Channel Encoding Security Via Puncturing and Pruning

P.Jwalitha¹, CH. Ravi Kumar²

PG Scholar, Dept. of ECE, Sir C.R. Reddy College of Engineering and Technology, Eluru, AP, India¹

Sr. Assistant Professor, Dept. of ECE, Sir C.R. Reddy College of Engineering and Technology, Eluru, AP, India²

ABSTRACT: Channel encoding schemes are commonly used in modern digital communication systems to protect against the error free data transmission to the varying channel conditions. In this project we propose a new coding concepts puncturing and trellis pruning to provide security to the channel encoders based on turbo codes and to achieve reliability as well. Puncturing is employed to downgrade the performance of the code and thus increase the error probability experienced by an eavesdropper at a given SNR. The cryptanalytic attacks will become impossible because of puncturing due to whose complexity depends on the error probability. The trellis pruning, is enable the legitimate users reliably communicate in a secret fashion. The legitimate users achieve superior performance, in terms bit error rate and word than the eavesdropper, which depends on puncturing and pruning rates. We propose an algorithm to compute the puncturing and pruning rates based on exit charts.

KEYWORDS: Puncturing, trellis pruning, cryptanalytic attacks, exit charts, snr.

I. INTRODUCTION

Error control and security are prominent functions in digital communication systems. Reliability, security and efficient digital data transmission systems having more demand in high speed communication networks. In 1948 Shannon demonstrated that errors introduced by a noisy channel can be reduced to a desired level by proper encoding of information[12]. The use of coding for error control is an important part in modern communication systems design and digital computers. For providing security to the channel encoder that may lead to less complex systems[20] and improve processing latency. A secure channel encoder is a channel encoder that provides both reliability and security as well. Secure encoder schemes in the past few years have been presented in [17],[19],[23-25],[29]. Secure encoder design based on turbo codes by proposing the puncturing pattern and/or interleaver to be secret[17],[19],[23]. Reliability and security can be achieved by adapting the pseudo-random puncturing technique to the conditions of the noisy channel. Punctured convolutional codes are obtained by puncturing of some of the outputs of the convolutional encoder. The puncturing rule selects the outputs that are eliminated and not transmitted to the channel. Puncturing increases the rate of a convolutional code and is a useful design tool to achieve the convolutional codes with relatively high rates[2] and low performance. The security is provided to the channel encoder by secret trellis pruning were proposed[25]. In modern communication system due to trellis structure convolutional codes have many applications that allows efficient decoding[12]. Trellis pruning is introduced in [4] and is a technique of removing state transitions from trellis diagram due to determinate or indeterminate information bits. It is also used for constructing variable rate codes [10],[28] and also used for reliability improvement among co-operating users[22]. He does not know how pruning applied and not obtain message from his estimate of the transmitted sequence. Analysis of chosen and known plaintext attacks against this scheme in [27], where high BER is need when full mother trellis is decoding to achieve high security level. Secret pruning introduced the selection of secret sub code of the mother code which is used by legitimate users. In this project we see the extension of secure channel code design in [25] and also combines the puncturing and pruning techniques based on turbo codes. The secret trellis pruning gives low bit error rate for legitimate users. EXIT chart [7],[8] is a very useful tool for the analysis of iterative decoders. This tool analyse the iteration process in decoders that utilize soft-input-soft-output estimates that are passed from one decoder to the other. Exit charts of a turbo code with randomly punctured constituent convolution encoders directly from the exit charts of mother code is obtained by proposing a technique in [18],[26]. Randomly pruned constituent convolutional codes for
turbo codes used a method which is similar is presented in [21],[26],[28]. Exit analysis is used to determine the values of the puncturing and pruning rate that gives high bit error rate at the eaves dropper and low bit error rate at the legitimate users.

II. SYSTEM MODEL

In this project we consider the case where two legitimate users need to communicate in the presence of eaves droppers over a noisy channel. After that we assume the users are using turbo codes to correct errors induced by the channel [12] as shown in fig.1. The turbo encoder consists of convolutional encoders and the interleaver apart from that it also describes the two functions namely the pruning function $f$ and the puncturing function $p$. In [25] the pruning function is controlled by a key $K$ that allows us to consider the values of the pruning bits and the locations $L$ within the sequence entering into the encoder.

