

Groupe de travail Réseau  
**Request for Comments : 5053**  
 Catégorie : Sur la voie de la normalisation  
 Traduction Claude Brière de L'Isle

M. Luby, Digital Fountain  
 A. Shokrollahi, EPFL  
 M. Watson, Digital Fountain  
 T. Stockhammer, Nomor Research  
 octobre 2007

## Schéma de correction d'erreur directe Raptor pour la livraison d'objet

### Statut du présent mémoire

Le présent document spécifie un protocole de l'Internet sur la voie de la normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

### Résumé

Le présent document décrit un schéma de correction d'erreur directe (FEC, *Forward Error Correction*) pleinement spécifiée, correspondant à l'identifiant de codage de FEC 1, pour le code de correction d'erreur de transmission Raptor et son application à la livraison fiable des objets de données.

Raptor est un code fontaine, c'est-à-dire, autant de symboles de codage que nécessaire peuvent être générés en vol par le codeur à partir des symboles de source d'un bloc de source de données. Le décodeur est capable de récupérer le bloc de source à partir de tout ensemble de symboles de codage seulement légèrement plus grand que le nombre de symboles de source.

Le code Raptor décrit ici est un code systématique, ce qui signifie que tous les symboles de source sont parmi les symboles de codage qui peuvent être générés.

### Table des Matières

1. Introduction.....	2
2. Exigences de notation.....	2
3. Formats et codes.....	2
3.1 Identifiants de charge utile de FEC.....	2
3.2 Informations de transmission d'objet de FEC.....	2
4. Procédures.....	3
4.1 Exigences du protocole de livraison de contenu.....	3
4.2 Exemple d'algorithme de déduction de paramètres.....	4
5. Spécification du code de FEC Raptor.....	5
5.1 Définitions, symboles, et abréviations.....	5
5.2 Vue d'ensemble.....	7
5.4 Codeur Raptor systématique.....	9
5.5 Exemple de décodeur de FEC.....	13
5.6 Nombres aléatoires.....	16
5.7 Indices systématiques J(K).....	17
6. Considérations sur la sécurité.....	23
7. Considérations relatives à l'IANA.....	23
8. Remerciements.....	23
9. Références.....	23
9.1 Références normatives.....	23
9.2 Références pour information.....	23
Adresse des auteurs.....	24
Déclaration complète de droits de reproduction.....	24

### 1. Introduction

Le présent document spécifie un schéma de FEC pour le code de correction d'erreur directe Raptor pour les applications de livraison d'objet. Le concept de schéma de FEC est défini dans la [RFC5052] et le présent document suit le format qui y est prescrit et utilise la terminologie de ce document. Les codes Raptor ont été introduits dans [Raptor]. Pour une vue d'ensemble, voir par exemple, [CCNC].

Le schéma de FEC Raptor est un schéma de FEC pleinement spécifié qui correspond à l'identifiant de codage de FEC 1.

Raptor est un code fontaine, c'est-à-dire, autant de symboles de codage que nécessaire peuvent être générés en vol par le codeur à partir des symboles de source d'un bloc. Le décodeur est capable de récupérer le bloc de source à partir de tout ensemble de symboles de codage seulement légèrement plus nombreux que le nombre de symboles de source.

Le code décrit dans le présent document est un code systématique, c'est-à-dire, les symboles de source originaux peuvent être envoyés non modifiés de l'envoyeur au receveur, ainsi qu'un certain nombre de symboles de réparation. Pour les détails de l'utilisation des codes de correction d'erreur directe en diffusion groupée fiable, voir la [RFC3453].

Le code décrit ici est identique à celui décrit dans [MBMS].

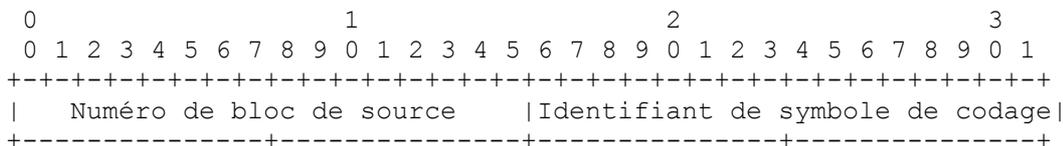
### 2. Exigences de notation

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

### 3. Formats et codes

#### 3.1 Identifiants de charge utile de FEC

L'identifiant de charge utile de FEC DOIT être un champ de 4 octets défini comme suit :



**Figure 1 : Format d'identifiant de charge utile de FEC**

Numéro de bloc de source (SBN, *Source Block Number*) (16 bits) : entier identifiant le bloc de source auquel les symboles de codage au sein du paquet se rapportent.

Identifiant de symbole de codage (ESI, *Encoding Symbol ID*), (16 bits) : entier identifiant les symboles de codage au sein du paquet.

L'interprétation du numéro de bloc de source et de l'identifiant de symbole de codage est définie à la Section 5.

#### 3.2 Informations de transmission d'objet de FEC

##### 3.2.1 Obligatoire

La valeur de l'identifiant de codage de FEC DOIT être 1 (un) comme allouée par l'IANA (voir la Section 7).

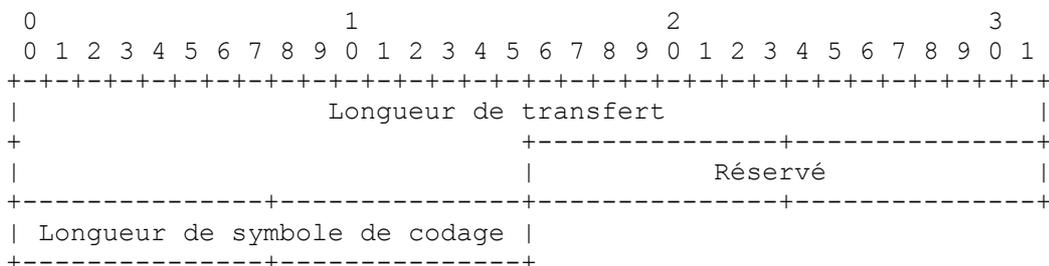
##### 3.2.2 Commun

Les éléments communs d'information de transmission d'objet de FEC utilisés par ce schéma de FEC sont :

- Longueur de transfert (F)
- Longueur de symbole de codage (T)

La longueur de transfert est un entier non négatif inférieur à  $2^{45}$ . La longueur de symbole de codage est un entier non négatif inférieur à  $2^{16}$ .

Le format des informations codées communes de transmission d'objet de FEC est montré à la Figure 2.



**Figure 2 : Informations de transmission de FEC codée communes pour le schéma de FEC Raptor**

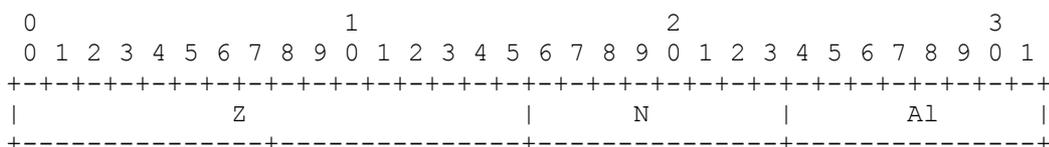
Note 1 : la limite de  $2^{45}$  de la longueur de transfert est la conséquence de la limitation de la taille de symbole à  $2^{16}-1$ , de la limitation du nombre de symboles dans un bloc de source à  $2^{13}$ , et de la limitation du nombre de blocs de source à  $2^{16}$ . Cependant, la longueur de transfert est codée comme un champ de 48 bits pour simplifier.

### 3.2.3 Spécifique du schéma

Les paramètres suivants sont portés dans l'élément Informations de transmission d'objet de FEC spécifiques du schéma pour ce schéma de FEC :

- le nombre de blocs de source (Z)
- le nombre de sous blocs (N)
- un paramètre d'alignement de symbole (Al)

Ces paramètres sont tous des entiers non négatifs. Les Informations de transmission d'objet spécifiques du schéma sont un champ de 4 octets consistant en les paramètres Z (2 octets), N (1 octet), et Al (1 octet) comme montré à la Figure 3.



**Figure 3 : Informations de transmission d'objet de FEC spécifiques du schéma codées**

Les informations de transmission d'objet de FEC codées sont un champ de 14 octets consistant en l'enchaînement des informations de transmission d'objet de FEC codées communes et des informations de transmission d'objet de FEC spécifiques du schéma codées.

