Overview of Turbo-Code Reconstruction Techniques

Johann Barbier[1,2]
[1]Département de Cryptologie
Centre d'ÉLectronique de l'ARmement
35 174 Bruz Cedex, France
Email: johann.barbier@esiea-ouest.fr

Éric Filiol[2]
[2]Laboratoire de Cryptologie et Virologie Opérationnelles
École Supérieure d'Informatique, Électronique, Automatique
53 000 Laval, France
Email: eric.filiol@esiea-ouest.fr

***Abstract*—In this paper we analyze different techniques to blindly recover the parameters of turbo-code encoders with only the knowledge of noisy eavesdropped binary stream. We compare different strategies to reconstruct convolutional encoders, particularly when both are recursive systematic coders. We explain the intrinsic indetermination due to these techniques but also how to overcome such an issue in the case of turbo-code encoders. Moreover, we generalize Burel and *al.*'s particular reconstruction of the second $(n, 1)$-encoder for (n, k)-encoders.**

I. Introduction

Turbo-codes were introduced in 1993 by C. Berrou, A. Glavieux and P. Thitimajshima [1] and became quickly a standard for error correction adapted to very noisy transmission channels. The general scheme of turbo-coders is composed of two parallel convolutional coders separated by a block interleaver. This scheme is represented in figure 1.

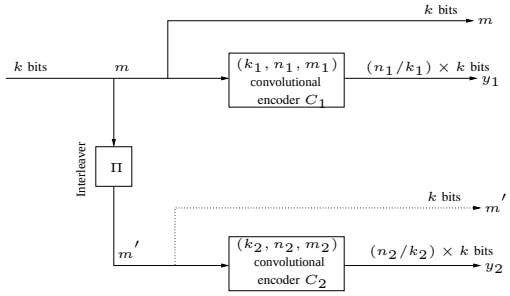

Fig. 1. Turbo-coder

In a non-cooperative context, a passive adversary eavesdrops the signal transmission exchanged by two legal users who keep secret all the parameters that they use to communicate. The adversary has no *a priori* knowledge about these parameters and has to recover them in order to have access to the information shared by the legal users. In this paper, we suppose that he has access to the bits stream. We also suppose that this binary stream is the output of a turbo-code encoder. The goal of the adversary is then to guess the convolutional coders but also the interleaver. We mainly focus here on the problem of recovering the parameters of convolutional encoders. We extend the work initialized by J. Barbier [2], [3], [4] who dealt with polynomial encoders and revisited by R. Gautier and *al.* [5] for Recursive Systematic Coders (RSC) in a perfect channel.

This paper is organized as follows. We first recall the principles of the convolutional codes reconstruction. We also discuss about the intrinsic indetermination of such techniques and explain how to directly recover the original coder and so resolve some of the indetermination. In section III, we detail state-of-art techniques to reconstruct linear block codes and describe the context in which they could be adapted to reconstruct convolutional codes. We present a complexity analysis of Gautier and *al.*'s adaptation and explain why this method is much less efficient than the standard one. Finally, in section IV we present Gautier and *al.*'s technique to recover the second convolutional encoder when the encoder is a $(n, 1)$-coder and when its output is partially punctured under the hypothesis of a perfect channel. We propose a generalization of this method to (n, k)-coders and noisy channels.

II. Reconstruction of Convolutional Codes

The convolutional code reconstruction problem has been settled and partially solved for $(n, 1)$-codes and noiseless channels, by B. Rice [6] in 1995. Then É. Filiol designed a technique to reconstruct general convolutional codes in noisy channels [7], [8], [9]. His work was the starting point of J. Barbier's algorithm which drastically improves the reconstruction complexity [10], [4] but also the probability of detection by introducing the use of linear block code reconstruction algorithms to solve linear systems with corrupted coefficients [4], [11].