Assume that the convolutional encoder $(n,1)$ having memory $m$ is used (encoders 1 and 2 are identical) and the sequences $z$ having length $h$ as the input, then the output of encoder 1 is the parity sequence $w_1$ having length $(h+m)n$ which results from multiplexing the $n$ sequences $w_1,w_2,...,w_n$. The resulting of the rate of the constituent mother convolutional codes is

$$R = \frac{h}{(h+m)n} \quad (1)$$

which gives a rate $R_M =1/2n$ for the mother turbo code, if $h>>m$(normal case). However, in the system of fig.1., the sequence entering into the encoder is not the information sequence; $z$ is produced by intermixing the length $h-l$ information sequence $u$ and the length $l$ pruning sequence generated by the function $f$. The quantity

$$r_{pr} = \frac{l}{h} \quad (2)$$

is the pruning rate and equals the rate at which pruning bits are entering into encoder. The locations and their values are unknown to adversaries from the above. Even though, the same pruning function is applied to encoders each $f$ operates on a different state and the generated pruning sequences are not identical.

In general, the pruning function can be linear or non linear form. The function $p$ that perform the puncturing are operating on the output sequences $w_1,w_2,...,w_n$ and $w_1,w_2,...,w_n$ in a random non-secret fashion. Let $l$ be the number of encoded bits that are punctured from each $w^{(i)}$, then the quantity is

$$r_{pu} = \frac{l}{(h+m)n} \quad (3)$$

be the pruning rate. The resulting sequences $v^{(1)},v^{(2)}$ are multiplexed to produce the codeword $v$ which is transmitted through the communication channel.

The legitimate users know the values of the pruning bits and the pruning positions so the decoding of the received sequence can be performed on the reduced trellis diagram that results from the trellis of the constituent mother convolutional code by pruning state transitions that cannot occur [25]. On the other hand, the shared key is unknown to the eaves dropper, can decode the received sequence using full mother trellis diagram to get an estimate $z^*$ of $z$. However, the mother code is inferior than the code define using pruning i.e., after decoding he gets noisy estimate of $z$ depends on the channel quality. And also he does not know how to obtain message $u$ from $z$.Finally, we assume constituent encoder are recursive non systematic convolutional encoders. In contrast to common practice we do not use systematic encoders because this may lead to a scheme more sensitive to cryptanalytic attacks.

![fig 1: The architecture of secure turbo encoder](image-url)
In this section we present the main points of the EXIT analysis of turbo codes, as well as the EXIT charts of turbo codes with punctured and pruned constituent convolution codes [8],[18],[21],[26],[28] are reviewed. The convolutional MAP decoders constitute the iterative turbo decoder exchange extrinsic log likelihood ratios. Each decoder uses the extrinsic LLRs produced by the other decoder as a priori LLRs for the information bits. The quality of the extrinsic LLRs is measured by the mutual information $I_a$ between them and the information bits. Let $I_a$ be the mutual information between the priori LLRs and the information bits. For a SNR $Es/No=\gamma$ the transfer characteristic $I_e=\tau(I_a,\gamma)$ of the convolutional decoder can be derived via Monte Carlo simulations. A priori input $A$ to the constituent decoder can be modelled by applying an independent Gaussian random variable $n_A$ with variance $\sigma_A^2$ and mean zero in conjunction with the known transmitted systematic bits $x$.

$$A = \mu_A x + \eta_A$$  \hspace{0.5cm} (4)

since $A$ is supposed to be a L-value based on Gaussian distributions, the mean value $\mu_A$

$$\mu_A = \frac{\sigma_A^2}{2}$$  \hspace{0.5cm} (5)

The conditional probability density function belonging to the L-value $A$ is

$$p_A(\xi/X = x) = \frac{e^{-\frac{(\xi - \mu_A^2/2)^2}{2\sigma_A^2}}}{\sqrt{2\pi\sigma_A}}$$  \hspace{0.5cm} (6)