Ces trois paramètres définissent la partition du bloc de source décrite au paragraphe 5.3.1.2.

## 4. Procédures

### 4.1 Exigences du protocole de livraison de contenu

Cette Section décrit l'échange d'informations entre le schéma de FEC Raptor et tout protocole de livraison de contenu (CDP, *Content Delivery Protocol*) qui utilise le schéma de FEC Raptor pour la livraison d'objet.

Le codeur et le décodeur Raptor pour la livraison d'objet exigent les informations suivantes de la part du CDP :

- la longueur de transfert de l'objet, F, en octets
- un paramètre d'alignement de symbole, Al

- la taille de symbole, T, en octets, qui DOIT être un multiple de Al
- le nombre de blocs de source, Z
- le nombre de sous-blocs dans chaque bloc de source, N

Le codeur Raptor pour la livraison d'objet exige de plus :

- l'objet à coder, F octets.

Le codeur Raptor fournit au CDP les informations suivantes pour chaque paquet à envoyer :

- le numéro de bloc de source (SBN)
- l'identifiant de symbole de codage (ESI)
- les symboles de codage.

Le CDP DOIT communiquer ces informations au receveur.

#### 4.2 Exemple d'algorithme de déduction de paramètres

Ce paragraphe donne des recommandations pour la déduction des trois paramètres de transport, T, Z, et N. Cette recommandation se fonde sur les paramètres d'entrée suivants :

F : longueur de transfert de l'objet, en octets,

W : cible sur la taille de sous bloc, en octets;

P : taille de charge utile maximum de paquet, en octets, qui est supposée être un multiple de Al,

Al : paramètre d'alignement de symbole, en octets,

Kmax : nombre maximum de symboles de source par bloc de source. Note : le paragraphe 5.1.2 définit Kmax comme 8192,

Kmin : cible minimum sur le nombre de symboles par bloc de source,

Gmax : nombre cible maximum de symboles par paquet.

Sur la base de ces entrées, les paramètres de transport T, Z, et N sont calculés comme suit :

Soit

$$G = \min \{ \text{ceil}(P * K_{\min} / F), P / Al, G_{\max} \}$$

$$T = \text{floor}(P / (Al * G)) * Al$$

$$K_t = \text{ceil}(F / T)$$

$$Z = \text{ceil}(K_t / K_{\max})$$

$$N = \min \{ \text{ceil}(\text{ceil}(K_t / Z) * T / W), T / Al \}$$

La valeur G représente le nombre maximum de symboles à transporter dans un seul paquet. La valeur K<sub>t</sub> est le nombre total de symboles requis pour représenter les données de source de l'objet. Les valeurs de G et N déduites ci-dessus devraient être considérées comme des limites inférieures. Il peut être avantageux d'augmenter ces valeurs, par exemple, à la plus proche puissance de deux. En particulier, l'algorithme ci-dessus ne garantit pas que la taille de symbole, T, divise la taille maximum de paquet, P, et donc il peut n'être pas possible d'utiliser des paquets de la taille exacte de P. Si, à la place, G est choisi comme étant une valeur qui divise P/Al, alors la taille de symbole, T, va être un diviseur de P et les paquets de taille P peuvent être utilisés.

L'algorithme ci-dessus et qui est défini au paragraphe 5.3.1.2 assure que les tailles de sous-symboles sont un multiple du paramètre d'alignement de symbole, Al. Ceci est utile parce que les opérations de OU exclusif (OUX) utilisées pour le codage et décodage sont généralement effectuées sur plusieurs octets à la fois, par exemple, au moins 4 octets à la fois sur un processeur de 32 bits. Donc, le codage et décodage peut être effectué plus vite si les tailles de sous-symboles sont un multiple de ce nombre d'octets.

Les réglages recommandés pour les paramètres d'entrée, Al, K<sub>min</sub>, et G<sub>max</sub> sont comme suit : Al = 4, K<sub>min</sub> = 1024, G<sub>max</sub> = 10.

Le paramètre W peut être utilisé pour générer des données codées qui peuvent être décodées efficacement avec une mémoire de travail limitée au décodeur. Noter que l'exigence réelle de mémoire maximum de décodeur pour une valeur donnée de W dépend de la mise en œuvre, mais il est possible d'effectuer le décodage en utilisant une mémoire de travail seulement légèrement plus grande que W.

## 5. Spécification du code de FEC Raptor

### 5.1 Définitions, symboles, et abréviations

#### 5.1.1 Définitions

Pour les besoins de la présente spécification, on applique les termes et définitions suivants :

Bloc de source : bloc de  $K$  symboles de source qui sont pris en compte ensemble pour les besoins du codage Raptor.

Code systématique : code dans lequel tous les symboles de source peuvent être inclus au titre des symboles de codage envoyés pour un bloc de source.

Groupe de symboles de codage : groupe de symboles de codage qui sont envoyés ensemble, c'est-à-dire, au sein du même paquet, dont la relation aux symboles de source peut être déduite d'un seul identifiant de symbole de codage.

Identifiant de symbole de codage : informations qui définissent les relations entre les symboles d'un groupe de symbole de codage et les symboles de source.

Paquet de codage : paquet de données qui contient des symboles de codage

Paquet de réparation : paquet de données qui contient des symboles de réparation.

Paquet de source : paquet de données qui contient des symboles de source.

Sous bloc : un bloc de source est parfois coupé en sous blocs, dont chacun est suffisamment petit pour être décodé dans la mémoire de travail. Pour un bloc de source consistant en  $K$  symboles de source, chaque sous bloc consiste en  $K$  sous symboles, chaque symbole du bloc de source étant composé d'un sous symbole provenant de chaque sous bloc.

Sous symbole : partie d'un symbole. Chaque symbole de source est composé d'autant de sous symboles qu'il y a de sous blocs dans le bloc de source.

Symbole : unité de données. La taille, en octets, d'un symbole est appelée la taille de symbole.

Symbole de codage : symbole qui est inclus dans un paquet de données. Les symboles de codage consistent en les symboles de source et les symboles de réparation. Les symboles de réparation générés à partir d'un bloc de source ont la même taille que les symboles de source de ce bloc de source.

Symbole de source : plus petite unité de données utilisée durant le processus de codage. Tous les symboles de source au sein d'un bloc de source ont la même taille.

Symboles intermédiaires : symboles générés à partir des symboles de source en utilisant un processus de codage inverse. Les symboles de réparation sont alors générés directement à partir des symboles intermédiaires. Les symboles de codage n'incluent pas les symboles intermédiaires, c'est-à-dire, les symboles intermédiaires ne sont pas inclus dans les paquets de données.

Symbole de réparation : symboles de codage envoyés pour un bloc de source qui ne sont pas les symboles de source. Les symboles de réparation sont générés sur la base des symboles de source.

#### 5.1.2 Symboles

$i, j, x, h, a, b, d, v, m$  représentent des entiers positifs.

$\text{ceil}(x)$  : note le plus petit entier positif supérieur ou égal à  $x$ .

$\text{choose}(i,j)$  : note le nombre de façons dont  $j$  objets peuvent être choisis parmi  $i$  objets sans répétition.

$\text{floor}(x)$  : note le plus grand entier positif inférieur ou égal à  $x$ .

$i \% j$  : note  $i$  modulo  $j$ .

$X \wedge Y$  : note, pour des chaînes binaires  $X$  et  $Y$  de longueur égale, l'opération OU exclusif au bit près de  $X$  et  $Y$ .

$Al$  : note un paramètre d'alignement de symbole. Les tailles de symbole et sous symbole ne peuvent être que des multiples de  $Al$ .

$A$  : note une matrice sur  $GF(2)$ .

$Transpose[A]$  : note la matrice transposée de la matrice  $A$ .

$A^{-1}$  : note la matrice inverse de la matrice  $A$ .

$K$  : note le nombre de symboles dans un seul bloc de source.

$K_{max}$  : note le nombre maximum de symboles de source qui peuvent être dans un seul bloc de source. Régulé à 8192.

$L$  : note le nombre de symboles de précodage pour un seul bloc de source.

$S$  : note le nombre de symboles LDPC pour un seul bloc de source.

$H$  : note le nombre de demi symboles pour un seul bloc de source.

$C$  : note un dispositif de symboles intermédiaires,  $C[0], C[1], C[2], \dots, C[L-1]$ .