A. Principles of the convolutional codes reconstruction

Convolutional (n, k, K)-coders are error-correcting stream coders which output n bits for every k input bits. Each bit of the output is a linear combination of the $k \times (K-1)$ previous input bits and the k current input bits. K is traditionally called the *constrain length*. Such encoders can be implemented by shift register circuits. If we map the input data stream to the coefficients of k Laurent's serials, $m_i(x)$ and the output coded stream to the coefficients of n Laurent's serials, $c_j(x)$, then using the algebraic approach, we express $c_j(x)$ from $m_i(x)$ by

$$\begin{cases} m_1(x)g_{1,1}(x) + \cdots + m_k(x)g_{1,k}(x) &= c_1(x), \\ \vdots & \vdots \\ m_1(x)g_{n,1}(x) + \cdots + m_k(x)g_{n,k}(x) &= c_n(x), \end{cases} \quad (1)$$

where the $g_{i,j}(x)$ are rational fractions with binary coefficients. These rational fractions entirely define the coder by its generator matrix $G(x) = [g_{i,j}(x)]_{i,j}$. The interested reader can find a good introduction of convolutional codes through the scope of the algebraic approach in [12].

Let F be $GF(2)$, the Galois field of characteristic 2. In the case of Non Recursive Coders (NRC), $g_{i,j}(x)$ are polynomials of degree at most K. Since $c_i(x)$ and $m_i(x)$ belong to the field of Laurent's serials, there exists an infinity of couples $(m(x), G(x))$ which satisfy equation 1. Now, we present how to recover a particular generator matrix (*canonical matrix*) for perfect channels. A canonical matrix is a polynomial generator matrix for which the maximum degree of its minors is at small as possible. We first consider the $(n-k)$ $(k+1,k)$-sub-coder, $G_i(x)$ defined by the first k rows of $G(x)$ plus the $(k+i)th$. For each sub-coder, we build the matrix

$$P_i(x) = \begin{pmatrix} g_{1,1}(x) & \dots & g_{1,k}(x) & c_1(x) \\ \vdots & & \vdots & \vdots \\ g_{k,1}(x) & \dots & g_{k,k}(c) & c_k(x) \\ g_{k+i,1}(x) & \dots & g_{k+i,k}(c) & c_{k+i}(x) \end{pmatrix}. \quad (2)$$

The determinant of $P_i(x)$ equals 0, since $c_j(x)$ are linear combinations of $g_{i,j}(x)$. Then, by developing it, we obtain

$$\sum_{j=1\dots k,k+i} c_j(x)\Delta_{j,i}(x) = 0, \quad (3)$$

where $\Delta_{j,i}(x)$ is the minor associated with $c_j(x)$ in the matrix $P_i(x)$. Moreover, it is easy to see that the $\Delta_{i,j}(x)$ which are solutions of the equations 3 for $i = 1\dots(n-k)$ belong to the same $F[x]$-module of dimension 1. Let D be the degree of the convolutional code, which is also the highest degree of minors of a canonical encoder, nN be the number of intercepted bits, then we define

$$C^{(j)}(N) = \begin{pmatrix} c_j^d & \dots & c_j^0 \\ c_j^{d+1} & \dots & c_j^1 \\ \vdots & & \vdots \\ c_j^N & \dots & c_j^{(N-d)} \end{pmatrix}, \; j = 1\dots n$$

and

$$C_i(N) = \left(\; C^{(1)}(N) \; \middle| \; \dots \; \middle| \; C^{(k)}(N) \; \middle| \; C^{(k+i)}(N) \; \right), \quad (4)$$

$i = 1\dots(n-k)$. To recover the minors, then we have to solve the linear systems $C_i(N)\Delta = 0$ for $i = 1\dots(n-k)$. When the channel is perfect, we just have to compute $Ker(C_i(N))$ and when the channel is noisy we adapt linear block code reconstruction algorithms as proposed in [4], [11]. In [4], J. Barbier explains how to construct a generator of the $F[x]$-module of solution and proves that this generator is associated with a canonical generator matrix. Finally, to recover the polynomials we solve the following systems as previously. $\forall i = 1\dots(n-k)$, $\forall j = 1\dots k$,

$$\sum_{l=1\dots k,k+i} \Delta_{l,i}(x)g_{l,j}(x) = 0. \quad (5)$$

Since k, n, D are unknown, we have to guess them iteratively. When the tested values are not correct, with a high probability, $Ker(C_i(N)) = \emptyset$, and when k, n are correct and $D+1 \leq d$, we obtain a solution.

