



**HAL**  
open science

# Network Coding with Random Packet-Index Assignment for Mobile Crowdsensing

Cédric Adjih, Michel Kieffer, Claudio Greco

► **To cite this version:**

Cédric Adjih, Michel Kieffer, Claudio Greco. Network Coding with Random Packet-Index Assignment for Mobile Crowdsensing. 2017. hal-01538115v1

**HAL Id: hal-01538115**

**<https://centralesupelec.hal.science/hal-01538115v1>**

Preprint submitted on 13 Jun 2017 (v1), last revised 28 Dec 2018 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Network Coding with Random Packet-Index Assignment for Mobile Crowdsensing

Cedric Adjih<sup>\*</sup>, Michel Kieffer<sup>†</sup>, and Claudio Greco<sup>\*†‡</sup>

<sup>\*</sup>INRIA Saclay, INFINE Project Team, 1 rue Honoré d'Estienne d'Orves,  
91120 Palaiseau, France

<sup>†</sup>L2S, CNRS-Supélec-Univ Paris-Sud, 3 rue Joliot-Curie, 91192  
Gif-sur-Yvette, France

<sup>‡</sup>Institut Mines-Télécom, Télécom ParisTech, 46 rue Barrault, 75634 Paris  
Cedex 13, France

## Abstract

The proliferation of mobile devices, such as smartphones or connected objects with rich sensing capabilities, has given rise to a new fast-growing sensing paradigm: *mobile crowdsensing*. Mobile crowdsensing (MCS) takes advantage of the ubiquity of the devices to process and collect information through voluntary sensing.

This paper focuses on one specific issue: decentralized data collection via MCS. While numerous techniques from managed networks can be adapted, one of the most efficient (from the energy and spectrum use perspective) is *network coding* (NC). NC is well suited to networks with mobility and unreliability, however, practical NC requires a precise identification of individual packets that have been mixed together. In a purely decentralized system, this requires either conveying identifiers in headers along with coded information, or integrating a more complex protocol in order to efficiently identify the sources (participants) and their payloads.

This paper presents a novel solution, Network Coding with Random Packet Index Assignment (NeCoRPIA), where packet indices in NC headers are selected in a decentralized way, by simply choosing them randomly. Traditional network decoding techniques apply directly when all original packets have different indices. When this is not the case, *i.e.*, in case of collisions of indices, a specific decoding algorithm is proposed. A theoretical analysis of its performance in terms of complexity and decoding error probability is described. Simulation results match well the theoretical results.

Using NeCoRPIA, NC may be performed without coordination between agents. Compared to classical NC, generations of 60 packets may be considered with a header overhead of 66 %, a

vanishing decoding error probability, and a decoding complexity about 10 times that of Gaussian elimination.

### Index Terms

Network coding, random source index, mobile crowdsensing, broadcast, data collection.

## I. INTRODUCTION

Currently, smartphones, as well as various connected objects with advanced connectivity and sensing capabilities are more and more widespread. This trend leads to the emergence of a novel category of applications, termed *mobile crowdsensing* (MCS) [1] where individual users or devices participate voluntarily to gather collectively environmental information. An example of practical experimentation of MCS is given by Nericell [2] in Bangalore where road and traffic conditions (including potholes, braking, and honking) were monitored through mobile phones. One of the challenges that has to be addressed is the distributed nature of the network with unpredictable stability, high churn rate, and node mobility. In this paper, we focus on scenarios of MCS where local measurements from some area are collected through multi-hop communication (*e.g.*, through WiFi or D2D communication [3], saving energy and cellular spectrum use) as represented later in Figure 1.

For these scenarios, a suitable communication technique is *Network Coding* (NC) [4]. NC is a transmission paradigm for multi-hop networks, in which, rather than merely relaying packets, the intermediate nodes may *mix* the packets they receive. In general, in wireless networks, the inherent broadcast capacity of the channel provides added benefits and improves the communication efficiency [5], [6], [7]. Moreover, some techniques permit to implement it in a more distributed manner [8], [9] fitting mobile and dynamic network requirements.

Although NC is ideally suited for the considered MCS applications, in practical NC (PNC) protocols, the mixing of the packets is achieved through linear combinations, and the corresponding coding coefficients are generally included in each mixed packet, either as an *encoding vector* [9], or in some compressed form [10], [11] (see Figure 4). In this way, a coefficient can be uniquely associated with the corresponding packet: the downside is the requirement for a global indexing of all packets in the network. Such global indexing is difficult to perform in a purely distributed system, where sources may appear, move, and disappear.

The major contribution of this paper is a novel approach that allows one to overcome this major hurdle: we propose Network Coding with Random Packet-Index Assignment (NeCoRPIA). With NeCoRPIA, packet indices in NC headers are selected in a decentralized way, by simply choosing them randomly. A preliminary version of this idea was first presented in [12]. Such random assignment is simple, fast, and fully distributed but does present the possibility of *collisions*, that is, two packets being assigned to the same index by different nodes. Here, we extend the idea in [12], by considering a random assignment of several indexes to each packet. This significantly reduces the probability of collision compared to a single index, even if represented on the same number of bits as several indexes. Since collisions cannot be totally avoided, we also propose an elaborate algorithm to decode the received packets in spite of possible collisions, also taking advantage of the content of the crowdsensing data packets themselves. We additionally detail how this approach is encompassed in a practical MCS NC protocol.

This paper reviews related work in Section II. Section III presents the architecture and protocol of NeCoRPIA, our version of PNC dedicated to MCS. Section IV describes network decoding techniques within the NeCoRPIA framework. Section V analyses the complexity of the proposed approach. Section VI evaluates the performance of NeCoRPIA. Finally, Section VII provides some conclusions and future research directions.

## II. RELATED WORK

The generic problem of efficiently collecting information from multiple sources to one or several sinks in multi-hop networks has been extensively studied in the literature: for instance, for static deployments of wireless sensor networks, a routing protocol such as the Collection Tree Protocol (CTP) [13] is typical. Focusing on our envisioned MCS applications [1], performance can be improved by the use of NC, as exemplified by [5] where NC is shown to outperform routing in a scenario with multiple sources (all-to-all broadcast). NC for broadcast is even proved to be asymptotically optimal [14].

Motivated by performance gains, a large number of PNC protocols have been proposed for wireless multi-hop networks: most of them adopt random linear coding [8] and the key proposal of prepending a global encoding vector to coded payloads [9] (see top of Figure 4).

A practical concern is the overhead brought by the encoding vector: [15] explores the trade-off between field size and generation (hence encoding vector) size. Two examples of different approaches aim at reducing its size: in [10], encoding vectors are themselves *coded*

(compressed) yielding gains when only limited subsets of linear combinations are possible, whereas in [11] a special coding scheme permits to represent it with one single symbol at the expense of generation size.

In the case of multiple sources, constructing an efficient encoding vector is more difficult as evidenced by COPE [6], one of the first NC implementations for a testbed: it resorts to a header that explicitly includes one identifier (32-bit hash) for each of the coded packets (generalized in Figure 4).

Going further, the encoding vectors may be entirely removed from the header of packets. Network decoding may then be seen as a *source separation* problem: recovering source vectors defined over a field (*e.g.*,  $\mathbb{R}$  or  $\mathbb{C}$ ) from observations linearly combined through an *unknown* encoding matrix [16]. In  $\mathbb{R}$ , one classical approach to source separation is *Independent Component Analysis* (ICA) [17], [18], which estimates the sources as the set of linear combinations that minimizes the joint entropy and the mutual similarity among vectors. ICA can still be used over a finite field  $\mathbb{F}_q$  [19], [20] by minimizing the marginal entropy of the estimated sources. In previous work, we have proposed different techniques that exploit ICA over finite fields in the context of NC: in [21], network-coded packets without encoding vectors are decoded using entropy minimization jointly with channel encoding, while in [22], we exploit the redundancy introduced by communication protocols to assist the receiver in decoding, via the solution of a maximum *a posteriori* (MAP) estimation problem. The price to pay is a significantly larger decoding complexity compared to simple Gaussian elimination when considering classical NC.

In a preliminary version of this paper [12], a simple random encoding vector was considered. This vector had to be quite large for generations with a typical number of packets in order to avoid collisions. This paper extends the idea of [12] by considering a random encoding vector split into  $n_v$  independent random subvectors. For a given level of performance (collision probability), this leads to a significant decrease of the size of the encoding vector, and thus reduces the NC overhead.

### III. NECORPIA ARCHITECTURE AND PROTOCOL

#### A. Mobile Crowdsensing Application Objective

An MCS architecture is considered where a set  $\mathcal{S} = \{S_1, \dots, S_N\}$  of  $N$  static nodes gathering measurements performed by mobile sensing nodes. Each node  $S_i$ , located in  $\theta_i$ , in some reference frame  $\mathcal{F}$ , acts as a data collection point for all mobile nodes located in its

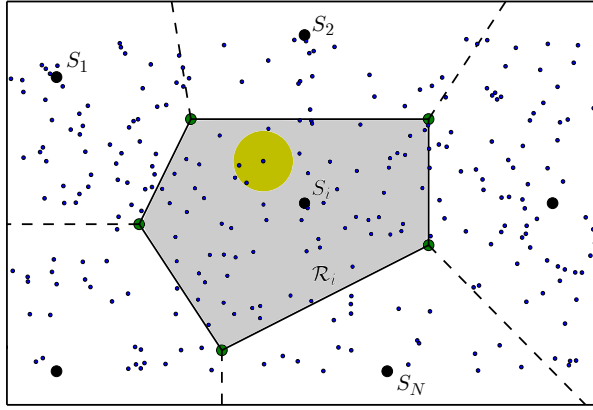


Figure 1. MCS data collection scenario: the static node  $S_i$  gathers measurements from all mobile nodes (with limited communication range) in its assigned data collection area  $\mathcal{R}_i$  in grey

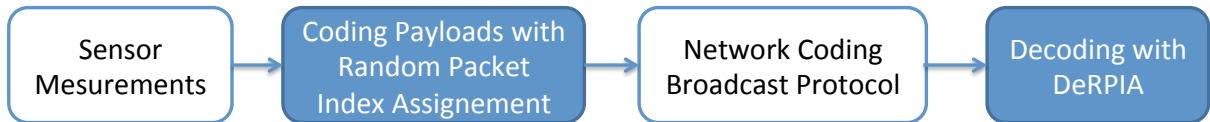


Figure 2. Modules of NeCoRPIA architecture

assigned area  $\mathcal{R}_i$  (e.g., its Voronoi cell) as in Figure 1. The data consist, for instance, in a set of measurements of some physical quantity  $D(\xi)$ , e.g., temperature, associated with the vector  $\xi$ , representing the experimental conditions under which the measurements were taken (location, time instant, regressor vector in case of model linear in some unknown parameters). Note that for MCS applications, the identity of the node that took the measurements is secondary, provided that there are no node polluting the set of measurements with outliers.

Typically, a mobile node measures periodically  $D$  under the experimental conditions  $\xi$ . The MCS application objective is to collect the tuples  $(\xi, \mathbf{d})$ , where  $\mathbf{d} = D(\xi)$ , to the appropriate collection point  $S_i$  (responsible for the area  $\mathcal{R}_i$  where the mobile node finds itself).

### B. General Architecture

For fulfilling the objectives of the previous section, a communication architecture based on NC is designed where information is propagated to the closest sink through dissemination of coded packets.

Figure 2 represents the modules involved in NeCoRPIA. The sensing module is in charge of collecting local sensor information with associated experimental conditions, *i.e.*, the tuple

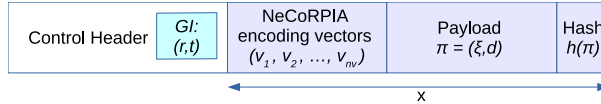


Figure 3. NeCoRPIA packet format

$(\xi, \mathbf{d})$ . The encoding module takes as input  $(\xi, \mathbf{d})$  and creates packets containing the data payload, and our specific NeCoRPIA header. The NC protocol aims at ensuring that all (coded) packets reach the collection points, by transmitting at intermediate nodes (re)combinations of received packets. The decoding module at the collection points applies the algorithm from Section IV to recover the experimental data collected in the considered area  $\mathcal{R}_i$ .

### C. Network Encoding Format

Figure 3 represents the general packet format used in NeCoRPIA: it includes a control header used by the NC dissemination protocol, followed by an encoded content, considered as symbols from  $\mathbb{F}_q$ , the Galois field with  $q$  elements.

The control header itself includes, as generation identifier (GI), a *spatio-temporal slot* (STS)  $(r, t)$ , where  $r$  is the index of the sink  $\mathcal{S}_r$  and  $t$  is the index of the considered time slot, within which the data has been collected. Only packets with the same GI are combined together by the protocol. The rest of the packet can be formally written as a vector  $\mathbf{x}$  of  $L_x$  entries in  $\mathbb{F}_q$  as

$$\mathbf{x} = (\mathbf{v}_1, \dots, \mathbf{v}_{n_v}, \boldsymbol{\pi}, \mathbf{h}), \quad (1)$$

where  $\mathbf{v}_\ell \in \mathbb{F}_q^{L_\ell}$ ,  $\ell = 1, \dots, n_v$  represent the encoding subvectors,  $\boldsymbol{\pi} = (\xi, \mathbf{d})$  is the payload, where  $\xi$  and  $\mathbf{d}$  are represented with finite precision on a fixed number  $L_\pi$  of symbols in  $\mathbb{F}_q$ , and  $\mathbf{h} = h(\boldsymbol{\pi}) \in \mathbb{F}_q^{L_h}$  is the hash of  $\boldsymbol{\pi}$ . In classical NC,  $n_v = 1$ , and  $\mathbf{v}_1$  corresponds to one of the canonical vectors of  $\mathbb{F}_q^{L_1}$ . The choice of the canonical vector requires an *agreement among mobile sensing nodes*, to avoid the same vector being selected by two or more mobile nodes in the same STS.

To avoid this resource-consuming agreement step, when a source generates a packet, NeCoRPIA assigns *random* canonical vectors  $\mathbf{e}_i \in \mathbb{F}_q^{L_\ell}$  to each  $\mathbf{v}_\ell$ ,  $\ell = 1, \dots, n_v$ . For each new payload  $\boldsymbol{\pi}$ , the random NC vector is then represented by  $(\mathbf{v}_1, \dots, \mathbf{v}_{n_v})$ . One may choose  $n_v = 1$ , but in this case,  $L_1$  should be quite long to avoid collisions, even for a moderate number of packets in each STS (this is reminiscent to the birthday paradox [23]),

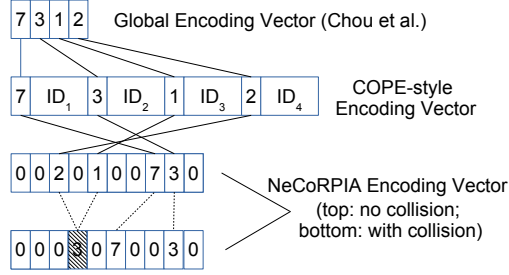


Figure 4. NeCoRPIA encoding vector format (case  $n_v = 1$ ), compared with classical formats, for a linear combination  $7d_1 + 3d_2 + d_3 + 2d_4$

see Section V-E1. This results in an encoding vector with the format represented in Figure 4 for the case  $n_v = 1$ .