$C'$  : note un dispositif de symboles de source,  $C'[0], C'[1], C'[2], \dots, C'[K-1]$ .

$X$  : valeur d'entier non négatif.

$V_0, V_1$  : deux dispositifs d'entiers de quatre octets,  $V_0[0], V_0[1], \dots, V_0[255]$  et  $V_1[0], V_1[1], \dots, V_1[255]$

$Rand[X, i, m]$  : générateur de nombres pseudo aléatoires.

$Deg[v]$  : générateur de degré.

$LTEnc[K, C, (d, a, b)]$  : générateur de symboles de codage LT.

$Trip[K, X]$  : fonction de triple générateur.

$G$  : nombre de symboles dans un groupe de symboles de codage.

$GF(n)$  : champ de Galois avec  $n$  éléments.

$N$  : nombre de sous blocs dans un bloc de source.

$T$  : taille de symbole en octets. Si le bloc de source est partagé en sous blocs, alors  $T = T' * N$ .

$T'$  : taille de sous symbole, en octets. Si le bloc de source n'est pas partagé en sous blocs, alors  $T'$  n'est pas pertinent.

$F$  : longueur de transfert d'un objet, en octets.

$I$  : taille de sous bloc en octets.

$P$  : pour la livraison d'objet, la taille de charge utile de chaque paquet, en octets, utilisée dans la déduction recommandée des paramètres de transport de la livraison de l'objet.

$Q$  :  $Q = 65521$ , c'est-à-dire,  $Q$  est le plus grand premier inférieur à  $2^{16}$ .

$Z$  : nombre de blocs de source, pour la livraison d'objet.

$J(K)$  : indice systématique associé à  $K$ .

$I_S$  : note la matrice d'identité  $S \times S$ .

$0_{S \times H}$  : note la matrice  $S \times H$  zéro.

$a^b$  : a à la puissance b.

### 5.1.3 Abréviations

Pour les besoins du présent document, les abréviations suivantes s'appliquent :

ESI (*Encoding Symbol ID*) : identifiant de symbole de codage.

LDPC (*Low Density Parity Check*) : vérification de parité à basse densité.

LT (*Luby Transform*) : transformation de Luby.

SBN (*Source Block Number*) : numéro de bloc de source

SBL (*Source Block Length*) : longueur de bloc de source (en unités de symboles)

## 5.2 Vue d'ensemble

Le principal composant du code systématique Raptor est le codeur de base décrit au paragraphe 5.4. On décrit d'abord comment déduire les valeurs pour un ensemble de symboles intermédiaires à partir des symboles de source originaux tels que la connaissance des symboles intermédiaires soit suffisante pour reconstruire les symboles de source. Ensuite, le codeur produit des symboles de réparation, qui sont chacun le OUX d'un certain nombre de symboles intermédiaires. Les symboles de codage sont la combinaison des symboles de source et de réparation. Les symboles de réparation sont produits d'une façon telle que les symboles intermédiaires, et donc aussi les symboles de source, puissent être récupérés à partir de tout ensemble suffisamment grand de symboles de codage.

Le présent document spécifie le codeur de code Raptor systématique. Un certain nombre d'algorithmes de décodage sont possibles. Un algorithme de décodage efficace est fourni au paragraphe 5.5.

La construction des symboles intermédiaires et de réparation se fonde en partie sur un générateur de nombres pseudo aléatoires décrit au paragraphe 5.4.4.1. Ce générateur se fonde sur un ensemble fixe de 512 nombres aléatoires qui DOIVENT être disponibles à l'envoyeur et au receveur. Ils sont donnés au paragraphe 5.6.

Finalement, la construction des symboles intermédiaires à partir des symboles de source est gouvernée par un "indice systématique", dont les valeurs sont données au paragraphe 5.7 pour les tailles de bloc de source de quatre symboles de source à  $K_{max} = 8192$  symboles de source.

## 5.3 Livraison d'objet

### 5.3.1 Construction de bloc de source

#### 5.3.1.1 Généralités

Pour appliquer le codeur Raptor à un objet de source, l'objet peut être coupé en  $Z \geq 1$  blocs, appelés des blocs de source. Le codeur Raptor est appliqué indépendamment à chaque bloc de source. Chaque bloc de source est identifié par un unique SBN entier, où le premier bloc de source a le SBN zéro, le second a le SBN un, etc. Chaque bloc de source est divisé en un nombre,  $K$ , de symboles de source d'une taille de  $T$  octets chaque. Chaque symbole de source est identifié par un unique ESI entier, où le premier symbole de source d'un bloc de source a l'ESI zéro, le second a l'ESI un, etc.

Chaque bloc de source avec  $K$  symboles de source est divisé en  $N \geq 1$  sous blocs, qui sont assez petits pour être décodés dans la mémoire de travail. Chaque sous bloc est divisé en  $K$  sous symboles de taille  $T$ .

Noter que la valeur de  $K$  n'est pas nécessairement la même pour chaque bloc de source d'un objet et que la valeur de  $T$  peut ne pas nécessairement être la même pour chaque sous bloc d'un bloc de source. Cependant, la taille de symbole  $T$  est la même pour tous les blocs de source d'un objet et le nombre de symboles,  $K$ , est le même pour chaque sous bloc d'un bloc de source. Le partage exact de l'objet en blocs de source et sous blocs est décrit au paragraphe 5.3.1.2.

### 5.3.1.2 Bloc de source et partition en sous blocs

La construction des blocs de source et sous blocs est déterminée sur la base de cinq paramètres d'entrée, F, Al, T, Z, et N, et d'une fonction Partition[]. Les cinq paramètres d'entrée sont définis comme suit :

- F : longueur de transfert de l'objet, en octets.
- Al : paramètre d'alignement de symbole, en octets.
- T : taille de symbole, en octets, qui DOIT être un multiple de Al.
- Z : nombre de blocs de source.
- N : nombre de sous blocs dans chaque bloc de source.

Ces paramètres DOIVENT être réglés de telle façon que  $\text{ceil}(\text{ceil}(F/T)/Z) \leq K_{\text{max}}$ . Des recommandations pour la déduction de ces paramètres sont fournies au paragraphe 4.2.

La fonction Partition[] prend une paire d'entiers (I, J) en entrée et déduit quatre entiers (IL, IS, JL, JS) en sortie. Précisément, la valeur de Partition[I, J] est une séquence de quatre entiers (IL, IS, JL, JS), où  $IL = \text{ceil}(I/J)$ ,  $IS = \text{floor}(I/J)$ ,  $JL = I - IS * J$ , et  $JS = J - JL$ . Partition[] déduit les paramètres pour partager un bloc de taille I en J blocs de taille approximativement égale. Précisément, JL blocs de longueur IL et JS blocs de longueur IS.

L'objet de source DOIT être partagé en blocs de source et sous blocs comme suit :

Soit

$$K_t = \text{ceil}(F/T)$$

$$(K_L, K_S, Z_L, Z_S) = \text{Partition}[K_t, Z]$$

$$(T_L, T_S, N_L, N_S) = \text{Partition}[T/Al, N]$$

Ensuite, l'objet DOIT être partagé en  $Z = Z_L + Z_S$  blocs de source contigus, les  $Z_L$  premiers blocs de source ayant chacun une longueur de  $K_L * T$  octets, et les  $Z_S$  blocs de source restants ayant chacun  $K_S * T$  octets.

Si  $K_t * T > F$ , alors pour les besoins du codage, le dernier symbole DOIT être bourré jusqu'à la fin avec  $K_t * T - F$  octets de zéros.

Ensuite, chaque bloc de source DOIT être divisé en  $N = N_L + N_S$  sous blocs contigus, les  $N_L$  premiers sous blocs consistant chacun en  $K$  sous symboles contigus de taille  $T_L * Al$  et les  $N_S$  sous blocs restants consistant chacun en  $K$  sous symboles contigus de taille  $T_S * Al$ . Le paramètre d'alignement de symbole Al assure que les sous symboles sont toujours un multiple de Al octets.

Finalement, le mième symbole d'un bloc de source consiste en l'enchaînement du mième sous symbole provenant de chacun des N sous blocs. Noter que cela implique que quand  $N > 1$ , un symbole n'est alors PAS une portion contiguë de l'objet.