B. On the impossibility of finding the original coder

The technique presented in the last paragraph focus on finding a canonical matrix $G(x)$, but all the matrices which are $F[x]$-equivalent may have generated the same output $(c_1(x), \dots, c_n(x))$. This can also be well illustrated by the equation 3. Indeed, the set of solutions is a $F[x]$-module of length 1. So, with no *a priori* knowledge, there is no way to guess if the coded sequence has been generated by a canonical matrix or by an equivalent one.

Moreover, a canonical matrix is not unique [12]. To be convinced of that, just mirror the left part of the system in equation 1. The message $(m_k(x), \dots, m_1(x))$ and the matrix $G^{'}(x) = [g_{1,k-j}(x)]$ also verifies the equation 1. This is also true when applying any permutation of columns of the left part of the system. Finally there exists at least $(k!)$ different couples $(m(x), G(x))$, where $G(x)$ is canonical and which verifies equation 1.

C. Reconstruction of a Recursive Systematic Coders

In [5], the authors use first block code reconstruction algorithm to recover one polynomial generator matrix $G_{NRNSC}(x) = [g_{i,j}(x)]$ of a convolutional code and then compute

$$G_{RSC}(x) = Q^{-1}(x)G_{NRNSC}(x)\,, \quad (6)$$

where $Q(x) = [g_{i,j}(x)]_{i,j=1\dots k}$, in order to obtain a systematic generator matrix. Two main drawbacks appear using this method. First, the computational cost of inverting $Q(x)$ with coefficients in $F(X)$ is basically $\mathcal{O}(kK^k)$. Secondly, because of the huge number of polynomial matrices which generate the coded sequence, there is also a huge number of acceptable $G_{RSC}(x)$ doing that way.

Now, we suppose that the coded sequence $c(x)$ has been generated by a systematic coder and so $G(X)$ can be written

$$G(x) = \begin{pmatrix} & I_k & \\ \frac{p_{k+1,1}(x)}{q_{k+1,1}(x)} & \dots & \frac{p_{k+1,k}(x)}{q_{k+1,k}(x)} \\ \vdots & & \vdots \\ \frac{p_{n,1}(x)}{q_{n,1}(x)} & \dots & \frac{p_{n,k}(x)}{q_{n,k}(x)} \end{pmatrix}. \quad (7)$$

Since $G(x)$ is systematic, this is the only systematic generator matrix which generates $c(x)$. Then $q(x)G(x) = GCM(q_{i,j}(x))G(x)$ is the only canonical generator matrix which generates $c(x)$ such that the upper part can be written as $p(x)I_k$. Finally, using Filiol and Barbier's algorithm, we obtain the set of possible canonical matrices which generates $c(x)$. Among all, $q(x)G(x)$ is the only one with the upper part of the form $p(x)I_k$. Using this algorithm, we are able to determine the right systematic coder without any additional computational cost.

III. Adaptation of Linear Block Codes Reconstruction

Two main approaches have been introduced first by G. Planquette [13] to recover linear block codes. The first one consists in assuming properties on parity checks and then validating them using statistical tests. The second approach is based on the study of the rank of a matrix, the interception matrix, built with the intercepted bits stream. It has been rediscovered later by G. Burel and R. Gautier [14]. When the bits stream is noisy, two strategies have been proposed to adapt the *rank criterion*. First, A. Valembois [15], [16] proposed to look for small Hamming weight codewords of the code generated by the columns of the interception matrix. Then, M. Cluzeau [17], [18], [19] improved the detection of such codewords by adapting the Canteaut-Chabaud algorithm [20] and introducing error correction during the reconstruction process for LDPC. The second strategy, introduced by G. Sicot and S. Houcke [21], [22], [23], [24] is a randomization of the Gauss algorithm. J. Barbier gave an algebraic analysis of this algorithm [4], [11]. Such algorithms are also dedicated to solve linear binary systems with erroneous coefficients.