The hash  $h$  is included to assist the decoding process in case of collisions.

#### D. Network Coding Protocol

The NC protocol is in charge of ensuring that the coded packets are properly reaching the data collection points for later decoding. Since the main contribution of our method lies in other parts, we only provide the sketch of a basic protocol. It operates by broadcasting measurements to all nodes within each area  $\mathcal{R}_i$ , with NC (in the spirit of [5]): with the effect that the collection point  $S_i$  will gather the information as well. It is a multi-hop protocol relying on the *control header*, shown in Figure 3, to propagate control information to the entire network (as DRAGONCAST [24] does for instance). The control headers are generated by the data collection points and copied in each encoded packet by the nodes.

The baseline functioning is as follows: at the beginning of each STS,  $S_i$  initiates data collection by generating packets with an empty payload, and with a source control header holding various parameters such as: number of encoding subvectors  $n_v$  and size of each encoding vector  $L_\ell$ ,  $\ell = 1, \dots, n_v$ , the buffer size  $G_B$ , the STS  $(r, t)$ , along with a compact description of its area  $\mathcal{R}_i$ , sensing parameters, *etc.* Upon receiving packets from a data collection point or from other nodes, and as long its current position matches  $\mathcal{R}_i$ , a node periodically retransmits (coded) packets with the most up-to-date control header. When several measurements are taken within the same STS, packets with different random encoding vectors should be generated. Furthermore, each node maintains a buffer of (at most)  $G_B$  coded vectors: when a packet associated to a given STS is received, it is (linearly) combined with all coded packets associated with the same STS in the buffer. Likewise, when a packet is



generated in some STS, a linear combination of all coded packets belonging to the same STS and stored in the buffer is computed.  $S_i$  (indirectly) instructs nodes to stop recoding of packets of a STS (and to switch to the next one) through proper indication in the control header. Note that many improvements of this scheme exist or can be designed.

#### IV. ESTIMATION OF THE TRANSMITTED PACKETS

Assume that within a STS  $(r, s)$ , mobile sensing nodes have generated a set of packets  $\mathbf{x}_1, \dots, \mathbf{x}_g$ , which may be stacked in a matrix  $\mathbf{X}$ . Assume that  $g' \geq g$  linear combinations of the packets  $\mathbf{x}_1, \dots, \mathbf{x}_g$  are received by the data collection point  $\mathcal{S}_r$  and collected in a matrix  $\mathbf{Y}'$  such that

$$\mathbf{Y}' = \mathbf{A}'\mathbf{X} = \mathbf{A}'(\mathbf{V}_1, \dots, \mathbf{V}_{n_v}, \mathbf{P}), \quad (2)$$

where  $\mathbf{A}'$  represents the NC operations that have been performed on the packets  $\mathbf{x}_1, \dots, \mathbf{x}_g$ .  $\mathbf{V}_1, \dots, \mathbf{V}_{n_v}$ , and  $\mathbf{P}$  are matrices which rows are the corresponding vectors  $\mathbf{v}_{1,i}, \dots, \mathbf{v}_{n_v,i}$ , and  $\mathbf{p}_i = (\boldsymbol{\pi}_i, \mathbf{h}_i) \in \mathbb{F}_q^{L_p}$  of the packets  $\mathbf{x}_i$ ,  $i = 1, \dots, g$ , with  $L_p = L_\pi + L_h$ . If enough linearly independent packets have been received, a full-rank  $g$  matrix  $\mathbf{Y}$  may be extracted by appropriately selecting<sup>1</sup>  $g$  rows of  $\mathbf{Y}'$ . The corresponding  $g$  rows of  $\mathbf{A}'$  form a  $g \times g$  full-rank matrix  $\mathbf{A}$ . Then, (2) becomes

$$\mathbf{Y} = \mathbf{A}(\mathbf{V}_1, \dots, \mathbf{V}_{n_v}, \mathbf{P}). \quad (3)$$

The problem is then to estimate the packets  $\mathbf{x}_1, \dots, \mathbf{x}_g$  from the received packets in  $\mathbf{Y}$ , without knowing  $\mathbf{A}$ .

Three situations have to be considered. The first is when the rows of  $(\mathbf{V}_1, \dots, \mathbf{V}_{n_v})$  are linearly independent due to the presence of some  $\mathbf{V}_\ell$  of full rank  $g$ . The second is when the rows of  $(\mathbf{V}_1, \dots, \mathbf{V}_{n_v})$  are linearly independent but there is no full rank  $\mathbf{V}_\ell$ . The third is when the rank of  $(\mathbf{V}_1, \dots, \mathbf{V}_{n_v})$  is strictly less than  $g$ , but the rank of  $\mathbf{Y}$  is equal to  $g$ . These three cases are illustrated in Examples 1-3. In the last two situations, a specific decoding procedure is required, which is detailed in Section IV-B.

**Example 1.** Consider a scenario where three nodes generate packets with  $n_v = 2$  random coding subvectors in  $\mathbb{F}_2^{L_\ell}$  with  $L_1 = L_2 = 3$ . When the generated coding vectors are  $((1, 0, 0), (0, 1, 0))$ ,  $((1, 0, 0), (0, 0, 1))$ , and  $((0, 0, 1), (1, 0, 0))$ , two nodes have selected the

<sup>1</sup>We assume that even if packet index collisions have occurred (which means  $\text{rank}(\mathbf{V}) < g$ ) the measurement process is sufficiently random to ensure that  $\mathbf{X}$  and thus  $\mathbf{Y}$  have full rank  $g$ .

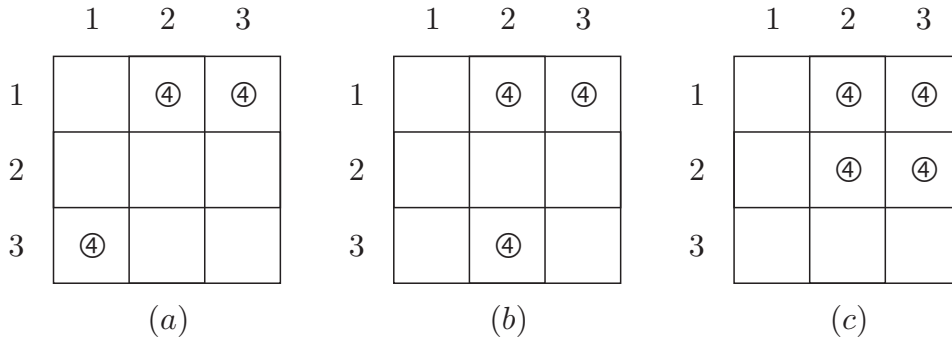


Figure 5. (a) Illustration of Example 1: No collision in the second subvectors; (b) Illustration of Example 2: single collisions in both subvectors; (c) Illustration of Example 3: single collisions in both subvectors leading to a cycle and a rank deficiency.

same first coding subvector, but all second coding subvectors are linearly independent, which allows one to recover the original packets via Gaussian elimination. This situation is illustrated in Figure 5 (a), where each coding vector may be associated to a point in a  $3 \times 3$  grid, the first coding vector representing the row index and the second coding subvector the column index. Three different column have been selected, decoding can be performed via Gaussian elimination on the second coding subvectors.

**Example 2.** Consider the same scenario as in Example 1. When the generated coding vectors are  $((1, 0, 0), (0, 1, 0))$ ,  $((1, 0, 0), (0, 0, 1))$ , and  $((0, 0, 1), (0, 1, 0))$ , collisions are observed in the first and second coding subvectors, but  $(\mathbf{V}_1, \mathbf{V}_2)$  is of full rank  $g = 3$ . This situation is illustrated in Figure 5 (b): three different entries have been chosen randomly, decoding will be easy.

**Example 3.** Consider now a scenario where four nodes generate packets with  $n_v = 2$  random coding subvectors in  $\mathbb{F}_2^{L_\ell}$  with  $L_1 = L_2 = 3$ . When the generated coding vectors are  $((1, 0, 0), (0, 1, 0))$ ,  $((1, 0, 0), (0, 0, 1))$ ,  $((0, 1, 0), (0, 1, 0))$ , and  $((0, 1, 0), (0, 0, 1))$ , the rank of  $(\mathbf{V}_1, \mathbf{V}_2)$  is only three.  $\mathbf{Y}$  will be of full rank  $g = 4$  only if  $(\mathbf{V}_1, \mathbf{V}_2, \mathbf{P})$  is of full rank. This situation is illustrated in Figure 5 (c): even if different entries have been chosen, the rank deficiency comes from the fact that the chosen entries may be indexed in such a way that they form a cycle.

### A. Decoding via Gaussian elimination

When  $\mathbf{A}$  is a  $g \times g$  full-rank matrix, a necessary and sufficient condition to have the rank of one of the matrices  $\mathbf{A}\mathbf{V}_\ell$  equal to  $g$  is that all mobile sensing nodes have chosen a different canonical subvector for the component  $\mathbf{v}_\ell$  of the NC vector. Decoding may then be performed via usual Gaussian elimination on  $\mathbf{A}\mathbf{V}_\ell$ , as in classical NC. As will be seen in Section VI, this event is unlikely, except for large values of  $L_\ell$  compared to the number of packets  $g$ .

### B. Decoding with packet index collisions

When  $\mathbf{A}$  is a  $g \times g$  full-rank matrix, the rank of  $\mathbf{A}\mathbf{V}_\ell$  is strictly less than  $g$  when at least two rows of  $\mathbf{V}_\ell$  are identical, *i.e.*, two nodes have chosen the same canonical subvector. This event is called a *collision* in the  $\ell$ -th component of the NC vector.

When  $\mathbf{A}$  is a  $g \times g$  full-rank matrix, the rank of  $\mathbf{A}[\mathbf{V}_1, \dots, \mathbf{V}_{n_v}]$  is strictly less than  $g$  when the rows of  $[\mathbf{V}_1, \dots, \mathbf{V}_{n_v}]$  are linearly dependent. This may obviously occur when the NC vector chosen by two nodes are identical, *i.e.*, there is a collision in *all*  $n_v$  components of their NC vector. This also occurs when there is no such collision, when nodes have randomly generated linearly dependent NC subvectors, as illustrated in Example 3, see also Section V.

1) *Main idea:* In both cases, one searches a full rank matrix  $\mathbf{W}$  such that  $\mathbf{X} = \mathbf{W}\mathbf{Y}$  up to a permutation of the rows of  $\mathbf{X}$ . We propose to build this *unmixing* matrix  $\mathbf{W}$  row-by-row exploiting the part of the packets containing the  $n_v$  NC subvectors  $(\mathbf{A}\mathbf{V}_1, \dots, \mathbf{A}\mathbf{V}_{n_v})$ , which helps defining a subspace in which admissible rows of  $\mathbf{W}$  have to belong. Additionally, one exploits the content of the packets and especially the hash  $h(\boldsymbol{\pi})$  introduced in Section III-C to eliminate candidate rows of  $\mathbf{W}$  leading to inconsistent payloads.

2) *Exploiting the collided NC vectors:* For all full rank  $g$  matrix  $\mathbf{Y}$ , there exists a matrix  $\mathbf{T}$  such that  $\mathbf{T}\mathbf{Y}$  is in reduced row echelon form (RREF)

$$\mathbf{T}\mathbf{Y} = \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & & \mathbf{B}_{1n_v} & \mathbf{C}_1 \\ \mathbf{0} & \mathbf{B}_{22} & & & \\ \mathbf{0} & \mathbf{0} & \ddots & & \vdots \\ \vdots & & \ddots & \mathbf{B}_{n_v n_v} & \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{C}_{n_v+1} \end{pmatrix}, \quad (4)$$

where  $\mathbf{B}_{\ell\ell}$  is a  $\rho_\ell \times L_\ell$  matrix with  $\text{rank}(\mathbf{B}_{\ell\ell}) = \rho_\ell$ . Since  $\text{rank}(\mathbf{Y}) = g$ ,  $\mathbf{C}_{n_v+1}$  is a  $\rho_{n_v+1} \times L_p$  matrix with  $\text{rank}(\mathbf{C}_{n_v+1}) = \rho_{n_v+1} = g - \sum_{\ell=1}^{n_v} \rho_\ell$ . The matrix  $\mathbf{B}_{11}$  is of rank  $\rho_1$  and its rows are  $\rho_1$  different vectors of  $\mathbf{V}_1$ .

One searches now for generic unmixing row vectors  $\mathbf{w}$  of the form  $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{n_v}, \mathbf{w}_{n_v+1})$ , with  $\mathbf{w}_1 \in \mathbb{F}_q^{\rho_1}$ ,  $\mathbf{w}_2 \in \mathbb{F}_q^{\rho_2}$ ,  $\dots$ ,  $\mathbf{w}_{n_v+1} \in \mathbb{F}_q^{\rho_{n_v+1}}$ , such that  $\mathbf{wTY} = \mathbf{x}_k$  for some  $k \in \{1, \dots, g\}$ . This implies that the structure of the decoded vector  $\mathbf{wTY}$  has to match the format introduced in (1) and imposes some constraints on  $\mathbf{w}$ , which components have to satisfy

$$\mathbf{w}_1 \mathbf{B}_{11} = \mathbf{e}_{j_1} \quad (5)$$

$$\mathbf{w}_1 \mathbf{B}_{12} + \mathbf{w}_2 \mathbf{B}_{22} = \mathbf{e}_{j_2} \quad (6)$$

$$\vdots$$

$$\mathbf{w}_1 \mathbf{B}_{1,\ell-1} + \mathbf{w}_2 \mathbf{B}_{2,\ell-1} + \dots + \mathbf{w}_{\ell-1} \mathbf{B}_{\ell-1,\ell-1} = \mathbf{e}_{j_{\ell-1}} \quad (7)$$

$$\mathbf{w}_1 \mathbf{B}_{1\ell} + \mathbf{w}_2 \mathbf{B}_{2\ell} + \dots + \mathbf{w}_\ell \mathbf{B}_{\ell\ell} = \mathbf{e}_{j_\ell} \quad (8)$$

$$\mathbf{w}_1 \mathbf{B}_{1,\ell+1} + \mathbf{w}_2 \mathbf{B}_{2,\ell+1} + \dots + \mathbf{w}_{\ell+1} \mathbf{B}_{\ell+1,\ell+1} = \mathbf{e}_{j_{\ell+1}} \quad (9)$$

$$\vdots$$

$$\mathbf{w}_1 \mathbf{B}_{1n_v} + \mathbf{w}_2 \mathbf{B}_{2n_v} + \dots + \mathbf{w}_{n_v} \mathbf{B}_{n_v n_v} = \mathbf{e}_{j_{n_v}} \quad (10)$$

$$c(\mathbf{w}_1 \mathbf{C}_1 + \mathbf{w}_2 \mathbf{C}_2 + \dots + \mathbf{w}_{n_v+1} \mathbf{C}_{n_v+1}) = 0, \quad (11)$$

where  $\mathbf{e}_{j_\ell}$  is the  $j_\ell$ -th canonical vector of  $\mathbb{F}_q^{L_\ell}$  and  $c$  is a hash-consistency verification function such that

$$c(\boldsymbol{\pi}, \mathbf{h}) = \begin{cases} 0 & \text{if } \mathbf{h} = h(\boldsymbol{\pi}), \\ 1 & \text{else,} \end{cases} \quad (12)$$

where both components  $\boldsymbol{\pi}$  and  $\mathbf{h}$  are extracted from  $\mathbf{w}_1 \mathbf{C}_1 + \mathbf{w}_2 \mathbf{C}_2 + \dots + \mathbf{w}_{n_v+1} \mathbf{C}_{n_v+1}$ .