### 5.3.2 Construction du paquet de codage

Chaque paquet de codage contient les informations suivantes :

- le numéro de bloc de source (SBN, *Source Block Number*)
- l'identifiant de symbole de codage (ESI, *Encoding Symbol ID*)
- le ou les symboles de codage.

Chaque bloc de source est codé indépendamment des autres. Les blocs de source sont numérotés consécutivement à partir de zéro.

Les valeurs d'ESI de 0 à K-1 identifient les symboles de source d'un bloc de source en ordre séquentiel, où K est le nombre de symboles dans le bloc de source. Les ESI à partir de K identifient les symboles de réparation.

Chaque paquet de codage consiste soit entièrement en symboles de source (paquet de source) soit entièrement en symboles de réparation (paquet de réparation). Un paquet peut contenir un nombre quelconque de symboles provenant du même bloc de source. Dans le cas où le dernier symbole de source dans un paquet de source comporte des octets de bourrage ajoutés pour les besoins du codage de FEC, ces octets ne doivent pas être inclus dans le paquet. Autrement, seulement les symboles complets DOIVENT être inclus.

L'ESI, X, porté dans chaque paquet de source est l'identifiant de symbole de codage du premier symbole de source porté dans ce paquet. Les symboles de source suivants dans le paquet ont les ESI, X+1 à X+G-1, à la suite, où G est le nombre de symboles dans le paquet.

De même, l'ESI X, placé dans un paquet de réparation est l'ESI du premier symbole de réparation dans le paquet de réparation et les symboles de réparation suivants dans le paquet ont les ESI X+1 à X+G-1 en ordre séquentiel, où G est le nombre de symboles dans le paquet.

Noter qu'il n'est pas nécessaire que le receveur sache le nombre total de paquets de réparation.

Associé à chaque symbole est un triplet d'entiers (d, a, b).

Les triplets de symbole de réparation G (d[0], a[0], b[0]),..., (d[G-1], a[G-1], b[G-1]) pour les symboles de réparation placés dans un paquet de réparation avec ESI X sont calculés en utilisant le générateur de triplets défini au paragraphe 5.4.4.4 comme suit :

Pour chaque  $i = 0, \dots, G-1$ ,  $(d[i], a[i], b[i]) = \text{Trip}[K, X+i]$

Les symboles de réparation G à placer dans un paquet de réparation avec l'ESI X sont calculés sur la base des triplets de symboles de réparation, comme décrit au paragraphe 5.4, en utilisant les symboles intermédiaires C et le codeur LT  $\text{LTEnc}[K, C, (d[i], a[i], b[i])]$ .

## 5.4 Codeur Raptor systématique

### 5.4.1 Vue d'ensemble du codage

Le codeur systématique Raptor est utilisé pour générer les symboles de réparation à partir d'un bloc de source qui consiste en K symboles de source.

Les symboles sont les unités de données fondamentales du processus de codage et décodage. Pour chaque bloc de source (sous bloc) tous les symboles (sous symboles) sont de la même taille. L'opération atomique effectuée sur les symboles (sous symboles) pour le codage et le décodage est l'opération OU exclusif.

Soit  $C'[0], \dots, C'[K-1]$  qui note les K symboles de source.

Soit  $C[0], \dots, C[L-1]$  qui note les L symboles intermédiaires.

La première étape de codage est de générer un nombre,  $L > K$ , de symboles intermédiaires à partir des K symboles de source. Dans cette étape, K triplets de symboles de source (d[0], a[0], b[0]), ..., (d[K-1], a[K-1], b[K-1]) sont générés en utilisant le générateur  $\text{Trip}[]$  décrit au paragraphe 5.4.2.2. Les K triplets de symboles de source sont associés aux K symboles de source et sont alors utilisés pour déterminer les L symboles intermédiaires  $C[0], \dots, C[L-1]$  à partir des symboles de source en utilisant un processus de codage inverse. Ce processus peut être réalisé par un processus de décodage Raptor.

Certaines "relations de précodage" DOIVENT être réalisées au sein des L symboles intermédiaires. Le paragraphe 5.4.2.3 décrit ces relations et comment les symboles intermédiaires sont générés à partir des symboles de source.

Une fois que les symboles intermédiaires ont été générés, les symboles de réparation sont produits et un ou plusieurs symboles de réparation sont placés comme un groupe dans un seul paquet de données. Chaque groupe de symboles de réparation est associé à un ESI et un certain nombre, G, de symboles de réparation. L'ESI est utilisé pour générer un triplet d'entiers, (d, a, b) pour chaque symbole de réparation, là encore en utilisant le générateur  $\text{Trip}[]$  décrit au paragraphe 5.4.4.4. Ensuite, chaque triplet (d,a,b) est utilisé pour générer le symbole de réparation correspondant à partir des symboles intermédiaires en utilisant le générateur  $\text{LTEnc}[K, C[0], \dots, C[L-1], (d,a,b)]$  décrit au paragraphe 5.4.4.3.

### 5.4.2 Première étape de codage : génération de symboles intermédiaires

#### 5.4.2.1 Généralités

La première étape du codage est une étape de précodage pour générer les L symboles intermédiaires  $C[0], \dots, C[L-1]$  à partir des symboles de source  $C'[0], \dots, C'[K-1]$ . Les symboles intermédiaires sont définis de façon univoque par deux ensembles de contraintes :

1. Les symboles intermédiaires sont en relation avec les symboles de source par un ensemble de triplets de symboles de source. La génération des triplets de symboles de source est définie au paragraphe 5.4.2.2 en utilisant le générateur

Trip[] décrit au paragraphe 5.4.4.4.

2. Un ensemble de relations de précodage est en vigueur entre les symboles intermédiaires eux-mêmes. Elles sont définies au paragraphe 5.4.2.3.

La génération des L symboles intermédiaires est alors définie au paragraphe 5.4.2.4

#### 5.4.2.2 Triplets de symboles de source

Chacun des K symboles de source est associé à un triplet (d[i], a[i], b[i]) pour  $0 \leq i < K$ . Les triplets de symboles de source sont déterminés en utilisant le générateur de triplets défini au paragraphe 5.4.4.4 comme :

Pour chaque i,  $0 \leq i < K$ , (d[i], a[i], b[i]) = Trip[K, i]

#### 5.4.2.3 Relations de pré codage

Les relations de précodage entre les L symboles intermédiaires sont définies en exprimant les L-K derniers symboles intermédiaires dans les termes des K premiers symboles intermédiaires.

Les L-K derniers symboles intermédiaires C[K],...,C[L-1] consistent en S symboles LDPC et H demi symboles. Les valeurs de S et H sont déterminées à partir de K comme décrit ci-dessous. Alors  $L = K+S+H$ .

Soient

X le plus petit entier positif tel que  $X*(X-1) \geq 2*K$ .

S le plus petit nombre premier entier tel que  $S \geq \text{ceil}(0.01*K) + X$ .

H le plus petit nombre entier tel que  $\text{choose}(H, \text{ceil}(H/2)) \geq K + S$ .

$H' = \text{ceil}(H/2)$ .

$L = K+S+H$ .

C[0],...,C[K-1] note les K premiers symboles intermédiaires,

C[K],...,C[K+S-1] note les S symboles LDPC, initialisés à zéro,

C[K+S],...,C[L-1] note les H demi symboles, initialisés à zéro.

Les S symboles LDPC sont définis comme étant les valeurs de C[K],...,C[K+S-1] à la fin du processus suivant :

Pour  $i = 0, \dots, K-1$  faire

$a = 1 + (\text{floor}(i/S) \% (S-1))$

$b = i \% S$

$C[K + b] = C[K + b] \wedge C[i]$

$b = (b + a) \% S$

$C[K + b] = C[K + b] \wedge C[i]$

$b = (b + a) \% S$

$C[K + b] = C[K + b] \wedge C[i]$

Les H demi symboles sont définis comme suit :

Soit

$g[i] = i \wedge (\text{floor}(i/2))$  pour tous les entiers positifs i. Note : g[i] est la séquence de Gray, dans laquelle chaque élément diffère du précédent d'une seule position de bit.

m[k] note la sous séquence de g[...] dont les éléments ont exactement k bits non à zéro dans leur représentation binaire.

m[j,k] note le j-ième élément de la séquence m[k], où  $j=0, 1, 2, \dots$

Ensuite, les demi symboles sont définis comme les valeurs de C[K+S],...,C[L-1] après le processus suivant :

Pour  $h = 0, \dots, H-1$  faire

Pour  $j = 0, \dots, K+S-1$  faire

Si le bit h de m[j,H'] est égal à 1, alors  $C[h+K+S] = C[h+K+S] \wedge C[j]$ .