A. Principles of the block codes reconstruction

In the context of block code reconstruction, the intercepted codewords $(\tilde{c_i})_i$ can be written as $\tilde{c_i} = Gm_i + e_i = c_i + e_i$ where $G = [g_{i,j}]_{i,j}$ of size $n \times k$ and such that $g_{i,j} \in F$, is the generator matrix of the code, m_i are the information words and e_i are vectors of length n such that the coefficients which equal one represent the noisy bits of $\tilde{c_i}$. To reconstruct the code associated with G, we preferentially reconstruct its dual code $\mathcal{H}$, that is to find a basis $(h_j)_j$ of $\mathcal{H}$, *i.e.* vectors such that for all i and j, $< c_i, h_j >=< h_j, c_i >= 0$. To achieve this, we build the interception matrix $\tilde{C}(n_e, d_e)$, by filling it from top left to bottom right using the intercepted bits. As the adversary has non *a priori* knowledge, the first intercepted bit is not necessary the first bit of a codeword. We denote by d, the *desynchronization parameter* which represents the index of the first intercepted bit in the first intercepted codeword. $\tilde{C}(n_e, d_e)$ has n_e columns and d_e is an estimation of d. In order that the first coefficient of the intercepted matrix be also the first bit of an intercepted codeword, the adversary skips the first $(n_e - d_e)$ intercepted bits. That is the case when $d_e = d$. First, let us write

$$\tilde{C}(n_e, d_e) = C(n_e, d_e) + E(n_e, d_e)\,, \tag{8}$$

where $C(n_e, d_e)$ and $E(n_e, d_e)$ are filled in the same way as $\tilde{C}(n_e, d_e)$ using respectively the bits of $(c_i)_i$ and $(e_i)_i$. We recall that $Ker(C(n, d)) = \mathcal{H}$, with a probability close to one [4], [11]; then the block code reconstruction problem is equivalent to compute $Ker(C(n, d))$ observing $\tilde{C}(n_e, d_e)$ and guessing n and d. Under the realistic hypothesis that $C(n_e, d_e)$ behaves like a random binary matrix when $n_e \neq \alpha n$, $\alpha \in \mathbb{N}^*$, G. Sicot, S. Houcke and J. Barbier [4], [11] proved that with a probability close to 1,

$$\frac{k}{n} = \frac{\text{rk}(C(n,d))}{n} \leq \frac{\text{rk}(C(n,d_e))}{n} \leq \frac{\text{rk}(C(n_e,d_e))}{n_e} = 1, \tag{9}$$

$\forall\, d_e, n_e \neq n$. Their analysis leads to the already known *rank criterion*,

$$(n, d) = \underset{n_e, d_e}{\text{Argmin}} \left(\frac{\text{rk}\ (C(n_e, d_e))}{n_e} \right). \tag{10}$$

The estimation of $\frac{\text{rk}(C(n_e,d_e))}{n_e}$ and $\mathcal{H}$ using $\tilde{C}(n_e, d_e)$ can be achieved by looking for small Hamming weight codewords in the code generated by the columns of the interception matrix as proposed by G. Planquette [13], A. Valembois [15], [16] and M. Cluzeau [17], [18], [19] or by an adaptation of the Gauss algorithm proposed by G. Sicot and S. Houcke [21], [22], [23], [24].

B. Adaptation to reconstruct convolutional codes

To analyze the adaptation proposed in [5], one should notice that the different bits streams $(c_i(x))_{i=1\ldots n}$ are not separated and grouped to build sub-encoders, so the system must contains all the sub-systems which verify equations 3. We conclude that for all $n_e < n(D+1)$, $C(n_e, d_e)$ is full rank with a probability close to 1. The same arguments as those detailed in [4], [11] lead us to conclude that for $n_e \neq \alpha n + n(D+1)$, $\alpha \in \mathbb{N}$, $C(n_e, d_e)$ is full rank with a probability close to one.