The constraint (8) can be rewritten as

$$\mathbf{w}_\ell \mathbf{B}_{\ell\ell} = \mathbf{e}_{j_\ell} - (\mathbf{w}_1 \mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1} \mathbf{B}_{\ell-1,\ell}). \quad (13)$$

This is a system of linear equations in  $\mathbf{w}_\ell$ . Since the  $\rho_\ell \times L_\ell$  matrix  $\mathbf{B}_{\ell,\ell}$  has full row rank  $\rho_\ell$ , for every  $\mathbf{e}_{j_\ell} \in \mathbb{F}_q^{L_\ell}$  there is at most one solution for  $\mathbf{w}_\ell$ . This property allows building all candidate decoding vectors  $\mathbf{w}$  using a branch-and-prune approach described in the following algorithm, which takes  $\mathbf{TY}$  as input.

**Algorithm 1.** DeRPIA (Decoding from Random Packet Index Assignment)

- Initialization: Initialize the root of the decoding tree with an empty unmixing vector  $\mathbf{w}$ .
- Level 1: From the root node, find branches corresponding to all possible values of  $\mathbf{e}_{j_1}$ ,  $j_1 = 1, \dots, L_1$ , for which there exists a value of  $\mathbf{w}_1$  satisfying (5).
- Level 2:

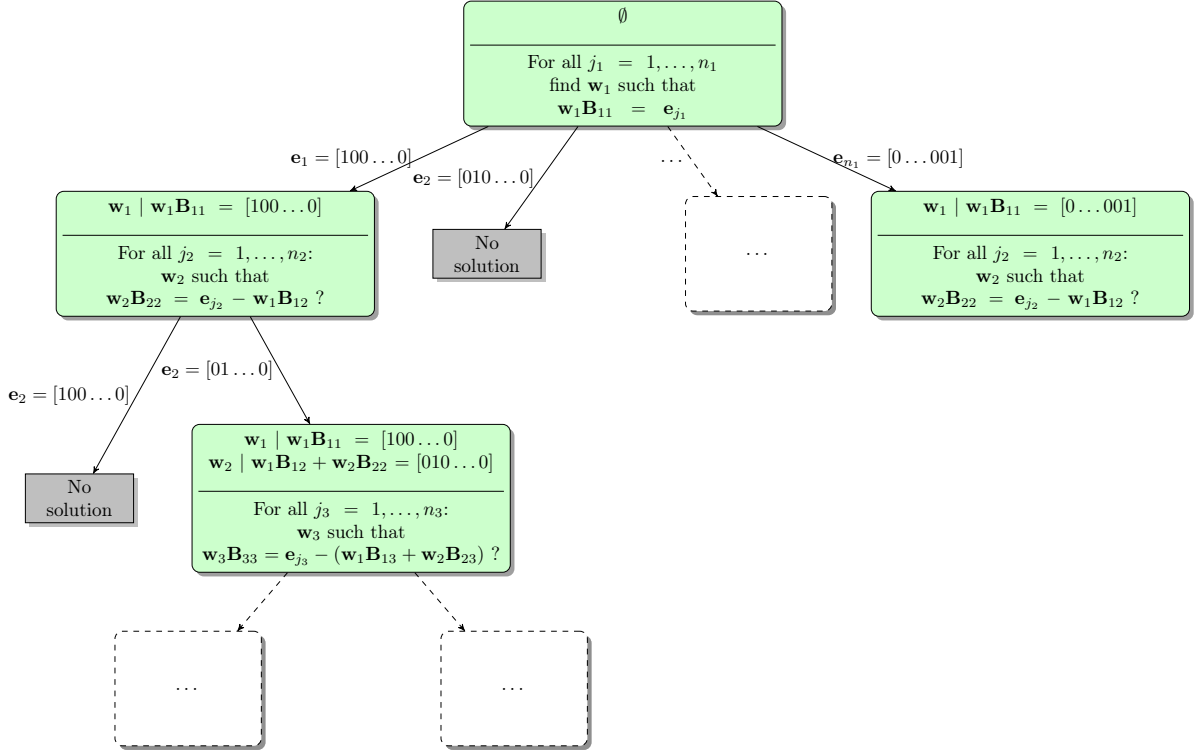


Figure 6. First steps of DERPIA starting from the root of the decoding tree.

- Expand each branch at Level 1 with branches corresponding to all possible values of  $e_{j_2}$ ,  $j_2 = 1, \dots, L_2$ , for which there exists a value of  $w_2$  satisfying (6).
- Prune all branches corresponding to a pair  $(w_1, w_2)$  for which there is no  $j_2 = 1, \dots, L_2$  such that (6) is satisfied.
- Level  $\ell$ : Expand all remaining branches at Level  $\ell - 1$  in the same way.
  - Expand each branch at Level  $\ell - 1$  for a given tuple  $(w_1, \dots, w_{\ell-1})$  with branches corresponding to all possible values of  $e_{j_\ell}$ ,  $j_\ell = 1, \dots, L_\ell$ , for which there exists a value of  $w_\ell$  satisfying (8).
  - Prune all branches corresponding to tuples  $(w_1, \dots, w_{\ell-1})$  for which there is no  $j_\ell = 1, \dots, L_\ell$  such that (8) is satisfied.
- Level  $n_v + 1$ :
  - if  $\rho_{n_v+1} = 0$ , all tuples  $(w_1, \dots, w_{n_v})$  found at Level  $n_v$  are unmixing vectors.
  - if  $\rho_{n_v+1} > 0$ , each branch of the tree corresponding to a vector  $(w_1, \dots, w_{n_v})$  satisfying all constraints (5)-(10), is expanded with all values of  $w_{n_v+1} \in \mathbb{F}_q^{\rho_{n_v+1}}$  such that (11) is satisfied. Note that (11) is not a linear equation.

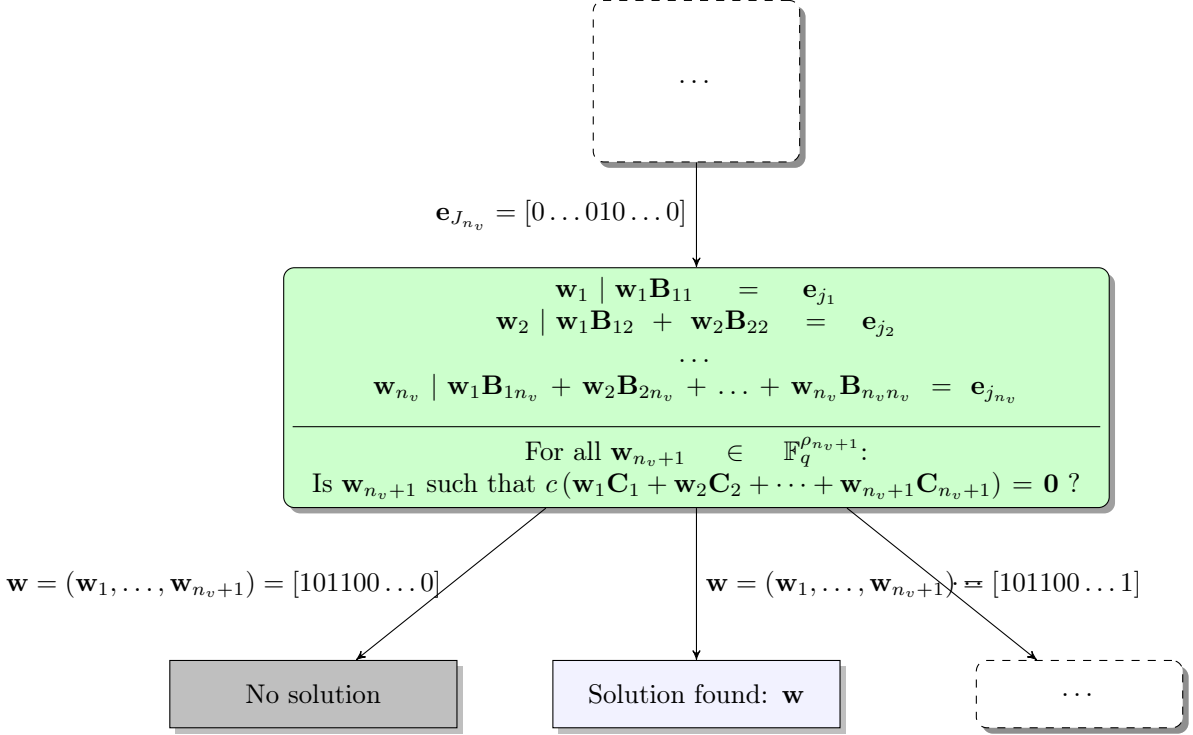


Figure 7. Last steps of DERPIA at the leaves at Level  $n_v + 1$  of the decoding tree.

The first steps of DeRPIA are illustrated in Figure 6. From the root node, several hypotheses are considered for  $\mathbf{w}_1$ . Only those satisfying (5) are kept at Level 1. The nodes at Level 1 are then expanded with candidates for  $\mathbf{w}_2$ . Figure 7 illustrates the behavior of DeRPIA at Level  $n_v + 1$ . Several hypotheses for  $\mathbf{w}_{n_v+1}$  are considered. Only those such that (11) is satisfied are kept to form the final unmixing vectors  $\mathbf{w}$ .

3) *Complexity reduction:* The aim of this section is to show that at each level of the tree built by DeRPIA, the solution of (13) does not need solving a system of linear equations but may be performed by the search in a look-up table. From Theorem 4, one sees that  $\mathbf{w}_\ell$  can take at most  $\rho_\ell + 1$  different values, which are the null vector and the canonical base vectors  $\mathbf{e}_i$  of  $\mathbb{F}^{\rho_\ell}$  corresponding to the pivots of  $\mathbf{B}_{\ell\ell}$ .

**Theorem 4.** Each  $\mathbf{w}_\ell$  satisfying (5)-(8) contains at most one non-zero component. In addition,  $\mathbf{w}_1$  contains exactly one non-zero component.

*Proof:* The proof is by contradiction. Consider first  $\ell = 1$ .  $\mathbf{B}_{11}$  is a pivot matrix of  $\rho_1$  lines. If  $\mathbf{w}_1$  contains more than one non-zero component, then  $\mathbf{w}_1 \mathbf{B}_{11}$  will contain more than one non-zero entries corresponding to the pivots of  $\mathbf{B}_{11}$  associated to the non-zero components

of  $\mathbf{w}_1$  and (5) cannot be satisfied. Moreover, with  $\mathbf{w}_1 = \mathbf{0}$ , (5) cannot be satisfied too.

Consider now some  $\ell > 1$ . In (13), since  $\mathbf{TY}$  is in RREF,  $\mathbf{B}_{\ell\ell}$  is a pivot matrix of  $\rho_\ell$  lines. Moreover all columns of the matrices  $\mathbf{B}_{1,\ell}, \dots, \mathbf{B}_{\ell-1,\ell}$  which correspond to the columns of the pivots of  $\mathbf{B}_{\ell\ell}$  are zero. This property is shared by the linear combination  $\mathbf{w}_1\mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1}\mathbf{B}_{\ell-1,\ell}$ . Thus  $\mathbf{e}_{j_\ell} - (\mathbf{w}_1\mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1}\mathbf{B}_{\ell-1,\ell})$  is either the null vector, in which case  $\mathbf{w}_\ell = \mathbf{0}$ , or contains at most one non-zero entry due to  $\mathbf{e}_{j_\ell}$  at the columns corresponding to the pivots of  $\mathbf{B}_{\ell\ell}$ . In the latter case, if  $\mathbf{w}_\ell$  contains more than one non-zero component, then  $\mathbf{w}_\ell\mathbf{B}_{\ell\ell}$  will contain more than one non-zero entry corresponding to the columns of the pivots of  $\mathbf{B}_{\ell\ell}$  associated to the non-zero components of  $\mathbf{w}_\ell$  and (8) cannot be satisfied.  $\blacksquare$

Note that for a given branch, when a set of vectors  $\mathbf{w}_1, \dots, \mathbf{w}_{\ell-1}$  has been found, a vector  $\mathbf{w}_\ell$  satisfying (5)-(11) does not necessarily exist. In such case the corresponding branch is pruned, see the last case of Example 7 in what follows.

Using Theorem 4, there is no linear system of equations to be solved any more. In practice, the search for  $\mathbf{w}_\ell$  can be even further simplified using the following corollary.

**Corollary 5.** *The search for  $\mathbf{w}_\ell$  satisfying (8) reduces to a simple look-up in a table.*

*Proof:* Assume that there exists  $\mathbf{w}_\ell$  satisfying (5)-(11). Then (13) can be rewritten as:  $\mathbf{w}_\ell\mathbf{B}_{\ell\ell} - \mathbf{e}_{j_\ell} = -(\mathbf{w}_1\mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1}\mathbf{B}_{\ell-1,\ell})$ . Here we assume that  $\mathbf{w}_\ell \neq \mathbf{0}$  and then further analyze properties established in Theorem 4:

- In the linear combination  $-(\mathbf{w}_1\mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1}\mathbf{B}_{\ell-1,\ell})$ , all components that correspond to the columns of the pivots of  $\mathbf{B}_{\ell\ell}$  are zero.
- Both sides of (13) can have at most one non-zero entry in the columns of the pivots of  $\mathbf{B}_{\ell\ell}$ . If there is one, that non-zero entry must correspond to  $\mathbf{e}_{j_\ell}$ .