To measure the information contents of the a priori knowledge, mutual information $I_a = I(x; A)$ between transmitted systematic bits $x$ and L-values $A$ is used

$$I_a = \frac{1}{2} \sum_{x=-1}^{1} \int_{-\infty}^{\infty} p_A(\xi/X = x) \cdot ld \frac{2p_A(\xi/X = 1)}{p_A(\xi/X = -1) + p_A(\xi/X = 1)} d\xi$$  \hspace{0.5cm} (7)

$0 \leq IA \leq 1$

with (6) and (7) becomes

$$I_a(\sigma_A) = 1 - \int_{-\infty}^{\infty} e^{-\frac{(\xi - \mu_A^2/2)^2}{2\sigma_A^2}} \cdot ld[1 + e^{-\xi}] d\xi$$  \hspace{0.5cm} (8)

For abbreviation we define

$$J(\sigma) := I_a(\sigma_A = \sigma)$$  \hspace{0.5cm} (9)

with $\lim_{\sigma \to 0} J(\sigma) = 0$, $\lim_{\sigma \to \infty} J(\sigma) = 1$, $\sigma > 0$.

In [14] The capacity $C_G$ (the function of $J(\sigma)$ respectively cannot be expressed in closed form. It is monotonically increasing function in $\sigma = 2/\sigma_0$ and thus reversible

$$\sigma_A = J^{-1}(I_a)$$  \hspace{0.5cm} (10)

In the case of AWGN channel, the a priori LLRs can be modelled as independent Gaussian random variable with variance $\sigma_A^2$ and mean $\mu_A = 1 \cdot \sigma_A^2/2$ with (9) and (10) becomes

$$I_a(\sigma_A = \sigma) = J(\sigma) = 1 - \int_{-\infty}^{\infty} e^{-\frac{(\xi - \mu_A^2/2)^2}{2\sigma^2}} \cdot ld[1 + e^{-\xi}] d\xi$$

Different values of $I_a$ we produce the priori LLRs and run the decoding algorithm. If $I_e$ as a function of $I_a$ and the $Es/No$ value, the extrinsic information transfer characteristics are defined in [14] as

$$I_e = \tau(I_a)$$  \hspace{0.5cm} (12)

or for fixed $\frac{Es}{No}$, just

$$I_e = \tau(1)$$  \hspace{0.5cm} (13)
The EXIT chart of the turbo code is formed by plotting $I_e = \tau(I_A,\gamma)$ and its symmetric to the line $I_e = I_A$ (the constituent decoders are identical which is assumed). If the two curves intersect only at the point $I_e = I_A = 1$, then the iterative decoder converges to low probability of error. Otherwise, if $I_e \neq 1$, no use how many iterations are performed. The iterative decoding cannot lead to low error probability.

The EXIT chart can be used to obtain an estimate on the BER after an arbitrary number of iterations. For both constituent decoders, the soft output on the systematic bits can be written as

$$D = Z + A + E$$

(14)

For deriving the formula (14) on bit error probability $P_b$, we assume $A$ is a priori knowledge and $E$ be the extrinsic output to be Gaussian distributed and $D$ is decoder soft output. $D$ be Gaussian distributed with variance $\sigma_D^2$ and mean value $\mu_D$.

with the complementary error function, the bit error probability, the bit error probability written as

$$P_b = \frac{1}{2} \text{erfc}(\frac{\mu_D}{\sqrt{2}\sigma_D}) = \frac{1}{2} \text{erfc}(\frac{\sigma_D}{\sqrt{2}})$$

(15)

Assume independence it is

$$\sigma_D^2 = \sigma_A^2 + \sigma_E^2$$

(16)

we know that

$$\frac{\bar{E}}{\bar{N}} = \frac{1}{2\sigma_D^2}$$

By using the EXIT chart, the estimation of bit error rate after no of iterations for a particular point $(I_A, I_E)$ may lead the decoding procedure to that point be

$$P_b \approx \frac{1}{2} \text{erfc} \left( \frac{\sqrt{\bar{Y} + f^{-1}(I_A) + f^{+1}(I_E)^2}}{2\sqrt{2}} \right)$$

(17)