Now, we suppose that $n_e = \alpha n + n(D+1)$. For $\alpha = 0$, all the equations 3 are verified in the same time. Resolving the system, we obtain $(n-k)$ independent solutions which define the generator of the $F[x]$-module of solutions. Then, for each additional n columns of the interception, another $(n-k)$ independent solutions are obtained. We can deduced them by multiplying by x the solutions obtained without these new n columns. They correspond to solutions of higher degree in the $F[x]$-module. So we can conclude that

$$\text{rank}(C(n_e, d_e)) = \alpha k + n(D+1) - (n-k) = (\alpha - 1)k + nD, \tag{11}$$

with a probability close to 1, for all $n_e = n(D+1)+\alpha n, \alpha \in \mathbb{N}$ and $\text{rank}(C(n_e, d_e)) = n_e$ otherwise. This expression puts right the formula of the rank proposed in [5]. When we observe two consecutive values n_e^1 and n_e^2 such that the interception matrix is not full rank, we deduce $n = |n_e^2 - n_e^1|$. Moreover the slope s of the rank for $n_e = n(D+1) + \alpha n$ is exactly the rate of the code; then we obtain $k = ns$. Finally, for $\alpha = 1$ and a rank R measured, we conclude $D = R/n$.
One should also notice that the convolutional encoders are stream encoders, so each sub-stream of length $n(D+1)$ may be used to build the interception matrix. For that kind of coders, we need not to find any synchronization parameter d.

C. Complexity analysis

First, we deal with a perfect channel. In order to observe the first rank deficiency, we need to compute the rank of the interception matrix for n_e from 1 to $n(D+1)$. The complexity of this adaptation is then

$$\mathcal{O}\left(\sum_{n_e=2}^{n(D+1)} n_e^3 \right) = \mathcal{O}(n^4 D^4). \tag{12}$$

Now, if we use Filiol and Barbier's algorithm [4], when $n_e \neq n$, an average of one Gauss iteration is need (with a probability close to one the interception matrices behave like random matrices). Each triplet (n_e, k_e, D_e) is tested. Then, the complexity of Filiol and Barbier's algorithm [2], [4] is

$$\mathcal{O}(n^5 D^4). \tag{13}$$

At the first sight, Gautier and *al.*'s strategy seems to be a little bit more efficient for perfect channels. Nevertheless, the systems to be solved in Filiol and Barbier's algorithm are composed of Hankel sub-systems, that is no more the case in the proposed technique. This particular form of the sub-systems drives us to foresee a great improvement of the complexity by replacing the Gauss algorithm with algorithms dedicated to solve Toeplitz systems. For instance, the use of a Fast Fourier Transform [25] reaches $\mathcal{O}(n \log n)$.

Now, we suppose that the channel is noisy. In that case, Gautier and *al.*'s strategy consists in using Sicot and Houcke's algorithm [21], [22], [23], [24] to dealt with corrupted coefficients in the systems that they solve. The main idea of this algorithm is to randomize the Gauss algorithm by randomly mixing the rows of the interception matrix. Then, for each mixed interception matrix they test the vectors of the new basis output by the Gauss algorithm. J. Barbier proved [4] that

$$\lim_{N\to\infty} \mathcal{P}r\left(v \in \mathcal{H} | w(\tilde{C}(n,d)v) \leq \gamma N\right) = 1, \tag{14}$$

where $w(.)$ is the Hamming weight, γ a threshold close to 0 and N the number of rows of the interception matrix. So, if we find a vector v such that $w(\tilde{C}(n,d)v) \leq \gamma N$, then with a high probability, v is a solution of the considered system. Now, with $2n(D+1)$ intercepted bits, we generate $(D+2)$ rows for the systems in Filiol and Barbier's algorithm opposed to only 2 when applying the technique proposed in [5]. One main issue is to find vector of small weight. Let us denote

$$P(v) = \left(\frac{1 + (1-2\varepsilon)^{w(v)}}{2}\right)^{n_e}, \tag{15}$$

where ε is Bit Error Rate (BER). Then, the analysis of Sicot and Houcke's algorithm points out that the probability $\mathcal{P}_{det}(v,N)$ to find such a vector v is bounded by [4]

$$P(v)^{n_e} \geq \lim_{N\to\infty} \mathcal{P}_{det}(v,N) \geq P(v)^{n_e}\left(1 - \left(\frac{1-P(v)}{\gamma}\right)^2\right), \tag{16}$$

when $n_e = n(D+1)+\alpha n$. That implies that the probability of detection of a solution exponentially tends towards 0 when the size of the system increases. As systems in Filiol and Barbier's algorithm are $r = n/(k+1)$ times smaller than those used in block code reconstruction, we can conclude that Filiol and Barbier's algorithm is at least $P(v)^{-r}$ times more efficient than applying Gautier and *al.*'s technique (for a fixed BER, N and a given solution v).