It follows that  $\mathbf{e}_{j_\ell}$  must exactly correspond to the component at the column of the unique pivot of  $\mathbf{B}_{\ell\ell}$  found in the vector  $\mathbf{w}_\ell\mathbf{B}_{\ell\ell}$  and must cancel it in the expression  $\mathbf{w}_\ell\mathbf{B}_{\ell\ell} - \mathbf{e}_{j_\ell}$ . Since  $\mathbf{w}_\ell$  (assumed non-zero) has only one non-zero entry and since coefficients corresponding to pivots are equal to 1, the non-zero component of  $\mathbf{w}_\ell$  must be 1 (as it is the case for  $\mathbf{e}_{j_\ell}$ ). As a result  $\mathbf{w}_\ell\mathbf{B}_{\ell\ell}$  is actually one of the row vectors of  $\mathbf{B}_{\ell\ell}$ , and the expression  $\mathbf{w}_\ell\mathbf{B}_{\ell\ell} - \mathbf{e}_{j_\ell}$  is that row vector with a zero at the place of the component of the associated pivot. Finding one non-zero  $\mathbf{w}_\ell$  satisfying (5)-(8) is then equivalent to identify all row vectors  $\mathbf{u}$  of  $\mathbf{B}_{\ell\ell}$  such that  $\bar{\mathbf{u}} = -(\mathbf{w}_1\mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1}\mathbf{B}_{\ell-1,\ell})$ , where  $\bar{\mathbf{u}} = \mathbf{u} - \mathbf{e}_{\gamma(\mathbf{u})}$  and  $\gamma(\mathbf{u})$  is the index of the pivot column of  $\mathbf{u}$ .  $\blacksquare$

One deduces the following look-up table-based algorithm, which consists in two parts. Algorithm 2a is run once and builds a set of look-up tables from  $\mathbf{B}_{1,1}, \dots, \mathbf{B}_{n_v, n_v}$ . Algorithm 3b, then uses these look-up tables, takes as input  $(\mathbf{w}_1, \dots, \mathbf{w}_{\ell-1})$  satisfying (7) and provides the set of vectors  $(\mathbf{w}_1, \dots, \mathbf{w}_{\ell-1}, \mathbf{w}_\ell)$  satisfying (8).

**Algorithm 2a.** Construction of the look-up tables from  $\mathbf{B}_{1,1}, \dots, \mathbf{B}_{n_v, n_v}$

1) For  $\ell = 1, \dots, n_v$

a) Initialization:  $\Pi_\ell = \emptyset$ .

b) For each row vectors  $\mathbf{u}$  of  $\mathbf{B}_{\ell\ell}$ ,

i) identify the index  $\gamma(\mathbf{u})$  of its pivot column, and denote by  $\bar{\mathbf{u}}$  the row vector with a zero at the place of its pivot:  $\bar{\mathbf{u}} = \mathbf{u} - \mathbf{e}_{\gamma(\mathbf{u})}$

ii) if  $\bar{\mathbf{u}} \notin \Pi_\ell$ , then  $\Pi_\ell = \Pi_\ell \cup \{\bar{\mathbf{u}}\}$ .

c) For each  $\mathbf{v} \in \Pi_\ell$ , evaluate

$$\mathcal{C}_\ell(\mathbf{v}) = \{\mathbf{e}_i \in \mathbb{F}_q^{\rho_\ell}, i = 1, \dots, \rho_\ell | \mathbf{u} = \mathbf{e}_i \mathbf{B}_{\ell\ell} \text{ and } \mathbf{u} - \mathbf{e}_{\gamma(\mathbf{u})} = \mathbf{v}\}.$$

The sets  $\mathcal{C}_\ell(\mathbf{v})$  contain the candidate  $\mathbf{w}_\ell$  that may satisfy (8).

**Example 6.** Assume for example that

$$\mathbf{B}_{\ell,\ell} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Using Algorithm 2a, one obtains

$$\Pi_\ell = \left\{ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right\}$$

and

$$\begin{aligned} \mathcal{C}_\ell \left( \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \right) &= \left\{ \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \right\}, \\ \mathcal{C}_\ell \left( \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right) &= \left\{ \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} \right\}. \end{aligned}$$

**Algorithm 2b.** Obtain the set  $\mathcal{W}_\ell$  of all  $\mathbf{w}_\ell$  satisfying (8), from  $(\mathbf{w}_1, \dots, \mathbf{w}_{\ell-1})$  satisfying (7).

1) Initialization:  $\mathcal{W}_\ell = \emptyset$ .

2) Compute  $\mathbf{v} = -(\mathbf{w}_1 \mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1} \mathbf{B}_{\ell-1,\ell})$ .



- 3) If  $\mathbf{w}_\ell = \mathbf{0}$  satisfies (8), *i.e.*, if  $\mathbf{v} + \mathbf{e}_{j_\ell} = \mathbf{0}$  for some canonical vector  $\mathbf{e}_{j_\ell} \in \mathbb{F}_q^{L_\ell}$ , then  $\mathcal{W}_\ell = \{\mathbf{0}\}$ .
- 4) If  $\mathbf{v} \in \Pi_\ell$ , then  $\mathcal{W}_\ell = \mathcal{W}_\ell \cup \mathcal{C}_\ell(\mathbf{v})$ .

**Example 7.** Consider the results of Example 6 and a branch at level  $\ell - 1$  of the decoding tree associated to  $(\mathbf{w}_1, \dots, \mathbf{w}_{\ell-1})$ . One searches the set of  $\mathbf{w}_\ell$ s satisfying

$$\mathbf{w}_\ell \mathbf{B}_{\ell\ell} = \mathbf{e}_{j_\ell} - (\mathbf{w}_1 \mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1} \mathbf{B}_{\ell-1,\ell}). \quad (14)$$

Assume first that  $(\mathbf{w}_1, \dots, \mathbf{w}_{\ell-1})$  is such that

$$\begin{aligned} \mathbf{v}_1 &= -(\mathbf{w}_1 \mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1} \mathbf{B}_{\ell-1,\ell}) \\ &= \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{aligned}$$

then  $\mathbf{w} = \mathbf{0}$  is a solution of (14) associated to  $\mathbf{e}_{j_\ell} = \mathbf{e}_5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$ . The other solutions are given by  $\mathcal{C}_\ell(\mathbf{v}_1) = \left\{ \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \right\}$  and  $\mathcal{W}_\ell = \left\{ \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \right\}$ .

Assume now that  $(\mathbf{w}_1, \dots, \mathbf{w}_{\ell-1})$  is such that

$$\begin{aligned} \mathbf{v}_2 &= -(\mathbf{w}_1 \mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1} \mathbf{B}_{\ell-1,\ell}) \\ &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \end{aligned}$$

then  $\mathbf{w} = \mathbf{0}$  is a solution of (14) associated to  $\mathbf{e}_{j_\ell} = \mathbf{e}_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$ . There is no other solution, since  $\mathbf{v}_2 \notin \Pi_\ell$ .

Assume finally that that  $(\mathbf{w}_1, \dots, \mathbf{w}_{\ell-1})$  is such that

$$\begin{aligned} \mathbf{v}_3 &= -(\mathbf{w}_1 \mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1} \mathbf{B}_{\ell-1,\ell}) \\ &= \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \end{aligned}$$

then  $\mathcal{W}_\ell = \emptyset$ , since  $\mathbf{w} = \mathbf{0}$  is not a solution of (14) and  $\mathbf{v}_3 \notin \Pi_\ell$ .

## V. COMPLEXITY EVALUATION

To evaluate the arithmetic complexity of the NeCoRPIA decoding algorithms, one assumes that the complexity of the product of an  $n \times m$  matrix and an  $m \times p$  matrix is at most  $K_m nmp$  operations where  $K_m$  is some constant. The complexity of the evaluation of the checksum of a vector in  $\mathbb{F}_q^n$  is  $K_c n$ , where  $K_c$  is also a constant. Finally determining whether two vectors of  $n$  entries in  $\mathbb{F}_q$  are equal requires at most  $n$  operations.

The decoding complexity of NeCoRPIA depends on several parameters. First, the  $g'$  received packets of length  $L_x$  collected in  $\mathbf{Y}'$  have to be put in RREF to get  $\mathbf{TY}$ , introduced in (4). The arithmetic complexity of this operation is  $K_R (g')^2 L_x$ , with  $K_R$  a constant.

Algorithm 1 is a tree traversal algorithm. Each branch at Level  $\ell$  of the tree corresponds to a partial decoding vector  $(\mathbf{w}_1, \dots, \mathbf{w}_\ell)$  satisfying (5)-(8). The complexity depends on the number of subbranches stemming from this branch and the cost related to the partial verification of (5)-(9) for each of these subbranches. At Level  $n_v + 1$ , for each branch corresponding to a partial decoding vector  $(\mathbf{w}_1, \dots, \mathbf{w}_{n_v})$  satisfying (5)-(10), an exhaustive search has to be performed for  $\mathbf{w}_{n_v+1}$  such that  $(\mathbf{w}_1, \dots, \mathbf{w}_{n_v+1})$  satisfies (5)-(11).

Section V-A evaluates an upper bound for the number of branches at each level of the decoding tree. Section V-B determines the complexity of Algorithm 1 when the satisfying  $\mathbf{w}_\ell$ s are obtained by the solution of a system of linear equations. This version of Algorithm 1 is called DeRPIA-SLE in what follows. Section V-C describes the complexity of Algorithm 1 when the satisfying  $\mathbf{w}_\ell$ s are obtained from look-up tables as described in Algorithms 2a and 2b. This version of Algorithm 1 is called DeRPIA-LUT in what follows. As will be seen, the complexities depend on the ranks  $\rho_1, \dots, \rho_{n_v+1}$ , which distribution is evaluated in Section V-E in the case  $n_v = 1$  and  $n_v = 2$ .

#### A. Number of branches in the tree

The following corollary of Theorem 4 provides an evaluation of the number of branches that needs to be explored in Algorithm 1.

**Corollary 8.** At Level  $\ell$ , with  $1 \leq \ell \leq n_v$ , the maximum number of vectors  $(\mathbf{w}_1, \dots, \mathbf{w}_\ell)$  satisfying (8) is

$$N_b(\ell) = \rho_1 (\rho_2 + 1) \dots (\rho_\ell + 1). \quad (15)$$

The maximum number of branches that need to be considered by Algorithm 1 at Level  $\ell = n_v + 1$  is

$$N_b(n_v + 1) = \rho_1 (\rho_2 + 1) \dots (\rho_{n_v} + 1) q^{\rho_{n_v+1}} \quad (16)$$

and the total number of branches in the decoding tree is upper bounded by

$$N_b = \rho_1 + \rho_1 (\rho_2 + 1) + \dots + \rho_1 (\rho_2 + 1) \dots (\rho_{n_v} + 1) + \rho_1 (\rho_2 + 1) \dots (\rho_{n_v} + 1) q^{\rho_{n_v+1}}. \quad (17)$$

*Proof:* From Theorem 4, one deduces that  $\mathbf{w}_1$ , of size  $\rho_1$  can take at most  $\rho_1$  different values. For  $1 < \ell \leq n_v$ ,  $\mathbf{w}_\ell$  of size  $\rho_\ell$  can either be the null vector or take  $\rho_\ell$  different non-zero values. For the vector  $\mathbf{w}_{n_v+1}$ , all possible values  $\mathbf{w}_{n_v+1} \in \mathbb{F}_q^{\rho_{n_v+1}}$  have to be considered to check whether (11) is verified. The number of vectors  $(\mathbf{w}_1, \dots, \mathbf{w}_\ell)$  satisfying (8) at level  $\ell$  is thus upper bounded by the product of the number of possible values of  $\mathbf{w}_k$ ,  $k = 1, \dots, \ell$  which is (15). Similarly, the number of branches that have to be considered at Level  $n_v + 1$  of the search tree of Algorithm 1 is the product of the number of all possible values of the  $\mathbf{w}_\ell$ ,  $\ell = 1, \dots, n_v + 1$  and thus upper-bounded by (16). An upper bound of the total number of branches to consider is then

$$N_b = \sum_{\ell=1}^{n_v+1} N_b(\ell),$$

which is given by (17). ■

### B. Arithmetic complexity of DeRPIA-SLE

An upper bound of the arithmetic complexity of the tree traversal algorithm, when the number of encoding vectors is  $n_v$ , is provided by Proposition 9

**Proposition 9.** *Assume that NeCoRPIA has been performed with  $n_v$  random coding subvectors. Then an upper bound of the total arithmetic complexity of DeRPIA-SLE is*

$$K_{SLE}(n_v) = \sum_{\ell=1}^{n_v+1} N_b(\ell-1) K(\ell) \quad (18)$$

with

$$K(1) = L_1 \rho_1 L_1,$$

$$K(\ell) = K_m L_\ell (\rho_1 + \dots + \rho_{\ell-1}) + L_\ell (\ell + (\rho_\ell + 1) L_\ell),$$

$$K(n_v + 1) = K_m g L_p + (n_v - 1) L_p + q^{\rho_{n_v+1}} (K_m \rho_{n_v+1} L_p + L_p + K_c L_p). \quad (19)$$

and  $N_b(0) = 1$ . In the case  $n_v = 1$ , (18) boils down to

$$K_{SLE}(1) = \rho_1 L_1^2 + \rho_1 L_p (K_m g + q^{\rho_2} (K_m \rho_2 + 1 + K_c)). \quad (20)$$

The proof of Proposition 9 is given in Appendix A.

One sees that (18) is expressed in terms of  $\rho_1, \dots, \rho_{n_v+1}$ , which are the ranks of the matrices  $\mathbf{B}_{\ell\ell}$ . As expected, the complexity is exponential in  $\rho_{n_v+1}$ , which has to be made as small as possible. The values of  $\rho_1, \dots, \rho_{n_v+1}$  depend on those of  $L_\ell$  and  $g$ , as will be seen in Section V-E.

### C. Arithmetic complexity of DeRPIA-LUT

The tree obtained with DeRPIA-LUT is the same as that obtained with DeRPIA-SLE. The main difference comes from the arithmetic complexity when expanding one branch of the tree. Algorithm 2a is run only once. Algorithm 2b is run for each branch expansion. An upper bound of the total arithmetic complexity of DeRPIA-LUT is given by the following proposition.

**Proposition 10.** *Assume that NeCoRPIA has been performed with  $n_v$  random coding subvectors. Then an upper bound of the total arithmetic complexity of DeRPIA-LUT is*

$$K_{LUT}(n_v) = \sum_{\ell=1}^{n_v} (K_{LU,1}(\ell) + K_{LU,2}(\ell)) + \sum_{\ell=1}^{n_v} N_b(\ell-1) K_{LU,3}(\ell) + N_b(n_v) K(n_v+1) \quad (21)$$

with

$$\begin{aligned} K_{LU,1}(\ell) &= \rho_\ell + L_\ell \frac{\rho_\ell(\rho_\ell+1)}{2}, \\ K_{LU,2}(\ell) &= \rho_\ell(L_\ell+1 + \rho_\ell L_\ell), \\ K_{LU,3}(\ell) &= K_m L_\ell(\rho_1 + \dots + \rho_{\ell-1}) + L_\ell(\ell + \rho_\ell), \end{aligned}$$

and  $K(n_v+1)$  given by (19).

The proof of Proposition 10 is provided in Appendix A3.

Again, as in (18), the complexity (21) depends on the ranks  $\rho_1, \dots, \rho_{n_v+1}$  of the matrices  $\mathbf{B}_{\ell\ell}$ .