#### 5.4.2.4 Symboles intermédiaires

##### 5.4.2.4.1 Définition

Étant donnés les K symboles de source C'[0], C'[1],..., C'[K-1], les L symboles intermédiaires C[0], C[1],..., C[L-1] sont les

valeurs uniques de symbole définies qui satisfont les conditions suivantes :

1. Les K symboles de source  $C'[0], C'[1], \dots, C'[K-1]$  satisfont les K contraintes  $C'[i] = \text{LEnc}[K, (C[0], \dots, C[L-1]), (d[i], a[i], b[i])]$ , pour tous les  $i, 0 \leq i < K$ .
2. Les L symboles intermédiaires  $C[0], C[1], \dots, C[L-1]$  satisfont les relations de précodage définies au paragraphe 5.4.2.3.

#### 5.4.2.4.2 Exemple de méthode de calcul des symboles intermédiaires

Ce paragraphe décrit une méthode possible pour le calcul des L symboles intermédiaires  $C[0], C[1], \dots, C[L-1]$  satisfaisant les contraintes du paragraphe 5.4.2.4.1.

La "matrice de générateur" pour un code qui génère N symboles de résultat à partir de K symboles d'entrée est une matrice  $N \times K$  sur  $\text{GF}(2)$ , où chaque rangée correspond à un des symboles de résultat et chaque colonne à un des symboles d'entrée et où le i-ème symbole de résultat est égal à la somme de ces symboles d'entrée dont la colonne contient une entrée non zéro dans la rangée i.

Ensuite, les L symboles intermédiaires peuvent être calculés comme suit :

Soit

C qui note le vecteur de colonne des L symboles intermédiaires,  $C[0], C[1], \dots, C[L-1]$ .

D qui note le vecteur de colonne consistant en S+H symboles zéro suivis par les K symboles de source  $C'[0], C'[1], \dots, C'[K-1]$ .

Ensuite les contraintes ci-dessus définissent une matrice  $L \times L$  sur  $\text{GF}(2)$ , A, telle que  $A * C = D$

La matrice A peut être construite comme suit :

Soient :

$G\_LDPC$  la matrice de générateur  $S \times K$  des symboles LDPC. Donc,  $G\_LDPC * \text{Transpose}[(C[0], \dots, C[K-1])] = \text{Transpose}[(C[K], \dots, C[K+S-1])]$  ;

$G\_Half$  la matrice de générateur  $H \times (K+S)$  des demi symboles, Donc,  $G\_Half * \text{Transpose}[(C[0], \dots, C[S+K-1])] = \text{Transpose}[(C[K+S], \dots, C[K+S+H-1])]$  ;

$I\_S$  la matrice d'identité  $S \times S$  ;

$I\_H$  la matrice d'identité  $H \times H$  ;

$0\_S \times H$  la matrice zéro  $S \times H$  ;

$G\_LT$  la matrice de générateur  $K \times L$  des symboles de codage générés par le codeur LT. Donc,  $G\_LT * \text{Transpose}[(C[0], \dots, C[L-1])] = \text{Transpose}[(C'[0], C'[1], \dots, C'[K-1])]$  ;

c'est-à-dire,  $G\_LT(i,j) = 1$  si et seulement si  $C[j]$  est inclus dans les symboles qui sont OUXés pour produire  $\text{LEnc}[K, (C[0], \dots, C[L-1]), (d[i], a[i], b[i])]$ .

Alors :

Les S premières rangées de A sont égales à  $G\_LDPC | I\_S | 0\_S \times H$ .

Les H prochaines rangées de A sont égales à  $G\_Half | I\_H$ .

Les K rangées restantes de A sont égales à  $G\_LT$ .

La matrice A est décrite dans la Figure 4 ci-dessous :

	K	S	H
S	G_LDPC	I_S	0_SxH
H	G_Half	I_H	
K	G_LT		

Figure 4 : matrice A

Les symboles intermédiaires peuvent alors être calculés comme :  $C = (A^{-1}) * D$

Les triplets de symboles de source sont générés de façon telle que pour toute matrice  $K$ ,  $A$  a un rang complet et est donc réversible. Ce calcul peut être réalisé en appliquant un processus de décodage Raptor aux  $K$  symboles de source  $C[0]$ ,  $C[1]$ ,...,  $C[K-1]$  pour produire les  $L$  symboles intermédiaires  $C[0]$ ,  $C[1]$ ,...,  $C[L-1]$ .

Pour générer efficacement les symboles intermédiaires à partir des symboles de source, il est recommandé qu'une mise en œuvre efficace de décodeur telle que celle décrite au paragraphe 5.5 soit utilisée. Les triplets de symboles de source sont conçus pour faciliter un décodage efficace des symboles de source en utilisant cet algorithme.

### 5.4.3 Seconde étape de codage : codage LT

Dans la seconde étape de codage, le symbole de réparation avec l'ESI  $X$  est généré en appliquant le générateur  $LTEnc[K, (C[0], C[1], \dots, C[L-1]), (d, a, b)]$  défini au paragraphe 5.4.4.3 aux  $L$  symboles intermédiaires  $C[0]$ ,  $C[1]$ ,...,  $C[L-1]$  en utilisant le triplet  $(d, a, b) = Trip[K, X]$  généré conformément au paragraphe 5.3.2

### 5.4.4 Générateurs

#### 5.4.4.1 Générateur aléatoire

Le générateur de nombres aléatoires  $Rand[X, i, m]$  est défini comme suit, où  $X$  est un entier non négatif,  $i$  est un entier non négatif, et  $m$  est un entier positif et la valeur produite est un entier entre 0 et  $m-1$ . Soient  $V0$  et  $V1$  des matrices de 256 entrées chacune, où chaque entrée est un entier non signé de 4 octets. Ces matrices sont données au paragraphe 5.6.

Alors,  $Rand[X, i, m] = (V0[(X + i) \% 256] \wedge V1[(\text{floor}(X/256) + i) \% 256]) \% m$

#### 5.4.4.2 Générateur de degré

Le générateur de degré  $Deg[v]$  est défini comme suit, où  $v$  est un entier qui est au moins 0 et inférieur à  $2^{20} = 1048576$ .

Dans le Tableau 1, on trouve l'indice  $j$  tel que  $f[j-1] \leq v < f[j]$

Ensuite,  $Deg[v] = d[j]$

Indice $j$	$f[j]$	$d[j]$
0	0	--
1	10241	1
2	491582	2
3	712794	3
4	831695	4
5	948446	10
6	1032189	11
7	1048576	40

Tableau 1 : Définition du degré de distribution pour les symboles de codage

#### 5.4.4.3 Générateur de symbole de codage LT

Le générateur de symboles de codage  $LTEnc[K, (C[0], C[1], \dots, C[L-1]), (d, a, b)]$  prend les entrées suivantes :

$K$  est le nombre de symboles de source (ou sous symboles) pour le bloc de source (sous bloc). Soit  $L'$  déduit de  $K$  comme décrit au paragraphe 5.4.2.3, et soit  $L'$  le plus petit entier premier supérieur ou égal à  $L$ .

$(C[0], C[1], \dots, C[L-1])$  est la matrice des  $L$  symboles intermédiaires (sous symboles) générée comme décrit au paragraphe 5.4.2.4.

$(d, a, b)$  est un triplet de source déterminé en utilisant le générateur de triplets défini au paragraphe 5.4.4.4, dans lequel  $d$  est un entier notant un degré de symbole de codage,

$a$  est un entier entre 1 et  $L'-1$  inclus,

$b$  est un entier entre 0 et  $L'-1$  inclus.

Le générateur de symboles de codage produit un seul symbole de codage en résultat, en accord avec l'algorithme suivant :

Quand  $(b \geq L)$  faire  $b = (b + a) \% L'$   
 Soit le résultat =  $C[b]$ .  
 Pour  $j = 1, \dots, \min(d-1, L-1)$  faire  
 $b = (b + a) \% L'$   
 Quand  $(b \geq L)$  faire  $b = (b + a) \% L'$   
 résultat = résultat  $\wedge C[b]$   
 Retourner le résultat

#### 5.4.4.4 Générateur de triplets

Le générateur de triplets  $\text{Trip}[K, X]$  prend les entrées suivantes :

$K$  : nombre de symboles de source  
 $X$  : identifiant de symbole de codage

Soit

$L$  déterminé à partir de  $K$  comme décrit au paragraphe 5.4.2.3.