IV. Generalization of Gautier and *al.*'s Technique

To reconstruct the second (k,n,D)-convolutional code, we reconstruct $(n-k)$ $(k,k+1)$-sub-encoders as explained in section II. The convolutional coder C_1 (see figure 1) can be reconstructed directly using Filiol and Barbier's algorithm as detailed in paragraph II-C. Then, at that point, k is known and we make the hypothesis that C_2 is an RSC and that the systematic part $m'(x)$ is punctured. Let us denote l_e the length of the interleaver Π.

A. Reconstruction of the second (k,n,D)-convolutional codes

1) $l_e = k(D+1)$: The interleaver Π can be entirely defined by a permutation $\sigma : [0, l_e-1] \rightarrow [0, l_e-1]$. We define $\rho = \sigma^{-1}$. The permutations σ and ρ can be extended in the same manner to $\mathbb{N}$ as follows. $\forall \alpha \in \mathbb{N}$ and $\forall i \in [0, l_e-1]$,

$$\sigma(\alpha l_e + i) = \alpha l_e + \sigma(i). \tag{17}$$

First, we express $m = m^0 m^1 m^2 \cdots = m_1^0 m_2^0 \ldots m_k^0 m_1^1 \ldots m_k^1 \ldots$. The interleaved message $m^{'}$ can also be written this way. Then, we rewrite equations 4 with $\forall i \in [1,k]$, $c_i(x) = m_i^{'}(x)$ and solve the systems $\forall i = 1 \ldots (n-k)$, $C_i(N)X =$

$$\left(C^{(1)}(N) | \ldots | C^{(k)}(N) | C^{(k+i)}(N)\right) X = 0, \tag{18}$$

where N is the number of rows of the matrices. Without loss of generality, N is considered as constant for better readability. Using the definition, we can expressed $C^{(i)}(N)$ with m and ρ. $\forall i = 1 \ldots k$, $C^{(i)}(N) =$

$$\begin{pmatrix} m^{\rho(Dk+i-1)} & \ldots & m^{\rho(k+i-1)} & m^{\rho(i-1)} \\ m^{l_e+\rho(i-1)} & m^{\rho(Dk+i-1)} & \ldots & m^{\rho(k+i-1)} \\ \vdots & \vdots & & \vdots \\ m^{l_e+\rho(Dk+i-1)} & \ldots & m^{l_e+\rho(k+i-1)} & m^{l_e+\rho(i-1)} \\ \vdots & \vdots & \ldots & \vdots \end{pmatrix}.$$

Since the adversary has no knowledge about ρ, he is not able to build such matrices but one should notice that each row of $(C^{(1)}(N)|\ldots|C^{(k)}(N))$ contains exactly l_e consecutive bits of m. Taking advantage of this, we solve the system $C_i^{'}(N)X = \left(M(N)|C^{'(k+i)}(N)\right)X = 0$, where M is built with the rows of indexes $(\beta D+1), \beta \in \mathbb{N}$ in $\left(C^{(1)}(N)|\ldots|C^{(k)}(N)\right)$ and then applying σ to its columns. $C^{'(k+i)}(N)$ is filled with the rows of indexes $(\beta D+1), \beta \in \mathbb{N}$ in $C^{(k+i)}(N)$. Finally, the $(n-k)$ binary systems we have to solve are $\forall i = 1 \ldots (n-k)$,

$$\begin{pmatrix} m^{l_e-1} & \ldots & m^0 & c_{k+i}^D & \ldots & c_{k+i}^0 \\ m^{2l_e-1} & \ldots & m^{l_e} & c_{k+i}^{2D+1} & \ldots & c_{k+i}^{D+1} \\ \vdots & & \vdots & \vdots & & \vdots \end{pmatrix} X = 0. \tag{19}$$

Now, by construction, it is easy to see that if $S = (s_0, \ldots, s_{l_e+D})^T$ is a solution of the equation 18 then $S^{'} = (s_{\sigma(0)}, \ldots, s_{\sigma(l_e-1)}, s_{l_e}, \ldots, s_{l_e+D})^T$ is solution of the equation 19. If the upper part of the canonical generator matrix associated with the systematic coder C_2 is $p(x)I_k$ then $(s_{l_e}, \ldots, s_{l_e+D})$ are the coefficients of $x^i p^k(x)$, $i \geq 0$.