The approximations (17) is accurate only for high BERs. The transfer characteristic $\tau^{(pa)}$ of a randomly punctured convolutional code, based on the transfer characteristic $\tau$ of the (un punctured) mother code is determine in [15],[16]. We assume that transmission takes place over an AWGN channel, then it holds that

$$\tau^{(pa)}(I_A, \gamma) = \tau(I_A, \gamma')$$

(18)

Equation (18) shows the behaviour of the decoder at SNR $\gamma$ after random puncturing is identical to the behaviour of the decoder without puncturing at a lower SNR value $\gamma'$. [16] where

$$J(2\sqrt{2\gamma}) = (1 - r_{pu})/(2\sqrt{2\gamma})$$

(19)

The transfer characteristic $\tau^{(pr)}$ of a randomly punctured convolutional code results from the transfer characteristic $\tau$ of the mother code as follows [from [10]]

$$\tau^{(pr)}(I_A, \gamma) = \tau(I_A', \gamma)$$

(20)

$$I_A' = I_A + r_{pr}(1-I_A) \geq I_A$$

(21)

That is some of input bits are known (the pruned bits) increases the mutual information between the a priori LLRs and the input bits compared to the case of the mother(un punctured) code. Here we consider the AWGN channel in that (18) and (20) relations are approximations. Because, the transfer characteristic $\tau$ of the mother code has been produced by the assumption of the priori LLRs and the channel LLRs are Gaussian random variables. Coming to puncturing and pruning case priori LLRs correspond to pruning bits and the channel LLRs correspond to puncturing bits respectively are not Gaussian variables. However, approximations are very close to actual transfer functions and if $r_{pu}$ and $r_{pr}$ are small or moderate, they are almost identical. If the channel is a binary erasure channel relations (18) and (20) provide the exact transfer characteristics.

IV. PROPOSED METHOD

In this paper we using the BCJR decoding algorithm to compute the puncturing rate that gives the high BER at the eaves dropper (by decoding the full mother trellis) and the pruning rate gives low BER at the legitimate user. The proposed algorithm is based on EXIT analysis and gives the results in [10],[14],[16],[17].

Copyright to IJIRCCE

DOI: 10.15680/IJIRCCE.2016.0404055
In this paper we assume the transmission is performed through an AWGN channel (additive white Gaussian noise channel) and the legitimate source knows the SNR at the legitimate destination and eavesdropper. Here, we assume that the constituent convolutional decoder transfer characteristic is known for each SNR value.

Let $P_{\text{eve}}$ be the bit error rate at the eavesdropper which is after decoding with the full mother trellis and $\delta$ be its required minimum value. Let $\gamma_i$ be the first SNR value of the mother turbo code such that the BER of mother turbo code is greater than or equal to $\delta$. This can be achieved by using eq (17) because $\delta$ has to be high. The eq(17) is derived for systematic constituent encoders, where the a posteriori LLR of the information bit is the sum of its apriori, extrinsic and channel LLRs. For non-systematic encoders, the a posteriori LLR of an information bit is the sum of its apriori and extrinsic LLRs. From eq(17) becomes

$$P_b \approx \frac{1}{2} \text{erfc}\left(\frac{\sqrt{2\gamma_i^2+\delta^2}}{2\gamma_i}\right)$$  \hspace{1cm} (22)

From eq(22) we get the transfer characteristics of the two constituent decoders must intersect at a point $(I_a,I_a)$ such that

$$\delta \leq \frac{1}{2} \text{erfc}\left(\frac{\sqrt{2\gamma^2}}{2\gamma}\right)$$  \hspace{1cm} (23)

$$I_a \leq f(2\text{erfc}^{-1}(2\delta)) = I_{a0}$$  \hspace{1cm} (24)

If the SNR $\gamma_0$ is moderate and $I_{a0} < I_{E0} = \tau(I_{A0};\gamma_0)$ we gradually decrease it with step $\epsilon$ for each value of $\gamma$ and calculate $I_{E0} = \tau(I_{A0};\gamma)$ using Monte Carlo simulations. The first SNR value $\gamma_i$ set by $I_{A0} \geq I_{E0}$. The SNR transfer functions intersect at $(I_{A0},I_{A0})$ and this gives the BER of the mother code is greater than or equal to $\delta$. The algorithm is summarized as