$L'$  le plus petit premier supérieur ou égal à  $L$ .

$Q = 65521$ , le plus grand premier plus petit que  $2^{16}$ .

$J(K)$  l'indice systématique associé à  $K$ , comme défini au paragraphe 5.7.

Le résultat du générateur de triplets est un triplet,  $(d, a, b)$  déterminé comme suit :

$A = (53591 + J(K) * 997) \% Q$

$B = 10267 * (J(K) + 1) \% Q$

$Y = (B + X * A) \% Q$

$v = \text{Rand}[Y, 0, 2^{20}]$

$d = \text{Deg}[v]$

$a = 1 + \text{Rand}[Y, 1, L'-1]$

$b = \text{Rand}[Y, 2, L']$

## 5.5 Exemple de décodeur de FEC

### 5.5.1 Généralités

Ce paragraphe décrit un algorithme de décodage efficace pour les codes Raptor décrits dans cette spécification. Noter que chaque symbole de codage reçu peut être considéré comme la valeur d'une équation entre les symboles intermédiaires. De ces équations simultanées, et des relations de précodage connues entre les symboles intermédiaires, tout algorithme pour résoudre les équations simultanées peut réussir à décoder les symboles intermédiaires et donc les symboles de source. Cependant, l'algorithme choisi a un effet majeur sur l'efficacité du calcul du décodage.

### 5.5.2 Décodage d'un bloc source

#### 5.5.2.1 Généralités

On suppose que le décodeur connaît la structure du bloc de source qu'il va décoder, y compris la taille de symbole,  $T$ , et le nombre  $K$  de symboles dans le bloc de source.

À partir des algorithmes décrits au paragraphe 5.4, le décodeur Raptor peut calculer le nombre total  $L = K + S + H$  des symboles de précodage et déterminer comment ils ont été générés à partir du bloc de source pour être décodés. Dans cette description, on suppose que les symboles de codage reçus pour le bloc de source à décoder sont passés au décodeur. Noter que, comme décrit au paragraphe 5.3.2, le dernier symbole de source d'un paquet de source peut inclure des octets de bourrage ajoutés pour les besoins du codage de FEC. Ces octets de bourrage ne peuvent pas être réellement inclus dans le paquet envoyé et doivent donc être réinsérés à réception avant de passer le symbole au décodeur.

Pour chacun de ces symboles de codage, on suppose que le nombre et l'ensemble de symboles intermédiaires dont le OUX est égal au symbole de codage sont aussi passés au décodeur. Dans le cas de symboles de source, les triplets de symboles de source décrits au paragraphe 5.4.2.2 indiquent le nombre et l'ensemble de symboles intermédiaires qui s'ajoutent pour donner chaque symbole de source.

Soit  $N \geq K$  le nombre de symboles de codage reçus pour un bloc de source et soit  $M = S+H+N$ . La matrice de bits  $M$  par  $L$  suivante  $A$  peut être déduite des informations passées au décodeur pour le bloc de source à décoder. Soit  $C$  le vecteur de colonne des  $L$  symboles intermédiaires, et soit  $D$  le vecteur de colonne des  $M$  symboles avec les valeurs connues du receveur, où les  $S+H$  premiers des  $M$  symboles sont des symboles de valeur zéro qui correspondent aux symboles LDPC et demi symboles (ce sont des symboles de vérification pour les symboles LDPC et demi symboles, et non les symboles LDPC et demi symboles eux mêmes) et les  $N$  restants des  $M$  symboles sont les symboles de codage reçus pour le bloc de source. Ensuite,  $A$  est la matrice de bits qui satisfait  $A * C = D$ , où  $*$  note la multiplication matricielle sur  $GF[2]$ . En particulier,  $A[i,j] = 1$  si le symbole intermédiaire correspondant à l'indice  $j$  est OUXé dans le LDPC, demi, ou symbole de codage correspondant à l'indice  $i$  dans le codage, ou si l'indice  $i$  correspond à un symbole LDPC ou demi symbole et l'indice  $j$  correspond au même symbole LDPC ou demi symbole. Pour tous les autres  $i$  et  $j$ ,  $A[i,j] = 0$ .

Décoder un bloc de source est équivalent à décoder  $C$  à partir de  $A$  et  $D$  connus. Il est clair que  $C$  peut être décodé si et seulement si le rang de  $A$  sur  $GF[2]$  est  $L$ . Une fois que  $C$  a été décodé, les symboles de source manquants peuvent être obtenus en utilisant les triplets de symboles de source pour déterminer le nombre et l'ensemble de symboles intermédiaires qui DOIVENT être OUXés pour obtenir chaque symbole de source manquant.

La première étape du décodage de  $C$  est de former un plan de décodage. Dans cette étape  $A$  est converti, en utilisant l'élimination gaussienne (en utilisant des opérations de rangées et des réarrangements de rangées et de colonnes) et après l'élimination de  $M - L$  rangées, dans la matrice d'identité  $L$  par  $L$ . Le plan de décodage consiste en la séquence d'opérations de rangées et de réarrangements de rangées et colonnes durant le processus d'élimination gaussienne, et dépend seulement de  $A$  et non de  $D$ . Le décodage de  $C$  à partir de  $D$  peut avoir lieu concurremment avec la formation du plan de décodage, ou le décodage peut avoir lieu ensuite sur la base du plan de décodage.

La correspondance entre le plan de décodage et le décodage de  $C$  est comme suit. Soit initialement  $c[0] = 0, c[1] = 1, \dots, c[L-1] = L-1$  et  $d[0] = 0, d[1] = 1, \dots, d[M-1] = M-1$ .

- Chaque fois que la rangée  $i$  de  $A$  est OUXée dans la rangée  $i'$  dans le plan de décodage, alors dans le processus de décodage, le symbole  $D[d[i]]$  est OUXé dans le symbole  $D[d[i']]$ .
- Chaque fois que la rangée  $i$  est échangée avec la rangée  $i'$  dans le plan de décodage, alors dans le processus de décodage, la valeur de  $d[i]$  est échangée avec la valeur de  $d[i']$ .
- Chaque fois que la colonne  $j$  est échangée avec la colonne  $j'$  dans le plan de décodage, alors dans le processus de décodage, la valeur de  $c[j]$  est échangée avec la valeur de  $c[j']$ .

À partir de cette correspondance, il est clair que le nombre total de OUXages de symboles dans le décodage de blocs de source est le nombre d'opérations de rangées (non d'échanges) dans l'élimination gaussienne. Comme  $A$  est la matrice d'identité  $L$  par  $L$  après l'élimination gaussienne et après l'élimination des  $M - L$  dernières rangées, il est clair à la fin du décodage réussi que les  $L$  symboles  $D[d[0]], D[d[1]], \dots, D[d[L-1]]$  sont les valeurs des  $L$  symboles  $C[c[0]], C[c[1]], \dots, C[c[L-1]]$ .

L'ordre dans lequel l'élimination gaussienne est effectuée pour former le plan de décodage n'a pas d'impact sur la réussite ou non du décodage. Cependant, la vitesse du décodage dépend largement de l'ordre dans lequel l'élimination gaussienne est effectuée. (De plus, maintenir une représentation frugale de  $A$  est crucial, bien que ce ne soit pas décrit ici). Le reste de cette section décrit l'ordre dans lequel l'élimination gaussienne pourrait être effectuée de façon relativement efficace.