2) $l_e = \alpha k(D+1)$*:* It is easy to deduce valid systems from equation 19 by concatenating rows by α. In that case, the $(n-k)$ systems we have to solve are $\forall i = 1 \dots (n-k)$,

$$\begin{pmatrix} m^{l_e-1} & \dots & m^0 & c_{k+i}^{\alpha(D+1)-1} & \dots & c_{k+i}^0 \\ m^{2l_e-1} & \dots & m^{l_e} & c_{k+i}^{2\alpha(D+1-1)} & \dots & c_{k+i}^{\alpha(D+1)} \\ \vdots & & \vdots & \vdots & & \vdots \end{pmatrix} X = 0. \tag{20}$$

In the case $l_e \neq \alpha k(D+1)$, then we reduce it to the case where $l_e = \alpha k(D+1)$ by considering the interleaver of size $l_e^{'} = GCM(l_e, k(D+1))$. Such an interleaver can be modeled by the concatenation of $l_e^{'}/l_e$ interleavers Π. In that case, $\alpha = l_e^{'}/(k(D+1))$. Applying this technique, we are then able to reconstruct the second convolutional code whatever l_e, $k+1$ and n are. The systems that we point out extend the case where $k=1$ and $l_e = D+1$ which was the only one solved in [5]. In order to deal with noisy channels, we apply the Sicot and Houcke's algorithm to solve the systems that we define.

B. Complexity analysis

In order to build the $(n-k)$ systems 20, we have to guess α, D and n, since k is obtained when recovering C_1. First, we fix α, then we try to guess $n > k$ and finally we try different values of D. For each triplet (α, n, D) we solve an average of one system 20 excepted for correct estimations. In that case, $(n-k)$ systems have to be solved. The dimension of each system 20 is $(\alpha(k+1)(D+1))$ variables. Then, the computational complexity is

$$\mathcal{O}\left(\sum_{\alpha_e=1}^{\alpha}\sum_{n_e=k+1}^{n}\sum_{D_e=2}^{D}\left(\alpha_e(k+1)(D_e+1)\right)^3\right) = \mathcal{O}\left(nk^3(\alpha D)^4\right), \tag{21}$$

where $\alpha = GCM(l_e, k(D+1))/(k(D+1))$.

V. Conclusion

We recalled algorithms to reconstruct convolutional codes when the intercepted bits stream is corrupted by the channel. We detailed a technique to build a RSC from the output of the reconstruction algorithm proposed by É. Filiol and J. Barbier which is more efficient than the one proposed in [5]. Then, we described algorithms to reconstruct linear block codes and discussed about the way and the context to adapt them for convolutional codes reconstruction purposes. We also gave a complexity analysis which points out that it is much more efficient to directly apply standard algorithms than the adaptation proposed in [5] for recovering turbo-codes parameters. Finally, we generalized R. Gautier and *al.*'s technique for general (n,k)-encoders and for noisy channels. If we want to reconstruct the entire turbo-code by applying the presented techniques, we also need to be able to determine the encoder type and then to recover the interleaver. An algorithm has been already designed to reconstruct the interleaver [4], [3], [2] for particular turbo-code schemes. One direction could be to adapt such an algorithm. The type of the encoder is directly determined by the output of É. Filiol and J. Barbier's algorithm. One open problem is to find an algorithm to recover the entire interleaver for this generic turbo-code scheme. This is one part of our current work.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes." in *Proc. of ICC 1993*, Geneva, Switzerland, May 1993.

[2] J. Barbier and V. Camion, "Reconstruction de permutations." in *Proc. MAJECSTIC'03*, Marseille, France, Oct. 2003.

[3] J. Barbier, "Reconstruction of turbo-code encoders." in *Proc. SPIE Security and Defense, Space Communication Technologies Symposium*, vol. 5819, Orlando, FL, USA, Mar. 2005, pp. 463–473.