### D. Complexity comparison

When comparing  $K_{SLE}(n_v)$  and  $K_{LUT}(n_v)$ , one observes that there is a (small) price to be paid for building the look-up tables corresponding to the first sum in (21). Then, the complexity gain provided by the look-up tables appears in the expression of  $K_{LU,3}(\ell)$ , which is linear in  $L_\ell$ , whereas  $K(\ell)$  is quadratic in  $L_\ell$ . Nevertheless, the look-up procedure is less useful when there are many terminal branches to consider, *i.e.*, when  $\rho_{n_v+1}$  is large, since in this case, the term  $N_b(n_v) K(n_v+1)$  dominates in both expressions of  $K_{SLE}(n_v)$  and  $K_{LUT}(n_v)$ .

### E. Distribution of the ranks $\rho_1, \dots, \rho_{n_v+1}$

Determining the distributions of  $\rho_1, \dots, \rho_{n_v+1}$  in the general case is relatively complex. In what follows, one focuses on the cases  $n_v = 1$  and  $n_v = 2$ . Experimental results for other values of  $n_v$  are provided in Section VI.

1) *NC vector with one component,  $n_v = 1$* : In this case, the random NC vectors are gathered in the matrix  $\mathbf{V}_1$ . Once  $\mathbf{Y}$  has been put in RREF, the rows of the matrix  $\mathbf{B}_{11}$  of rank  $\rho_1$  are the  $\rho_1$  linearly independent vectors of  $\mathbf{V}_1$ . The distribution of  $\rho_1$  may be analyzed considering the classical urn problem described in [23]. This problem may be formulated as: assume that  $g$  indistinguishable balls are randomly dropped in  $L_1$  distinguishable boxes. The distribution of the number  $X_g^{L_1}$  of boxes that contain at least one ball is described in [23] citing De Moivre as

$$P(X_g^{L_1} = k) = f(g, L_1, k), \quad (22)$$

with

$$f(g, L, k) = \frac{L(L-1)\dots(L-k+1)}{L^g} S(g, k),$$

where  $S(g, k)$  denotes the Stirling numbers of the second kind [25]. In the context of NeCorPIA, a *collision* corresponds to at least two balls dropped in the same box. The probability mass function (pmf) of the rank  $\rho_1$  of  $\mathbf{B}_{11}$  is then that of the number of boxes containing at least one ball and is given by (22). The pmf of  $\rho_2$  is deduced from (22) as

$$\begin{aligned} P(\rho_2 = g - k) &= 1 - P(\rho_1 = k) \\ &= 1 - f(g, L_1, k). \end{aligned} \quad (23)$$

Figure 8 shows the distribution of  $\rho_1$  (left) and the complementary cumulative distribution function  $\rho_2$  (right) for different values of  $g$  when  $L_1 = 100$ . One sees that  $\Pr(\rho_2 > 1) < 0.0025$  when  $g = 5$  and  $\Pr(\rho_2 > 1) < 0.062$  when  $g = 10$ . In the decoding complexity, the exponential term of  $K(n_v + 1)$  in both (18) and (21) will thus be of limited impact. When  $g = 30$ ,  $\Pr(\rho_2 > 1)$  is larger than 94%. The exponential term in the complexity may thus become overwhelming.

2) *NC vector with two components,  $n_v = 2$* : In this case, the random NC vectors form the matrix  $(\mathbf{V}_1, \mathbf{V}_2)$ . The pmf of  $\rho_1$  is still given by (22). Determining the pmf of the rank  $\rho_2$  of  $\mathbf{B}_{22}$  is much more complicated. Hence, we will first evaluate an upper bound on the

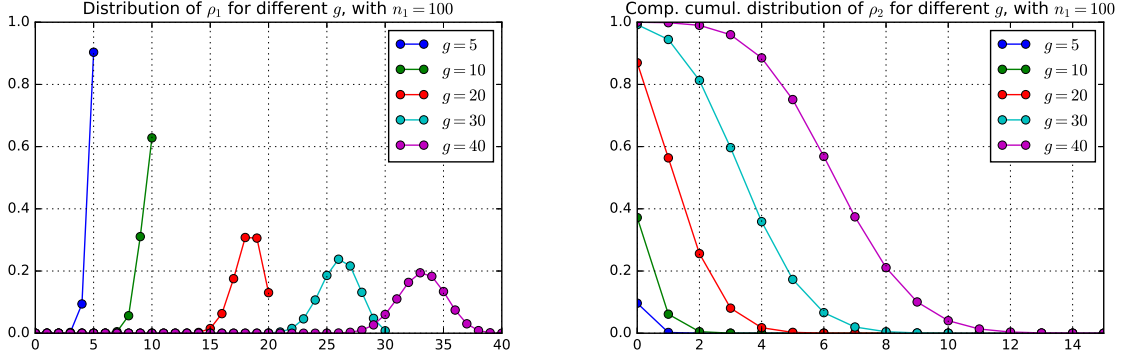


Figure 8. Case  $n_v = 1$ , theoretical and experimental distributions of  $\rho_1$  (left) and theoretical and practical complementary cumulative distribution function of  $\rho_2$  (right) for different values of  $g$  for  $L_1 = 100$ .

probability that  $\rho_3 = 0$  and an approximation of the pmf of  $\rho_3$ . Using these results, one will derive an estimate of the pmf of  $\rho_2$ .

In general, to have  $(\mathbf{V}_1, \dots, \mathbf{V}_{n_v})$  of full rank, *i.e.*,  $\rho_{n_v+1} = 0$ , it is necessary that no pair of nodes has generated packets with the same random coding subvectors. In the corresponding urn problem, one has now to consider  $g$  balls thrown into  $L_1 L_2 \dots L_{n_v}$  boxes. In the case  $n_v = 2$ , a node selects two random NC subvectors  $\mathbf{e}_i \in \mathbb{F}_q^{L_1}$  and  $\mathbf{e}_j \in \mathbb{F}_q^{L_2}$ . The pair of indices  $(i, j)$  may be interpreted as the row and column index of the box in which a ball has been dropped, when  $L_1 L_2$  boxes are arranged in a rectangle with  $L_1$  rows and  $L_2$  columns. The probability of having  $g$  balls thrown in  $L_1 L_2 \dots L_{n_v}$  boxes reaching  $g$  different boxes, *i.e.*, of having coding subvectors different for all  $g$  nodes is  $P(X_g^{L_1 \dots L_{n_v}} = g)$  and can again be evaluated with (22).

A rank deficiency happens when the previous necessary condition is not satisfied, but it may also happen in other cases, see Example 3. As a consequence, one only gets the following upper bound

$$P(\rho_1 + \dots + \rho_{n_v} = g) \leq f(g, L_1 L_2 \dots L_{n_v}, g). \quad (24)$$

If one assumes that the rank deficiency is only due to nodes having selected the same random coding subvectors, similarly, one may apply (22) as in the case  $n_v = 1$  and get the following approximation

$$P(\rho_1 + \dots + \rho_{n_v} = k) \approx f(g, L_1 L_2 \dots L_{n_v}, k) \quad (25)$$

which leads to

$$P(\rho_{n_v+1} = g - k) \approx f(g, L_1 L_2 \dots L_{n_v}, k). \quad (26)$$

This is an approximation, since even if all nodes have selected different coding subvectors, we might have a rank deficiency, as illustrated by Example 3.

In the case  $n_v = 2$ , one will now build an approximation of

$$\begin{aligned} P(\rho_1 = k_1, \rho_2 = k_2, \rho_3 = k_3) &= P(\rho_2 = k_2 | \rho_1 = k_1, \rho_3 = k_3) \\ &P(\rho_1 = k_1 | \rho_3 = k_3) P(\rho_3 = k_3). \end{aligned} \quad (27)$$

Using the fact  $k_1 + k_2 + k_3 = g$ , one has

$$P(\rho_2 = k_2 | \rho_1 = k_1, \rho_3 = k_3) = \begin{cases} 1 & \text{if } k_2 = g - k_1 - k_3 \\ 0 & \text{else.} \end{cases}$$

One will assume that the only dependency between  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$  that has to be taken into account is that  $\rho_1 + \rho_2 + \rho_3 = g$  and that  $P(\rho_3 = k_3)$  is given by (26). Then (27) becomes

$$P(\rho_1 = k_1, \rho_2 = g - k_1 - k_3, \rho_3 = k_3) = P(\rho_1 = k_1) f(g, L_1 L_2, g - k_3). \quad (28)$$

Combining (22) and (28), one gets

$$P(\rho_1 = k_1, \rho_2 = g - k_1 - k_3, \rho_3 = k_3) = f(g, L_1, k_1) f(g, L_1 L_2, g - k_3), \quad (29)$$

which may also be written as

$$P(\rho_1 = k_1, \rho_2 = k_2, \rho_3 = g - k_1 - k_2) = f(g, L_1, k_1) f(g, L_1 L_2, k_1 + k_2). \quad (30)$$

Figure 9 (left) shows the joint pmf  $P(\rho_1 = k_1, \rho_2 = k_2)$  deduced from (27) as a function of  $k_1$  and  $k_2$  for different values of  $g = 30$  with  $L_1 = 50$  and  $L_2 = 50$ . Figure 9 (right) shows the complementary CDF of  $\rho_3$  again deduced from (27) for different values of  $g$  with  $L_1 = 50$  and  $L_2 = 50$ . Now, when  $g = 30$ ,  $\Pr(\rho_3 > 1)$  is about 1.3%. When  $g = 40$ ,  $\Pr(\rho_3 > 1)$  is about 3.8%. In both cases, the exponential term of  $K(n_v + 1)$  in both (18) and (21) will thus be of limited impact. Considering  $n_v = 2$  allows one to consider much larger generations than with  $n_v = 1$ .

Finally, Figure 10 shows the complementary CDF of  $\rho_3$  for  $g = 40$  and different values of the pair  $(L_1, L_2)$  such that  $L_1 + L_2 = 100$ . Choosing  $L_1 = L_2$  provides the smallest probability of rank deficiency, which is consistent with the hypothesis that the rank deficiency is mainly due to nodes having selected the same random coding subvectors. The maximum of the product  $L_1 \dots L_{n_v}$  with a constraint on  $L_1 + \dots + L_{n_v}$  is obtained taking  $L_1 = L_2 = \dots = L_{n_v}$ .

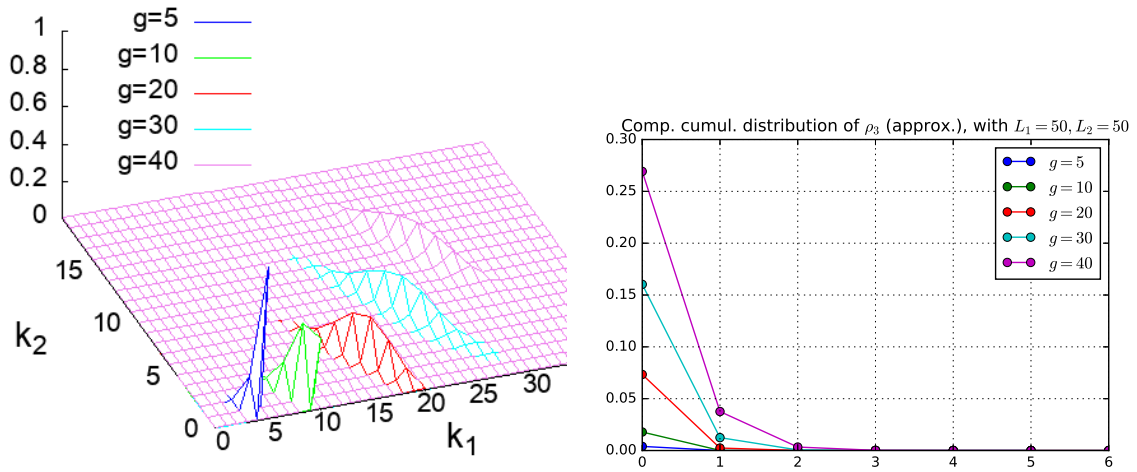


Figure 9. Case  $n_v = 2$ , joint distribution of  $\rho_1$  and  $\rho_2$  (left) and approximated complementary cumulative distribution function of  $\rho_3$  deduced from (26) (right) for different values of  $g$  for  $L_1 = 50$  and  $L_2 = 50$ .

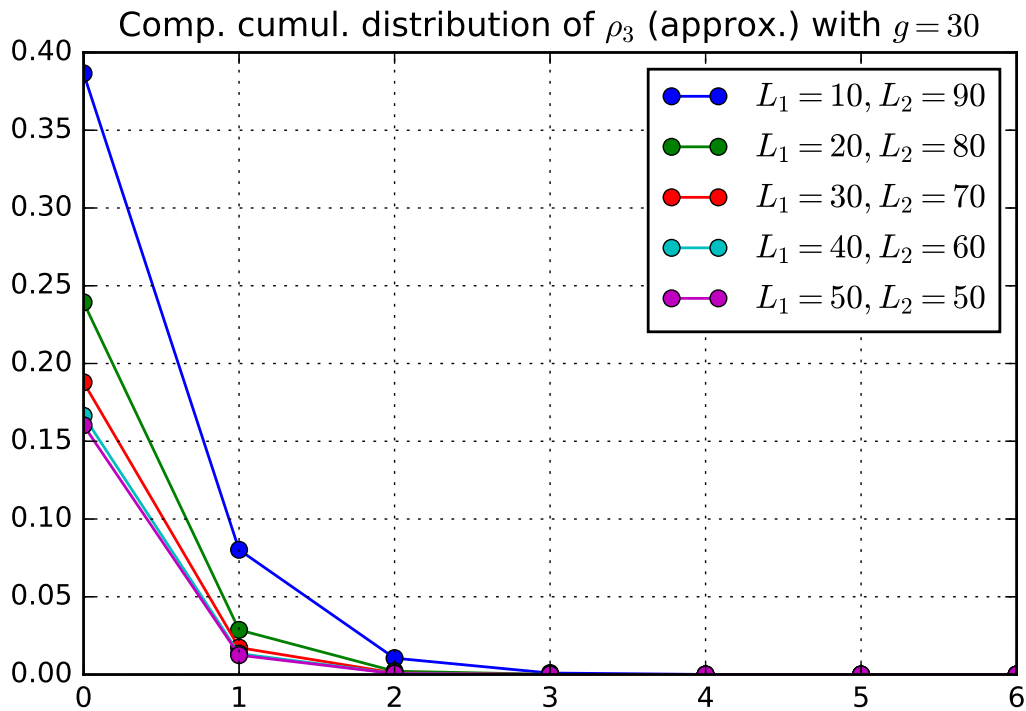


Figure 10. Case  $n_v = 2$ , complementary cumulative distribution function of  $\rho_3$  deduced from (26) for  $g = 30$  and different values of  $L_1$  and  $L_2$  such that  $L_1 + L_2 = 100$ .