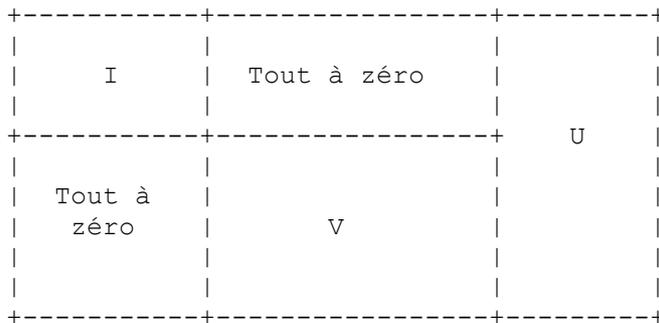
### 5.5.2.2 Première phase

À la première phase de l'élimination gaussienne, la matrice  $A$ , est conceptuellement partagée en sous matrices. Les tailles des sous matrices sont paramétrées par des entiers non négatifs  $i$  et  $u$ , qui sont initialisés à 0. Les sous matrices de  $A$  sont :

- (1) La sous matrice  $I$  définie par l'intersection des  $i$  premières rangées et des  $i$  premières colonnes. C'est la matrice d'identité à la fin de chaque étape de la phase.
- (2) La sous matrice définie par l'intersection des  $i$  premières rangées et de toutes sauf les  $i$  premières colonnes et des  $u$  dernières colonnes. Toutes les entrées de cette sous matrice sont à zéro.
- (3) La sous matrice définie par l'intersection des  $i$  premières colonnes et de toutes sauf les  $i$  premières rangées. Toutes les entrées de cette sous matrice sont à zéro.

- (4) La sous matrice  $U$  définie par l'intersection de toutes les rangées et des  $u$  dernières colonnes.
- (5) La sous matrice  $V$  formée par l'intersection de toutes sauf les  $i$  premières colonnes et les  $u$  dernières colonnes et toutes sauf les  $i$  premières rangées.

La Figure 5 illustre les sous matrices de  $A$ . Au début de la première phase,  $V = A$ . À chaque étape, une rangée de  $A$  est choisie.



**Figure 5 : Sous matrices de  $A$  dans la première phase**

Le graphe suivant défini par la structure de  $V$  est utilisé pour déterminer quelle rangée de  $A$  est choisie. Les colonnes qui coupent  $V$  sont les nœuds du graphe, et les rangées qui ont exactement 2 uns dans  $V$  sont les bords du graphe qui connectent les deux colonnes (nœuds) dans les positions des deux uns. Un composant dans ce graphe est un ensemble maximal de nœuds (colonnes) et de bords (rangées) tel que il y a un chemin entre chaque paire de nœuds/bords dans le graphe. La taille d'un composant est le nombre de nœuds (colonnes) dans le composant.

Il y a au plus  $L$  étapes dans la première phase. La phase se termine avec succès quand  $i + u = L$ , c'est-à-dire, quand  $V$  et la sous matrice toute de zéros au-dessus de  $V$  a disparu et  $A$  consiste en  $I$ , la sous matrice toute de zéros au-dessus de  $I$ , et  $U$ . La phase se termine sans succès en échec de décodage si, à une étape avant que  $V$  disparaisse, il n'y a pas de rangée non à zéro dans  $V$  à choisir à cette étape. Chaque fois que il y a des rangées non à zéro dans  $V$ , l'étape suivante commence alors par le choix d'une rangée de  $A$  comme suit :

- o Soit  $r$  l'entier minimum tel que au moins une rangée de  $A$  a exactement  $r$  uns dans  $V$ .
- \* Si  $r \neq 2$ , on choisit alors une rangée avec exactement  $r$  uns dans  $V$  avec le degré minimum original parmi toutes les rangées.
- \* Si  $r = 2$ , on choisit alors une rangée quelconque avec exactement 2 uns dans  $V$  qui fait partie d'un composant de taille maximum dans le graphe défini par  $V$ .

Après le choix de la rangée dans cette étape, la première rangée de  $A$  qui coupe  $V$  est échangée avec la rangée choisie afin que la rangée choisie soit la première rangée qui intersecte  $V$ . Les colonnes de  $A$  parmi celles qui intersectent  $V$  sont réordonnées afin que une des  $r$  de la rangée choisie apparaisse dans la première colonne de  $V$  et que les  $r-1$  restantes apparaissent dans la dernière colonne de  $V$ . Ensuite, la rangée choisie est OUXée dans toutes les autres rangées de  $A$  en dessous de la rangée choisie qui ont un un dans la première colonne de  $V$ . Finalement,  $i$  est incrémenté de 1 et  $u$  est incrémenté de  $r-1$ , ce qui termine l'étape.

### 5.5.2.3 Seconde phase

La sous matrice  $U$  est encore partagée en les  $i$  premières rangées,  $U_{\text{upper}}$ , et les  $M - i$  rangées restantes,  $U_{\text{lower}}$ . L'élimination gaussienne est effectuée dans la seconde phase sur  $U_{\text{lower}}$  pour déterminer que son rang est inférieur à  $u$  (échec de décodage) ou pour le convertir en une matrice où les  $u$  premières rangées sont la matrice d'identité (succès de la seconde phase). On l'appelle la matrice d'identité  $I_u$  par  $u$ . Les  $M - L$  rangées de  $A$  qui intersectent  $U_{\text{lower}} - I_u$  sont éliminées. Après cette phase,  $A$  a  $L$  rangées et  $L$  colonnes.

### 5.5.2.4 Troisième phase

Après la seconde phase, la seule portion de  $A$  qui a besoin d'être mise à zéro pour finir de convertir  $A$  en la matrice

d'identité  $L$  par  $L$  est  $U_{\text{upper}}$ . Le nombre de rangées  $i$  de la sous matrice  $U_{\text{upper}}$  est généralement beaucoup plus grand que le nombre des  $u$  colonnes de  $U_{\text{upper}}$ . Pour mettre à zéro efficacement  $U_{\text{upper}}$ , la matrice de précalcul  $U'$  est calculée sur la base de  $I_u$  dans la troisième phase et ensuite  $U'$  est utilisé dans la quatrième phase pour mettre à zéro  $U_{\text{upper}}$ . Les  $u$  rangées de  $I_u$  sont partagées en  $\text{ceil}(u/8)$  groupes de 8 rangées chacun. Ensuite, pour chaque groupe de 8 rangées, toutes les combinaisons non à zéro des 8 rangées sont calculées, résultant en  $2^8 - 1 = 255$  rangées (cela peut être fait avec  $2^{8-8-1} = 247$  OUX des rangées par groupe, car les combinaisons de Hamming de poids un qui apparaissent dans  $I_u$  n'ont pas besoin d'être recalculées). Donc, la matrice de précalcul résultante  $U'$  a  $\text{ceil}(u/8) * 255$  rangées et  $u$  colonnes. Noter que  $U'$  ne fait pas formellement partie de la matrice  $A$ , mais va être utilisée dans la quatrième phase pour mettre à zéro  $U_{\text{upper}}$ .

### 5.5.2.5 Quatrième phase

Pour chacune des  $i$  premières rangées de  $A$ , pour chaque groupe de 8 colonnes dans la sous matrice  $U_{\text{upper}}$  de cette rangée, si l'ensemble des entrées de 8 colonnes dans  $U_{\text{upper}}$  n'est pas tout à zéro, alors la rangée de la matrice de précalcul  $U'$  qui correspond au schéma dans les 8 colonnes est OUXé dans la rangée, donc mettant à zéro ces 8 colonnes dans la rangée au prix d'un OUX d'une rangée de  $U'$  dans la rangée.

Après cette phase,  $A$  est la matrice d'identité  $L$  par  $L$  et un plan de décodage complet a été formé avec succès. Ensuite, comme expliqué au paragraphe 5.5.2.1, le décodage correspondant, consistant en le OUX des symboles de codage connus, peut être exécuté pour récupérer les symboles intermédiaires sur la base du plan de décodage. Les triplets associés à tous les symboles de source sont calculés conformément au paragraphe 5.4.2.2. Les triplets pour les symboles de source reçus sont utilisés dans le décodage. Les triplets pour les symboles de source manquants sont utilisés pour déterminer quels symboles intermédiaires ont besoin d'être OUXés pour récupérer les symboles de source manquants.

## 5.6 Nombres aléatoires

Les deux tableaux  $V0$  et  $V1$  décrits au paragraphe 5.4.4.1 sont donnés ci-dessous. Chaque entrée est un entier de 32 bits en représentation décimale.