[4] ——, "Analyse de canaux de communication dans un contexte non-coopératif. Application aux codes correcteurs d'erreurs et à la stéganalyse." Ph.D. dissertation, École Polytechnique, Palaiseau, France, Nov. 2007.

[5] R. Gautier, M. Marazin, and G. Burel, "Blind recovery of the second convolutional encoder of turbo-code when its systematic outputs are punctured." in *Proc. of the 7th IEEE-Communication*, Bucharest, Romania, June 2008.

[6] B. Rice, "Determining the parameters of a rate $\frac{1}{n}$ convolutional encoder over GF(q)." in *Proc. Third International Conference on Finite Fields and Applications*, Glasgow, 1995.

[7] E. Filiol, "Techniques de reconstruction en cryptologie et théorie des codes." Ph.D. dissertation, École Polytechnique, Palaiseau, France, Mar. 2001.

[8] ——, "Reconstruction of punctured convolutional encoders." in *Proc. 2000 International Symposium on Information Theory and Applications*, T. Fujiwara, Ed. SITA and IEICE Publishing, 2000, pp. 4–7.

[9] ——, "Reconstruction of convolutional encoders over GF(q)." in *Proc. 6th IMA Conference on Cryptography and Coding*, ser. Lecture Notes in Computer Science, M. Darnell, Ed., no. 1355. Springer Verlag, 1997, pp. 100–110.

[10] J. Barbier, "Reconstruction des turbo-codes." Mémoire de DEA, École Polytechnique, Palaiseau, France, 2003.

[11] J. Barbier, G. Sicot, and S. Houcke, "Algebraic approach for the reconstruction of linear and convolutional error correcting codes," *International Journal of Applied Mathematics and Computer Sciences*, vol. 2, no. 3, pp. 113 – 118, 2006.

[12] R. McEliece, *Handbook of coding theory.* Elsevier Science, 1998, vol. 2, ch. 12, The algebraic theory of convolutional codes, pp. 1065–1138.

[13] G. Planquette, "Identification de trains binaires codés." Ph.D. dissertation, Université de Rennes I, France, Dec. 1996.

[14] G. Burel and R. Gautier, "Blind estimation of encoder and interleaver characteristics in a non cooperative context." in *Proc. IASTED International Conference on Communications, Internet and Information Technology*, Scottsdale, AZ, USA, Nov. 2003.

[15] A. Valembois, "Detection and recognition of a binary linear code." *Discrete Applied Mathematics*, vol. 111, pp. 199–218, 2001.

[16] ——, "Détection, reconnaissance et décodage de codes linéaires binaires." Ph.D. dissertation, Université de Limoges, France, Sept. 2000.

[17] M. Cluzeau, "Reconnaissance d'un schéma de codage." Ph.D. dissertation, École Polytechnique, Palaiseau, France, Nov. 2006.

[18] ——, "Reconnaissance d'un code linéaire en bloc en utilisant un algorithme de décodage itératif." in *Journées Codage et Cryptographie*, Eymoutiers, France, Oct. 2006.

[19] ——, "Block code reconstruction using iterative decoding techniques." in *Proc. 2006 IEEE International Symposium on Information Theory, ISIT06*, Seattle, USA, July 2006.

[20] A. Canteaut and F. Chabaud, "A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511." *IEEE Transactions on Information Theory*, vol. IT-44, no. 1, pp. 367–378, Jan. 1998.

[21] G. Sicot and S. Houcke, "Theorical study of the performance of a blind interleaver estimator." in *Proc. ISIVC 2006*, Hammamet, Tunisia, 2006.

[22] ——, "Blind detection of interleaver parameters." in *Proc. ICASSP 2005*, Philadelphia, USA, 2005.

[23] ——, "Etude statistique du seuil dans la détection d'entrelaceur." in *Proc. GRESTSI 2005*, Louvain la Neuve, Belgium, 2005.

[24] G. Sicot, S. Houcke, and J. Barbier, "Blind detection of interleaver parameters," *Journal of Signal Processing*, (To appear).

[25] I. Gohberg, T. Kailath, and V. Olshevsky, "Fast gaussian elimination with partial pivoting for matrices with displacement structure." *Mathematics of Computation*, vol. 64, pp. 1557–1576, 1995.