia of Integer Sequences*, <https://oeis.org/A001122>.