## VI. PERFORMANCE EVALUATION

Several simulation scenarios have been considered to evaluate the performance of NeCoRPIA in terms of complexity and decoding error probability. In each simulation run, a set of  $g$  packets  $\mathbf{x}_1, \dots, \mathbf{x}_g$  containing  $n_v$  random coding subvectors with elements in  $\mathbb{F}_2$  and with the same STS information are generated. The payload  $\boldsymbol{\pi} = (\boldsymbol{\xi}, \mathbf{d})$  is replaced by a unique sufficiently long packet identifier to ensure that the  $g$  randomly generated packets are linearly independent. Hash functions producing hash of different lengths are considered. The NC operations are simulated by the generation of a full-rank  $g \times g$  random coding matrix  $\mathbf{A}$  with elements in  $\mathbb{F}_2$ . The received packets are stored in a matrix  $\mathbf{Y} = \mathbf{A}\mathbf{X}$  of full rank. Writing  $\mathbf{Y}$  in RREF, one obtains (4).

In all simulations, the total length  $\sum_{\ell=1}^{n_v} L_\ell$  of the NC subvectors is fixed at 100.

Upon completion of Algorithm 1, with the final list of candidate unmixing vectors satisfying (5)-(11), one is always able to get  $\mathbf{x}_1, \dots, \mathbf{x}_g$ , since  $\mathbf{A}$  is of full rank. Nevertheless, other unmixed packets may be obtained, even if they satisfy all the previous constraints when the rank of the NC header is not sufficient and the hash was inefficient. To evaluate the probability of such event, one considers first the number  $n_w$  of different unmixing vectors provided by Algorithm 1. Among these  $n_w$  vectors,  $g$  of them lead to the generated packets, the  $n_w - g$  others to erroneous packets. Considering a hash of  $L_h$  elements of  $\mathbb{F}_q$ , the probability of getting a given hash for a randomly generated payload is uniform and equal to  $1/q^{L_h}$ . The probability that one of the  $n_w - g$  erroneous packets has a satisfying hash is thus  $1/q^{L_h}$  and the probability that none of them has a satisfying hash is  $(1 - 1/q^{L_h})^{n_w - g}$ . The probability of decoding error is then

$$P_e = 1 - (1 - 1/q^{L_h})^{n_w - g}. \quad (31)$$

An upper bound for  $n_w$  is provided by  $N_b(n_v + 1)$ . In what follows, this upper bound is used to get an upper bound for  $P_e$  evaluated as

$$\bar{P}_e = E \left( 1 - (1 - 1/q^{L_h})^{N_b(n_v + 1) - g} \right), \quad (32)$$

where the expectation is taken either considering the pmf of  $(\rho_1, \dots, \rho_{n_v + 1})$  or an estimated pmf obtained from experiments.

The other performance metrics used to evaluate the performance of NeCoRPIA are:

- the number of branches explored in the decoding tree before being able to decode all the packets  $\mathbf{x}_1, \dots, \mathbf{x}_g$ ,

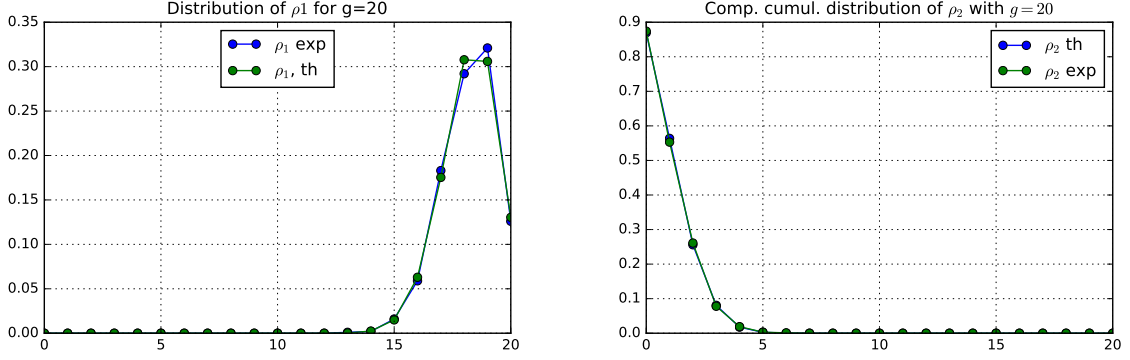


Figure 11. Case  $n_v = 1$ , theoretical and experimental distributions of  $\rho_1$  (left) when  $g = 20$  and theoretical and practical complementary cumulative distribution function of  $\rho_2$  (right) when  $g = 20$  for  $L_1 = 100$ .

- the arithmetic complexity of the decoding process.

In what follows, to evaluate the complexity, one assumes that  $L_h = 2$  bytes or  $L_h = 4$  bytes and that  $L_x = 256$  bytes. For the various constants in the arithmetic complexity, one chooses  $K_m = 2$ ,  $K_R = 3$ , see, *e.g.*, [26], and  $K_c = 3$ , see [27].

Averages are taken over 1000 realizations.

#### A. Case $n_v = 1$

The theoretical and experimental distributions of  $\rho_1$ , evaluated using (22), for  $L_1 = 100$  and  $g = 20$ , are represented in Figure 8 (left). A very good match between theoretical and experimental distributions is observed.

For a fixed value of  $\rho_1$ , the upper-bound (16) for the number of branches in the decoding tree boils down to

$$N_b = \rho_1 + \rho_1 q^{g-\rho_1}.$$

One is then able to evaluate the average value of  $N_b$

$$E(N_b) = \sum_{\rho_1=0}^g f(g, L_1, \rho_1) (\rho_1 + \rho_1 q^{g-\rho_1}). \quad (33)$$

Figure 12 represents the theoretical and experimental values of  $E(N_b)$  (left) and of  $\bar{P}_e$  (right) as a function of  $g$  when  $L = 100$ . The expression of  $E(N_b)$  provided by (33) as well as that of  $\bar{P}_e$  given by (32) match well the experimental results. As expected, for a given value of  $g$ , the decoding error probability is much less with  $L_h = 32$  than with  $L_h = 16$ .

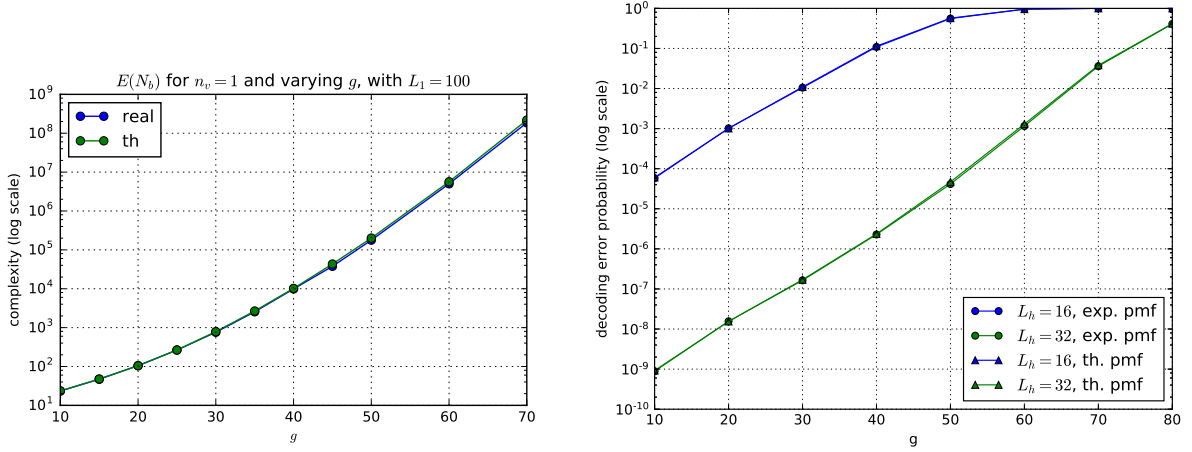


Figure 12. Case  $n_v = 1$ , theoretical and practical values of  $E(N_b)$  (left) of the decoding error probability  $\bar{P}_e$  with  $L_h = 16$  and  $L_h = 32$  (right) for different values of  $g$  when  $L_1 = 100$ .

When  $g = 20$  and  $L_h = 16$ , one gets  $\bar{P}_e = 10^{-3}$ , which may be sufficiently small for some applications.

The arithmetic complexity of both decoding algorithms is then compared to that of a plain NC decoding. The latter requires only a single RREF (once the nodes have agreed on the NC vector they should select). Since both DeRPIA decoding algorithms also require an initial RREF, the complexity ratio is always larger than one.

In the case  $n_v = 1$ , the arithmetic complexity of decoding of plain NC-encoded packets is thus

$$A_{\text{NC}} = 3g^2 L_x,$$

where the length  $L_x$  is expressed as the number of elements of  $\mathbb{F}_q$  in which the NC operations are performed. The arithmetic complexities of DeRPIA-SLE and DeRPIA-LUT depend on  $\rho_1$  and  $\rho_2$ . Their expected values are

$$A_{\text{SLE}}(n_v = 1) \simeq A_{\text{NC}} + E \left[ \rho_1 L_1^2 + \rho_1 L_p (K_m g + q^{\rho_2} (K_m \rho_2 + 1 + K_c)) \right],$$

and

$$A_{\text{LUT}}(n_v = 1) \simeq A_{\text{NC}} + \sum_{\ell=1}^2 E \left[ \rho_1 + L_1 \frac{\rho_1 (\rho_1 + 1)}{2} + \rho_1 (L_1 + 1 + \rho_1 L_1) \right. \\ \left. + L_1 (1 + \rho_1) + \rho_1 L_p (K_m g + q^{\rho_2} (K_m \rho_2 + 1 + K_c)) \right]$$

where the expectation is evaluated using (22) and (23).

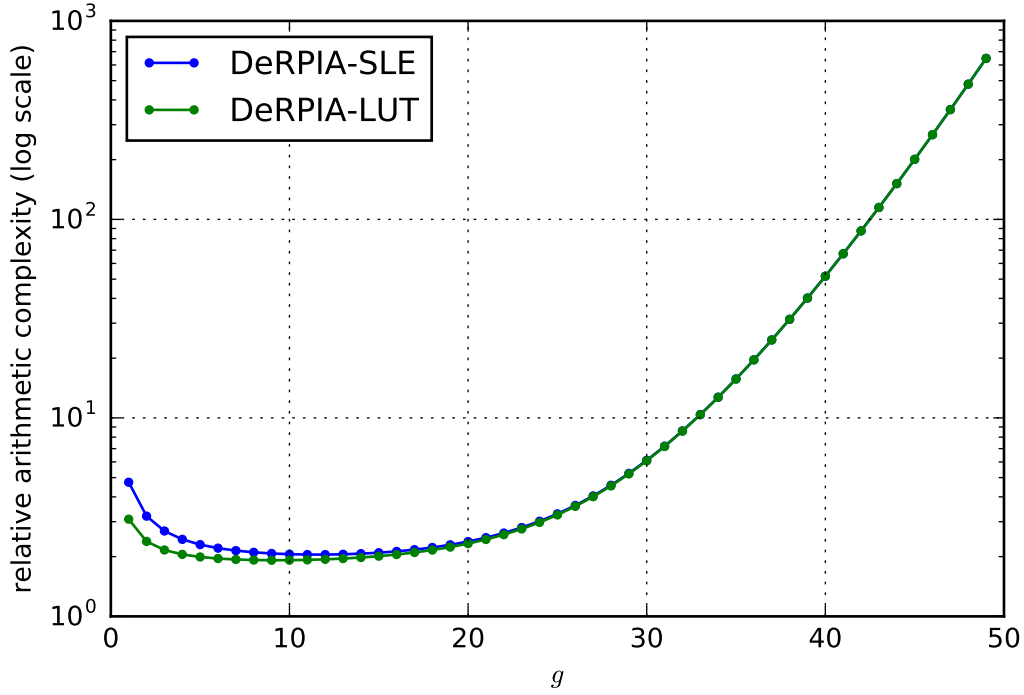


Figure 13. Case  $n_v = 1$ , evolution of the ratio of the expected decoding complexity of DeRPIA-SLE and DeRPIA-LUT with respect to the complexity of a simple RREF transformation for different value of  $g$  when  $L_1 = 100$ .

Figure 13 compares the relative arithmetic complexity of the two variants of DeRPIA with respect to  $A_{NC}$ . One sees that  $A_{SLE}$  and  $A_{LUT}$  are about twice that of  $A_{NC}$  when  $g \leq 25$ . When  $g \geq 30$ , the complexity increases exponentially.  $A_{LUT}$  is only slightly less than  $A_{SLE}$  for small values of  $g$ . Again, for values of  $g$  larger than 25, the exponential term dominates.

### B. Case $n_v = 2$

In this case, an expression is only available for the pmf of  $\rho_1$  as a function of  $g$  and  $L_1$ , see Section V-E2. Figure 14 represents the histograms of  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$  for different values of  $g$ , as well as the theoretical pmf of  $\rho_1$ .

Figure 15 shows that the approximation of  $P(\rho_3 = k)$  provided by (26) matches well the histogram of  $\rho_3$  when  $g \leq 40$ . When  $g = 80$ , the approximation is no more valid: the effect of cycles, illustrated in Example 3, becomes significant.

For the complexity evaluation, in the case  $N_v = 2$ , for a given value of  $\rho_1$  and  $\rho_2$ , the upper bound (16) becomes

$$N_b = \rho_1 + \rho_1(\rho_2 + 1) + \rho_1(\rho_2 + 1)q^{g - \rho_1 - \rho_2}. \quad (34)$$

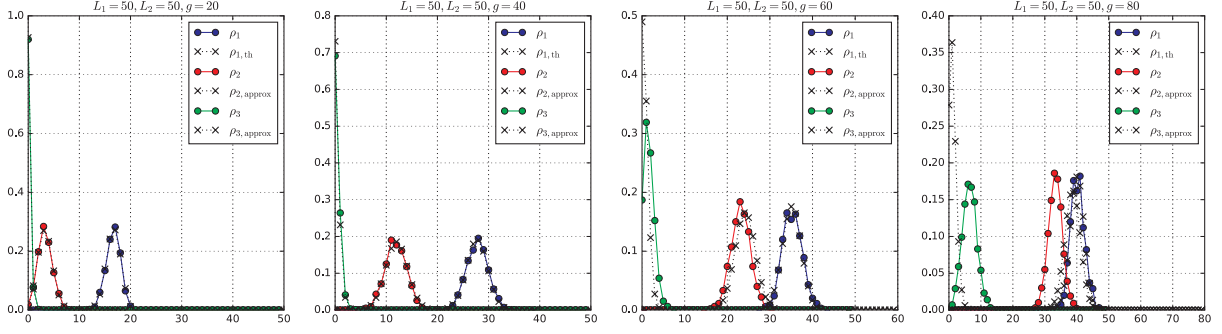


Figure 14. Case  $n_v = 2$ , theoretical pmf of  $\rho_1$  and histograms of  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$  for different values of  $g$  when  $L_1 = L_2 = 50$ .