**Algorithm 1**: computation of $\gamma_i$

**Input**: $I_{A0}, \gamma_0$,

\[ \gamma = \gamma_0 \]
\[ \text{while}(I_{A0} < I_{E0}) \text{ do} \]
\[ \gamma = \gamma - \epsilon \]
\[ I_{E0} = \tau(I_{A0};\gamma) \]
\[ \text{end} \]
\[ \gamma_i = \gamma \]

**Output**: $\gamma_i$

The two basic steps of the main algorithm that computes the puncturing and pruning rate is present below.

a) The Puncturing step:

Let $\gamma_i > \gamma_i$ be the SNR at the eavesdropper. $P_{\text{eve}}$ be the bit error rate at the eavesdropper, after decoding the full mother trellis is greater or equal to $\delta$. Using (19) we calculate the value of

$$r_{pu} = 1 - \frac{f(2\sqrt{r_{pu}})}{f(2\sqrt{2}\gamma_i)}$$ \hspace{1cm} (25)

for which it holds that

$$\tau_{pu}(I_a, \gamma_0) = \tau(I, \gamma_0)$$ \hspace{1cm} (26)

for $0 \leq r_{pu} \leq 1$.

The value of $r_{pu}$ that gives the transfer function of the randomly punctured convolutional code at SNR $\gamma_e$ is identical to the transfer function of the (unpunctured) mother code at SNR $\gamma_i$. Then eq(22) gives that $P_{\text{eve}} \geq \delta$.

b) The Pruning step:

In the pruning step, it was employed it was employed to increase the bit error rate at the eavesdropper. The pruning technique is applied in a secret fashion to improve reliability. If $\gamma_d$ is the SNR at the destination from (18)

$$\tau_{pu}(I_a, \gamma_d) = \tau(I, \gamma_d)$$ \hspace{1cm} (27)

where from(19)

$$f(2\sqrt{r_{pu}}) = (1 - r_{pu})f(2\sqrt{2}\gamma_d)$$
we have to find the minimum value of $r_{pr}^{\min}$ of $r_{pr}$ gives the iterative decoding converges to low bit error rate i.e., the two corresponding transfer functions intersect only at the point (1,1). This implies the following condition must hold:

$$\tau(\hat{I}_A, \gamma_d) - I_A \leq 0 \quad \text{for} \quad 0 \leq I_A \leq 1$$

The decoder will converge after a moderate no. of iterations, it is better to improve the constraint

$$\tau(\hat{I}_A, \gamma_d) - I_A > \Omega \quad \text{for} \quad 0 \leq I_A \leq 1$$

where \(\Omega = \left\{ \emptyset \mid 0 \leq I_A \leq 1 - \emptyset \right\} \) \(0 < \emptyset < 1\)

A large value of \(\emptyset\) gives fast convergence, after a small no of iterations. If \(\Omega = 0\) for \(I_A \geq 1 - \emptyset\) otherwise eq(28) yields

$$\tau(\hat{I}_A, \gamma_d) \geq 1$$

we have two possible approaches to determine the value of $r_{pr}^{\min}$. The first one is the algorithm[10]. $r_{pr}$ is continuously increased and the corresponding transfer characteristics is computed using (20) and (21) until (28) is satisfied. An alternative procedure is the following from (20) and (28), $r_{pr}^{\min}$ must be such that

$$\tau(\hat{I}_A, \gamma_d) - I_A \geq \Omega \quad \text{for} \quad 0 \leq I_A \leq 1$$

or equivalently

$$I_A^* \geq \tau^{-1}(\Omega + I_A, \gamma_d), 0 \leq I_A \leq 1$$

Let $r_{pr}^*(I_A)$ be the of $r_{pr}$ for which (30) holds with equality for a particular value of $I_A$. which means $r_{pr}^*(I_A) = r_{pr}$. From (21) and (30) straight forward computations yield

$$r_{pr} = \frac{I_A^* - I_A}{1 - I_A}$$

(31)

by setting $I_E = \Omega + I_A$ in (31). Then, the minimum pruning rate guaranteeing that every point of $\tau(\hat{I}_A, \gamma_d)$ satisfies (28) is equal to

$$r_{pr}^{\min} = \max_{I_E \leq 1} \left\{ r_{pr}^*(I_E) \right\}$$

(33)