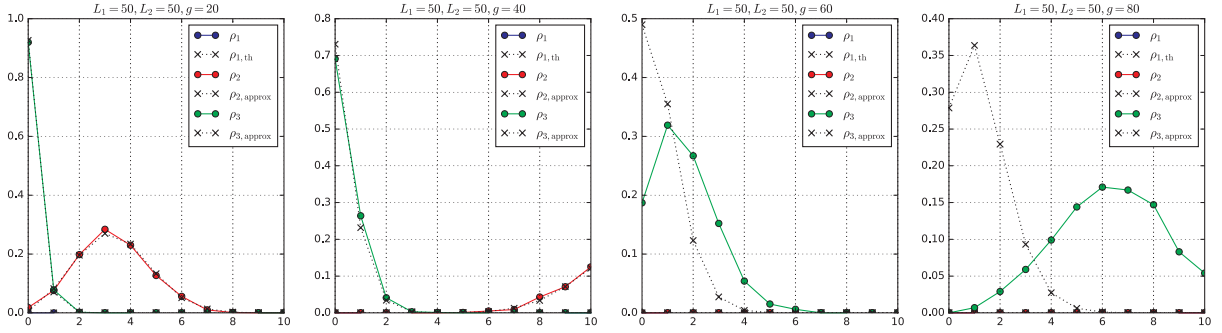


Figure 15. Case  $n_v = 2$ , distribution of  $\rho_1$  and histograms of  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$  for different values of  $g$  for  $L_1 = L_2 = 50$  (zoom of Fig 14).

The average value of  $N_b$  is then evaluated using the approximation (30) of the joint pmf of  $(\rho_1, \rho_2, \rho_3)$  as

$$E(N_b) = \sum_{\rho_1 + \rho_2 \leq g} f(g, L_1, \rho_1) f(g, L_1 L_2, \rho_1 + \rho_2) (\rho_1 + \rho_1(\rho_2 + 1) + \rho_1(\rho_2 + 1)q^{g - \rho_1 - \rho_2}). \quad (35)$$

Figure 16 (left) represents the theoretical and experimental values of  $E(N_b(3))$  and  $E(N_b)$  as a function of  $g$  when  $L_1 = 50$  and  $L_2 = 50$ . The theoretical values of  $E(N_b(3))$  and  $E(N_b)$  are evaluated in two ways: First, with the estimated pmf of  $(\rho_1, \rho_2, \rho_3)$  given by (30) and second with the estimate of this pmf obtained from experiments. The average number  $\tilde{N}_b$  of branches in the tree and in the last level of the tree  $\tilde{N}_b(3)$  obtained from experiments are also provided. The value of  $E(N_b(3))$  and  $E(N_b)$  evaluated from the experimental pmf are good upper-bounds for  $\tilde{N}_b(3)$  and  $\tilde{N}_b$ . The values of  $E(N_b(3))$  and  $E(N_b)$  obtained from (30) match well those obtained from the estimated pmf of  $(\rho_1, \rho_2, \rho_3)$  only for  $g \leq 50$ . When  $g \geq 60$ , the lack of accuracy of the theoretical pmf of  $\rho_3$  becomes significant:  $\tilde{N}_b(3)$  and

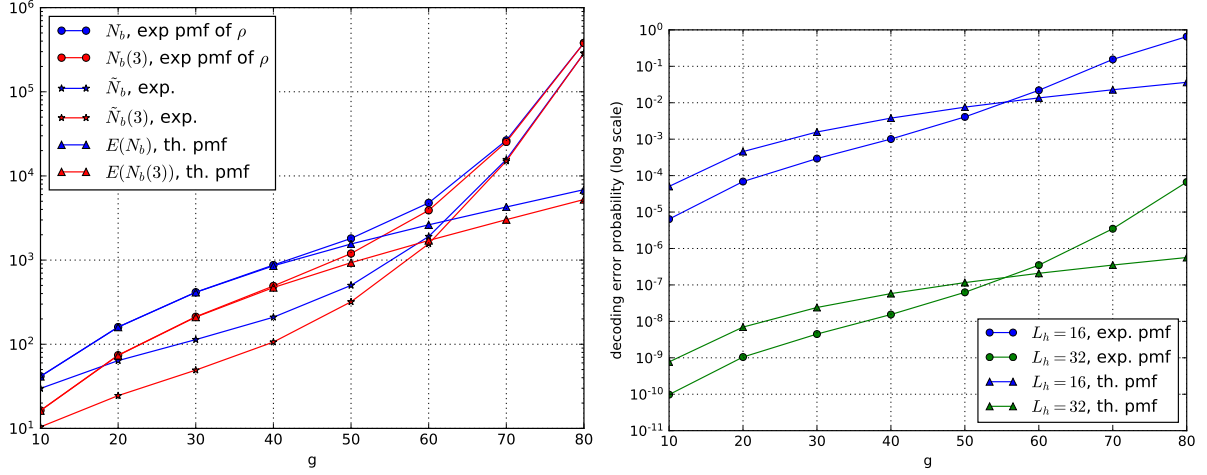


Figure 16. Case  $n_v = 2$ , theoretical and experimental values of  $E(N_b(3))$  and  $E(N_b)$  (left) and of the decoding error probability  $\bar{P}_e$  with  $L_h = 16$  and  $L_h = 32$  (right) for different values of  $g$  when  $L_1 = 50$  and  $L_2 = 50$ .

$\tilde{N}_b$  are underestimated. One observes that considering two encoding subvectors significantly reduces the number of branches that have to be explored in the decoding tree. For example, when  $g = 70$ ,  $E(N_b) \simeq 1.6 \times 10^4$  with  $n_v = 2$ , whereas  $E(N_b) \simeq 1.8 \times 10^8$  with  $n_v = 1$ .

Figure 16 (right) represents  $\bar{P}_e$  obtained from the approximate pmf of  $(\rho_1, \rho_2, \rho_3)$  given by (30) and from the estimate of this pmf obtained from experiments. Again, both evaluations match well when  $g \leq 50$ . Compared to the case  $n_v = 1$ , the probability of decoding error reduces significantly thanks to the reduction of the number of branches in the decoding tree at level  $n_v + 1$ .

In the case  $n_v = 2$ , the arithmetic complexity of decoding of plain NC-encoded packets is still  $A_{NC}$ . The arithmetic complexities of DeRPIA-SLE and DeRPIA-LUT depend now on  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$ . Their expected values are

$$A_{SLE}(n_v = 2) \simeq A_{NC} + E \left[ \rho_1 L_1^2 + \rho_1 L_2 (K_m \rho_1 + 2 + (\rho_2 + 1) L_2) \right] \\ + E \left[ \rho_1 (\rho_2 + 1) L_p (K_m g + 1 + q^{\rho_3} (K_m \rho_3 + 1 + K_c)) \right]$$

and

$$A_{LUT}(n_v = 2) \simeq A_{NC} + \sum_{\ell=1}^2 E \left[ \rho_\ell + L_\ell \frac{\rho_\ell (\rho_\ell + 1)}{2} + \rho_\ell (L_\ell + 1 + \rho_\ell L_\ell) \right] \\ + E \left[ L_1 (1 + \rho_1) + \rho_1 L_2 (K_m \rho_1 + 2 + \rho_2) \right] \\ + E \left[ \rho_1 (\rho_2 + 1) L_p (K_m g + 1 + q^{\rho_3} (K_m \rho_3 + 1 + K_c)) \right]$$

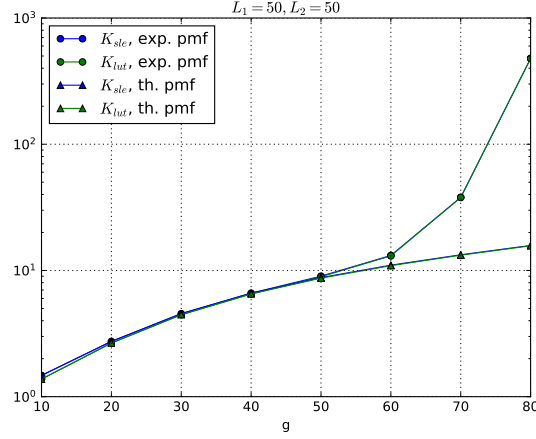


Figure 17. Case  $n_v = 2$ , evolution of the theoretical and experimental values of the ratio of the expected decoding complexity of DeRPIA-SLE and DeRPIA-LUT with respect to the complexity of a simple RREF transformation as a function of  $g$  when  $L_1 = L_2 = 50$ .

where the expectations are evaluated using (30).

Figure 17 compares the relative arithmetic complexity of the two variants of DeRPIA with respect to  $A_{NC}$ . One sees that  $A_{SLE}$  and  $A_{LUT}$  are almost equal and less than ten times  $A_{NC}$  when  $g \leq 50$ . When  $g \geq 60$ , the complexity increases exponentially. This is again not well predicted by the theoretical approximation, due to the degraded accuracy of the theoretical expression of  $\rho_3$  when  $g \geq 60$ .  $A_{LUT}$  is again only slightly less than  $A_{SLE}$  for small values of  $g$ . Now, the exponential term dominates in the complexity for values of  $g$  larger than 70.

### C. Case $n_v > 2$

In this case, since even an approximate expression of the joint pmf of  $(\rho_1, \dots, \rho_{n_v+1})$  is difficult to obtain, only the pmf of  $\rho_1$  and the histograms for  $\rho_2, \dots, \rho_{n_v+1}$  are provided, see Figure (18) for  $n_v = 3$  and Figure (18) for  $n_v = 4$ .

One observes that when  $n_v = 4$ , even for  $g = 80$ ,  $P(\rho_5 \geq 1) \leq 0.012$ . The contribution of the exponential term in the decoding complexity will thus remain negligible. This can be observed in Figure 20, which shows the evolution of the ratio of the decoding complexity of DeRPIA-SLE and DeRPIA-LUT with respect to the complexity of a simple RREF transformation as a function of  $g$  for different values of  $n_v$ . For values of  $g$  for which  $\rho_{n_v+1}$  remains small, the complexity increases with  $n_v$ , due to the increasing number of branches that have to be considered at intermediate levels of the decoding tree. When  $\rho_{n_v+1}$  increases, the complexity is dominated by the exponential term in the complexity due to the number

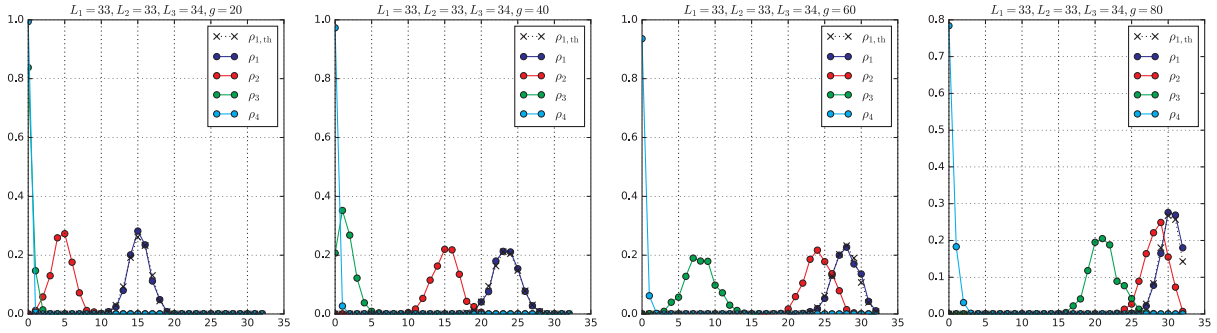


Figure 18. Case  $n_v = 3$ , distribution of  $\rho_1$  and histograms of  $\rho_1$ ,  $\rho_2$ ,  $\rho_3$ , and  $\rho_4$  for different values of  $g$  for  $L_1 = 33$ ,  $L_2 = 33$ ,  $L_3 = 34$ .

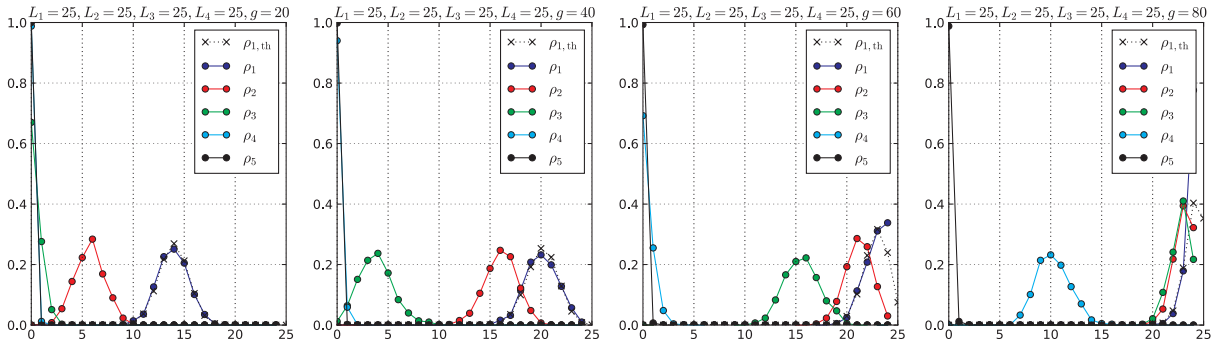


Figure 19. Case  $n_v = 4$ , distribution of  $\rho_1$  and histograms of  $\rho_1 \dots \rho_5$  for different values of  $g$  for  $L_1 = \dots = L_4 = 25$ .

of branches to consider at level  $n_v + 1$  in the decoding tree. Considering a larger value of  $n_v$  becomes then interesting from a complexity point of view. This phenomenon appears when  $g \geq 30$  for  $n_v = 1$ , when  $g \geq 80$  for  $n_v = 2$  and does not appear for larger values of  $n_v$ .

The proposed NeCoRPIA scheme is thus able to perform NC without coordination between agents. With  $n_v = 2$ , compared to classical NC, generations of 60 packets may be considered with a header overhead of 66 %, a vanishing decoding error probability, and a decoding complexity about 10 times that of Gaussian elimination. With  $n_v = 3$ , generations of 80 packets may be considered, leading to a header overhead of 25 %, but a decoding complexity about 100 times that of Gaussian elimination.

## VII. CONCLUSIONS

This paper presents NeCoRPIA, a NC algorithm with random packet index assignment. This technique is well-suited to data collection using MCS, as it does not require any prior agreement on the NC vectors, which are chosen randomly. As a consequence, different packets



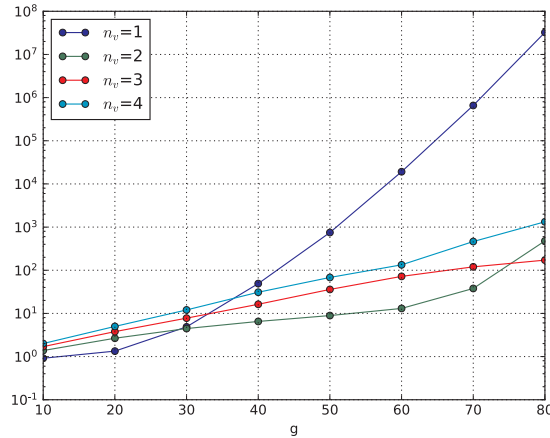


Figure 20. Evolution of the experimental values of the ratio of the decoding complexity of DeRPIA-SLE and DeRPIA-LUT with respect to the complexity of a simple RREF transformation as a function of  $g$  for different values of  $n_v$ .

may share the same coding vector, leading to a collision, and to the impossibility to perform decoding with standard Gaussian elimination. Collisions are more frequent when the size of the generation increases. A branch-and-prune approach is adapted to decode in presence of collisions. This approach is efficient in presence of a low number of collisions. To reduce the number of collisions, the NC vector is split into subvectors. Each packet is then assigned a set of NC subvectors.

A detailed analysis of the decoding complexity and of the probability of decoding error shows the potential of this approach: when a NC of  $L = 100$  elements in  $\mathbb{F}_2$  is split into 2 NC subvectors, generations of about 60 packets may be considered with a decoding complexity that is about 10 times that of plain network decoding. When the NC vector is split into 4 subvectors, generations of about 80 packets may be considered.

This approach may also be interesting in the context of COPE, where a relatively long hash is used in the NC header to identify the packets in the mixtures and prevent any collision. Since collisions may be solved using the proposed approach, shorter hash may be considered, reducing the NC overhead.

## REFERENCES

- [1] R. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: current state and future challenges,” *Communications Magazine, IEEE*, vol. 49, pp. 32–39, November 2011.
- [2] P. Mohan, V. Padmanabhan, and R. Ramjee, “Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones,” in *ACM Sensys*, November 2008. Raleigh, NC.

- [3] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Comm. Mag.*, vol. 47, no. 12, pp. 42–49, 2009.
- [4] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, pp. 1204–1216, July 2000.
- [5] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, "Efficient broadcasting using network coding," *IEEE/ACM Trans. Netw.*, vol. 16, pp. 450–463, Apr. 2008.
- [6] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," *Networking, IEEE/ACM Transactions on*, vol. 16, pp. 497–510, June 2008.
- [7] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '07, (New York, NY, USA), pp. 169–180, ACM, 2007.
- [8] T. Ho, R. Koetter, M. Médard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. of IEEE Int. Symp. Inf. Theory*, (Kanagawa, Japan), June 2003.
- [9] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. of Allerton Conf. on Commun. Control and Comput.*, (Monticello, IL, USA), Oct. 2003.
- [10] M. Jafari, L. Keller, C. Fragouli, and K. Argyraki, "Compressed network coding vectors," in *Proc. of IEEE Int. Symp. Inf. Theory*, (Seoul, Republic of Korea), pp. 109–113, June 2009.
- [11] N. Thomos and P. Frossard, "Toward one symbol network coding vectors," *IEEE Commun. Lett.*, vol. 16, no. 11, pp. 1860–1863, 2012.
- [12] C. Greco, M. Kieffer, and C. Adjih, "NeCoRPIA: Network coding with random packet-index assignment for mobile crowdsensing," in *Proc. IEEE International Conference on Communications*, pp. 6338–6344, June 2015.
- [13] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*, November 2009.
- [14] C. Adjih, S. Y. Cho, and P. Jacquet, "Near Optimal Broadcast with Network Coding in Large Sensor Networks," in *Proc. Workshop on Inf. Th. for Sens. Netw.*, June 2007.
- [15] J. Heide, M. V. Pedersen, F. H. Fitzek, and M. Médard, "On Code Parameters and Coding Vector Representation for Practical RLNC," in *Proc. IEEE ICC*, pp. 1–5, IEEE, 2011.
- [16] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press, 1st ed., 2010.
- [17] P. Comon, "Independent component analysis, a new concept?," *Signal Proc. (Elsevier Science)*, vol. 36, pp. 287–314, Apr. 1994.
- [18] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, pp. 411–430, May 2000.
- [19] A. Yeredor, "Independent component analysis over galois fields of prime order," *IEEE Trans. Inf. Theory*, vol. 57, pp. 5342–5359, Aug. 2011.
- [20] H. W. Gutch, P. Gruber, A. Yeredor, and F. J. Theis, "ICA over finite field—separability and algorithms," *Signal Proc. (Elsevier Science)*, vol. 92, no. 8, pp. 1796–1808, 2012.
- [21] I.-D. Nemoianu, C. Greco, M. Cagnazzo, and B. Pesquet-Popescu, "On a hashing-based enhancement of source separation algorithms over finite fields for network coding applications," *IEEE Trans. Multimedia*, vol. 16, no. 7, pp. 2011–2024, 2014.
- [22] C. Greco, M. Kieffer, C. Adjih, and B. Pesquet-Popescu, "PANDA: A protocol-assisted network decoding algorithm," in *Proc. of IEEE Int. Symp. on Network Cod.*, (Aalborg, Denmark), pp. 1–6, June 2014.

- [23] L. Holst, "On birthday, collectors, occupancy and other classical urn problems," *International Statistical Review*, vol. 54, no. 1, pp. 15–27, 1986.
- [24] C. Adjih, S. Cho, and E. Baccelli, "Broadcast With Network Coding: DRAGONCAST," IRTF Internet Draft – work in progress 01, individual, July 2013.
- [25] M. Abramowitz and I. A. Stegun, eds., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, ch. Stirling Numbers of the Second Kind, pp. 824–825. New-York: Dover, 1972.
- [26] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT Press, 2009.
- [27] J. D. Touch, "Performance analysis of md5," in *ACM SIGCOMM Computer Communication Review*, vol. 25, pp. 77–86, ACM, 1995.

## APPENDIX

### A. Arithmetic complexity of DeRPIA-SLE

1) *Arithmetic complexity for intermediate branches:* Consider Level  $\ell = 1$ . For each of the  $L_1$  canonical vectors  $\mathbf{e}_{j_1} \in \mathbb{F}_q^{L_1}$ , finding  $\mathbf{w}_1 \in \mathbb{F}_q^{\rho_1}$  satisfying (13) takes at most  $\rho_1 L_1$  operations, since, according to Theorem 4, one searches for a row of  $\mathbf{B}_{11} \in \mathbb{F}_q^{\rho_1 \times L_1}$  equal to  $\mathbf{e}_{j_1}$ . The arithmetic complexity to get all branches at Level 1 is thus upper-bounded by

$$K(1) = L_1 \rho_1 L_1. \quad (36)$$

Consider Level  $\ell$  with  $1 < \ell \leq n_v$ . For each branch associated to a candidate decoding vector  $(\mathbf{w}_1, \dots, \mathbf{w}_{\ell-1})$ , one first evaluates  $-(\mathbf{w}_1 \mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1} \mathbf{B}_{\ell-1,\ell})$  which takes  $K_m(\rho_1 L_\ell + \dots + \rho_{\ell-1} L_\ell) + (\ell - 1) L_\ell$  operations, the last term  $(\ell - 1) L_\ell$  accounting for the additions of the vector-matrix products and the final sign change.

Then, for each of the  $L_\ell$  canonical vectors  $\mathbf{e}_{j_\ell} \in \mathbb{F}_q^{L_\ell}$ , evaluating  $\mathbf{e}_{j_\ell} - (\mathbf{w}_1 \mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1} \mathbf{B}_{\ell-1,\ell})$  needs a single addition, and finding  $\mathbf{w}_\ell$  satisfying (13) takes at most  $\rho_\ell L_\ell$  operations, since one searches for a row of  $\mathbf{B}_{\ell\ell}$  equal to  $\mathbf{e}_{j_\ell} - (\mathbf{w}_1 \mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1} \mathbf{B}_{\ell-1,\ell})$ . Additionally, one has to verify whether  $\mathbf{w}_\ell = \mathbf{0}$  is satisfying, which requires  $L_\ell$  operations.

At Level  $\ell$ , the arithmetic complexity, for a given candidate  $(\mathbf{w}_1, \dots, \mathbf{w}_{\ell-1})$ , to find all candidates  $(\mathbf{w}_1, \dots, \mathbf{w}_\ell)$  satisfying (8) is thus upper-bounded by

$$K(\ell) = K_m(\rho_1 L_\ell + \dots + \rho_{\ell-1} L_\ell) + (\ell - 1) L_\ell + L_\ell (1 + \rho_\ell L_\ell + L_\ell) \quad (37)$$

$$= K_m L_\ell (\rho_1 + \dots + \rho_{\ell-1}) + L_\ell (\ell + (\rho_\ell + 1) L_\ell) \quad (38)$$

2) *Arithmetic complexity for terminal branches:* Consider now Level  $n_v + 1$ . For each candidate  $(\mathbf{w}_1, \dots, \mathbf{w}_{n_v})$ , one has first to compute  $\mathbf{w}_1 \mathbf{C}_1 + \dots + \mathbf{w}_{n_v} \mathbf{C}_{n_v}$ , which requires  $K_m(\rho_1 L_p + \rho_2 L_p + \dots + \rho_{n_v} L_p) + (n_v - 1) L_p \leq K_m g L_p + (n_v - 1) L_p$  operations. Then for each candidate  $\mathbf{w}_{n_v+1} \in \mathbb{F}_q^{\rho_{n_v+1}}$ , the evaluation of  $\mathbf{w}_1 \mathbf{C}_1 + \dots + \mathbf{w}_{n_v} \mathbf{C}_{n_v} + \mathbf{w}_{n_v+1} \mathbf{C}_{n_v+1}$  and the

checksum verification cost  $K_m \rho_{n_v+1} L_p + L_p + K_c L_p$  operations. The arithmetic complexity, for a given candidate  $(\mathbf{w}_1, \dots, \mathbf{w}_{n_v})$ , to find all solutions  $(\mathbf{w}_1, \dots, \mathbf{w}_{n_v+1})$  is thus upper-bounded by

$$K(n_v + 1) = K_m g L_p + (n_v - 1) L_p + q^{\rho_{n_v+1}} (K_m \rho_{n_v+1} L_p + L_p + K_c L_p). \quad (39)$$

3) *Total arithmetic complexity of DeRPIA-SLE*: To upper bound the arithmetic complexity of the tree traversal algorithm when the number of encoding vectors is  $n_v$ , one combines (15), (37), and (39) to get (18) with  $N_b(0) = 1$ . In the case  $n_v = 1$ , (20) follows directly from (18).

One first evaluates the complexity of the look-up table construction with Algorithm 2a. The look-up table is built once, after the RREF evaluation. The worst-case complexity is evaluated, assuming that for each row vector  $\mathbf{u}$  of  $\mathbf{B}_{\ell\ell}$ , the resulting vector  $\bar{\mathbf{u}}$  is added to  $\Pi_\ell$ .

For each of the  $\rho_\ell$  lines  $\mathbf{u}$  of  $\mathbf{B}_{\ell\ell}$ , the identification of the index of its pivot column takes at most  $L_\ell$  operations. The evaluation of  $\bar{\mathbf{u}}$  takes one operation. Then determining whether  $\bar{\mathbf{u}} \in \Pi_\ell$  takes no operation for the first vector ( $\Pi_\ell$  is empty), at most  $L_\ell$  operations for the second vector, at most  $2L_\ell$  operations for the third vector, and at most  $(\rho_\ell - 1) L_\ell$  operations for the last vector. The number of operations required in this step is upper bounded by

$$\begin{aligned} K_{\text{LU},1}(\ell) &= \rho_\ell (L_\ell + 1) + 0 + L_\ell + 2L_\ell + \dots + (\rho_\ell - 1) L_\ell \\ &= \rho_\ell + L_\ell \frac{\rho_\ell (\rho_\ell + 1)}{2}. \end{aligned} \quad (40)$$

Then the canonical vectors  $\mathbf{e}_i \in \mathbb{F}_q^{\rho_\ell}, i = 1, \dots, \rho_\ell$  have to be partitionned into the various  $\mathcal{S}_\ell(\mathbf{v}), \mathbf{v} \in \Pi_\ell$ . This is done by considering again each of the  $\rho_\ell$  lines  $\mathbf{u}$  of  $\mathbf{B}_{\ell\ell}$ , evaluating  $\mathbf{u} - \mathbf{e}_{\gamma(\mathbf{u})}$ , which needs up to  $L_\ell + 1$  operations. Then determining the vectors  $\mathbf{v} \in \Pi_\ell$  such that  $\mathbf{u} - \mathbf{e}_{\gamma(\mathbf{u})} = \mathbf{v}$  requires at most  $\rho_\ell L_\ell$  operations, since  $\Pi_\ell$  contains at most  $\rho_\ell$  vectors of  $L_\ell$  elements. The number of operations required in this partitionning is upper bounded by

$$K_{\text{LU},2}(\ell) = \rho_\ell (L_\ell + 1 + \rho_\ell L_\ell). \quad (41)$$

Considering a satisfying  $(\mathbf{w}_1, \dots, \mathbf{w}_{\ell-1})$  at Level  $\ell - 1$ , the search complexity to find some satisfying  $(\mathbf{w}_1, \dots, \mathbf{w}_{\ell-1}, \mathbf{w}_\ell)$  at Level  $\ell$  in Algorithm 2b is now evaluated. The evaluation of  $\mathbf{v} = -(\mathbf{w}_1 \mathbf{B}_{1,\ell} + \dots + \mathbf{w}_{\ell-1} \mathbf{B}_{\ell-1,\ell})$  takes  $K_m(\rho_1 L_\ell + \dots + \rho_{\ell-1} L_\ell) + (\ell - 1) L_\ell$  operations. Then, determining whether  $\mathbf{w}_\ell = \mathbf{0}$  satisfies (8), *i.e.*, whether  $\mathbf{v} + \mathbf{e}_{j_\ell} = \mathbf{0}$  for some canonical vector  $\mathbf{e}_{j_\ell} \in \mathbb{F}_q^{L_\ell}$ , can be made checking whether  $\mathbf{v}$  contains a single non-zero entry in  $L_\ell$  operations. Finally, the look-up of  $\mathbf{v} \in \Pi_\ell$  takes at most  $\rho_\ell L_\ell$  operations. In summary, the

number of operations required for this part of the algorithm is

$$\begin{aligned} K_{\text{LU},3}(\ell) &= K_m(\rho_1 L_\ell + \dots + \rho_{\ell-1} L_\ell) + (\ell - 1) L_\ell + L_\ell + \rho_\ell L_\ell \\ &= K_m L_\ell (\rho_1 + \dots + \rho_{\ell-1}) + L_\ell (\ell + \rho_\ell). \end{aligned} \quad (42)$$

Compared to the expression of  $K(\ell)$  given by (37), which is quadratic in  $L_\ell$ ,  $K_{\text{LU},3}(\ell)$  is linear in  $L_\ell$ .

The complexity of DeRPIA-LUT, given by (21), when the number of encoding vectors is  $n_v$ , is then obtained combining the results of Corollary 8 with (40), (42), (42), and (39), since Algorithm 2b is not used at Level  $n_v + 1$ .