The exact form of the functions $\tau^{-1}(\hat{I}_E, \gamma_d)$ and $\tau(I_A, \gamma_d)$ are not known. The only information we have for these functions is a set of N pairs $(I_{A1}, I_{E1})$ which have resulted from Monte Carlo simulations. $r_{pr}^{\min}$ will be the result from these pairs. Relation (32) becomes

$$r_{pr}^*(i) = \frac{I_{A1} + \Omega - I_{E1}}{1 + \Omega - I_{E1}} \quad 1 \leq i \leq N - 1$$

(34)

and

$$r_{pr}^{\min} = \max \left\{ r_{pr}^*(i) \right\}$$

(35)

In this we assume the Nth pair is (1,1) but is not taken into account in (34). Because $I_A$ has taken its maximum value but it is not exist an $r_{pr}$, which can it further.

The total code rate equals

$$R = \frac{h}{2n(h + m)} \frac{1 - r_{pr}^{\min}}{1 - r_{pu}}$$

If $h >> m$ then

$$R = \frac{1}{2n} \frac{1 - r_{pr}^{\min}}{1 - r_{pu}}$$

(36)

ALGORITHM 2: To compute $r_{pr}^{\min}$ and $r_{pu}$

Input: $\gamma_E, \gamma_d, \gamma_i, \tau, \emptyset$
Initialization : \( r_{pr}^{min} = 0 \)

\[
\begin{align*}
    r_{pu} &= 1 - \frac{f(2\sqrt{2\gamma})}{f(2\sqrt{2\gamma_e})} \\
    \gamma'_d &= \frac{1}{8} \left[ f^{-1}((1 - r_{pu})f(2\sqrt{2\gamma_d})) \right]^2
\end{align*}
\]

for i=1:N-1 do
    \( r_{pr}^*(i) = \frac{I_{det} - I_{det}}{1 + I_{det} - I_{det}} \)
    if \( r_{pr}^{min} < r_{pr}^* \) then
        \( r_{pr}^{min} = r_{pr}^* \)
    end if
end for

Output : \( r_{pr}^{min} , r_{pu} \).

V. SIMULATION RESULTS

The simulation studies involves the extrinsic information transfer characteristics of mother turbo code and the exit chart of the randomly punctured mother turbo code which shows in fig 2. And fig 3 depicts the exit chart of the mother turbo code, which approximates the randomly punctured and pruned mother turbo code. The exit chart analysis is implemented with the MATLAB. The bit error rate performance over additive white Gaussian noise channel is shows in fig 4. The priori information which is given as input and the bit error rate performance was depicted in fig 5.

![fig 2: Exit chart of the mother turbo code and exit chart of the randomly punctured mother turbo code.](image1)

![fig 3: Exit chart of the mother turbo code and exit chart of the randomly punctured and pruned mother turbo code.](image2)
The security in channel encoder can be improved by puncturing and trellis pruning. The constituent encoders are randomly punctured to produce codes of higher rate and poor performance for specific channel conditions. In addition with secret trellis pruning which increases performance allows legitimate users to experience a low bit error rate. In that the key defines how pruning is applied on the trellis of a mother convolutional code, this results into a secret pruned trellis that legitimate users are using to perform decoding , in contrast to the eavesdropper that employed the full mother trellis diagram. The minimum pruning rate guaranteeing the desired reliability levels. However, security requirements may impose various constraints on $r_{pr}$, as pruning is performed in a secret fashion. The complexity of cryptanalytic attack depend on $r_{pr}$, which needs to be large enough to allow for adequate security. The EXIT chart has been presented as an engineering tool for the design of iterative decoding schemes. we have presented the extrinsic information transfer characteristics based on mutual information to describe the flow of extrinsic information through the soft in/soft out constituent decoders.

REFERENCES