### 5.6.1 Tableau $V0$

251291136, 3952231631, 3370958628, 4070167936, 123631495, 3351110283, 3218676425, 2011642291, 774603218, 2402805061, 1004366930, 1843948209, 428891132, 3746331984, 1591258008, 3067016507, 1433388735, 504005498, 2032657933, 3419319784, 2805686246, 3102436986, 3808671154, 2501582075, 3978944421, 246043949, 4016898363, 649743608, 1974987508, 2651273766, 2357956801, 689605112, 715807172, 2722736134, 191939188, 3535520147, 3277019569, 1470435941, 3763101702, 3232409631, 122701163, 3920852693, 782246947, 372121310, 2995604341, 2045698575, 2332962102, 4005368743, 218596347, 3415381967, 4207612806, 861117671, 3676575285, 2581671944, 3312220480, 681232419, 307306866, 4112503940, 1158111502, 709227802, 2724140433, 4201101115, 4215970289, 4048876515, 3031661061, 1909085522, 510985033, 1361682810, 129243379, 3142379587, 2569842483, 3033268270, 1658118006, 932109358, 1982290045, 2983082771, 3007670818, 3448104768, 683749698, 778296777, 1399125101, 1939403708, 1692176003, 3868299200, 1422476658, 593093658, 1878973865, 2526292949, 1591602827, 3986158854, 3964389521, 2695031039, 1942050155, 424618399, 1347204291, 2669179716, 2434425874, 2540801947, 1384069776, 4123580443, 1523670218, 2708475297, 1046771089, 2229796016, 1255426612, 4213663089, 1521339547, 3041843489, 420130494, 10677091, 515623176, 3457502702, 2115821274, 2720124766, 3242576090, 854310108, 425973987, 325832382, 1796851292, 2462744411, 1976681690, 1408671665, 1228817808, 3917210003, 263976645, 2593736473, 2471651269, 4291353919, 650792940, 1191583883, 3046561335, 2466530435, 2545983082, 969168436, 2019348792, 2268075521, 1169345068, 3250240009, 3963499681, 2560755113, 911182396, 760842409, 3569308693, 2687243553, 381854665, 2613828404, 2761078866, 1456668111, 883760091, 3294951678, 1604598575, 1985308198, 1014570543, 2724959607, 3062518035, 3115293053, 138853680, 4160398285, 3322241130, 2068983570, 2247491078, 3669524410, 1575146607, 828029864, 3732001371, 3422026452, 3370954177, 4006626915, 543812220, 1243116171, 3928372514, 2791443445, 4081325272, 2280435605, 885616073, 616452097, 3188863436, 2780382310, 2340014831, 1208439576, 258356309, 3837963200, 2075009450, 3214181212, 3303882142, 880813252, 1355575717, 207231484, 2420803184, 358923368, 1617557768, 3272161958, 1771154147, 2842106362, 1751209208, 1421030790, 658316681, 194065839, 3241510581, 38625260, 301875395, 4176141739, 297312930, 2137802113, 1502984205, 3669376622, 3728477036, 234652930, 2213589897, 2734638932, 1129721478, 3187422815, 2859178611, 3284308411, 3819792700, 3557526733, 451874476, 1740576081, 3592838701, 1709429513, 3702918379, 3533351328, 1641660745, 179350258, 2380520112, 3936163904, 3685256204, 3156252216, 1854258901, 2861641019, 3176611298, 834787554, 331353807, 517858103, 3010168884, 4012642001, 2217188075, 3756943137, 3077882590, 2054995199, 3081443129, 3895398812, 1141097543, 2376261053, 2626898255, 2554703076, 401233789, 1460049922, 678083952, 1064990737, 940909784, 1673396780, 528881783, 1712547446, 3629685652, 1358307511

**5.6.2 Tableau V1**

807385413, 2043073223, 3336749796, 1302105833, 2278607931, 541015020, 1684564270, 372709334, 3508252125, 1768346005, 1270451292, 2603029534, 2049387273, 3891424859, 2152948345, 4114760273, 915180310, 3754787998, 700503826, 2131559305, 1308908630, 224437350, 4065424007, 3638665944, 1679385496, 3431345226, 1779595665, 3068494238, 1424062773, 1033448464, 4050396853, 3302235057, 420600373, 2868446243, 311689386, 259047959, 4057180909, 1575367248, 4151214153, 110249784, 3006865921, 4293710613, 3501256572, 998007483, 499288295, 1205710710, 2997199489, 640417429, 3044194711, 486690751, 2686640734, 2394526209, 2521660077, 49993987, 3843885867, 4201106668, 415906198, 19296841, 2402488407, 2137119134, 1744097284, 579965637, 2037662632, 852173610, 2681403713, 1047144830, 2982173936, 910285038, 4187576520, 2589870048, 989448887, 3292758024, 506322719, 176010738, 1865471968, 2619324712, 564829442, 1996870325, 339697593, 4071072948, 3618966336, 2111320126, 1093955153, 957978696, 892010560, 1854601078, 1873407527, 2498544695, 2694156259, 1927339682, 1650555729, 183933047, 3061444337, 2067387204, 228962564, 3904109414, 1595995433, 1780701372, 2463145963, 307281463, 3237929991, 3852995239, 2398693510, 3754138664, 522074127, 146352474, 4104915256, 3029415884, 3545667983, 332038910, 976628269, 3123492423, 3041418372, 2258059298, 2139377204, 3243642973, 3226247917, 3674004636, 2698992189, 3453843574, 1963216666, 3509855005, 2358481858, 747331248, 1957348676, 1097574450, 2435697214, 3870972145, 1888833893, 2914085525, 4161315584, 1273113343, 3269644828, 3681293816, 412536684, 1156034077, 3823026442, 1066971017, 3598330293, 1979273937, 2079029895, 1195045909, 1071986421, 2712821515, 3377754595, 2184151095, 750918864, 2585729879, 4249895712, 1832579367, 1192240192, 946734366, 31230688, 3174399083, 3549375728, 1642430184, 1904857554, 861877404, 3277825584, 4267074718, 3122860549, 666423581, 644189126, 226475395, 307789415, 1196105631, 3191691839, 782852669, 1608507813, 1847685900, 4069766876, 3931548641, 2526471011, 766865139, 2115084288, 4259411376, 3323683436, 568512177, 3736601419, 1800276898, 4012458395, 1823982, 27980198, 2023839966, 869505096, 431161506, 1024804023, 1853869307, 3393537983, 1500703614, 3019471560, 1351086955, 3096933631, 3034634988, 2544598006, 1230942551, 3362230798, 159984793, 491590373, 3993872886, 3681855622, 903593547, 3535062472, 1799803217, 772984149, 895863112, 1899036275, 4187322100, 101856048, 234650315, 3183125617, 3190039692, 525584357, 1286834489, 455810374, 1869181575, 922673938, 3877430102, 3422391938, 1414347295, 1971054608, 3061798054, 830555096, 2822905141, 167033190, 1079139428, 4210126723, 3593797804, 429192890, 372093950, 1779187770, 3312189287, 204349348, 452421568, 2800540462, 3733109044, 1235082423, 1765319556, 3174729780, 3762994475, 3171962488, 442160826, 198349622, 45942637, 1324086311, 2901868599, 678860040, 3812229107, 19936821, 1119590141, 3640121682, 3545931032, 2102949142, 2828208598, 3603378023, 4135048896

**5.7 Indices systématiques J(K)**

Pour chaque valeur de K, l'indice systématique J(K) est destiné à avoir la propriété que l'ensemble des triplets de symboles de source (d[0], a[0], b[0]), ..., (d[L-1], a[L-1], b[L-1]) sont tels que les L symboles intermédiaires sont définis de façon univoque, c'est-à-dire, la matrice A du paragraphe 5.4.2.4.2 a un rang complet et est donc inversible.

Ci-après figure la liste des indices systématiques pour les valeurs de K entre 4 et 8192 inclus.

18, 14, 61, 46, 14, 22, 20, 40, 48, 1, 29, 40, 43, 46, 18, 8, 20, 2, 61, 26, 13, 29, 36, 19, 58, 5, 58, 0, 54, 56, 24, 14, 5, 67, 39, 31, 25, 29, 24, 19, 14, 56, 49, 49, 63, 30, 4, 39, 2, 1, 20, 19, 61, 4, 54, 70, 25, 52, 9, 26, 55, 69, 27, 68, 75, 19, 64, 57, 45, 3, 37, 31, 100, 41, 25, 41, 53, 23, 9, 31, 26, 30, 30, 46, 90, 50, 13, 90, 77, 61, 31, 54, 54, 3, 21, 66, 21, 11, 23, 11, 29, 21, 7, 1, 27, 4, 34, 17, 85, 69, 17, 75, 93, 57, 0, 53, 71, 88, 119, 88, 90, 22, 0, 58, 41, 22, 96, 26, 79, 118, 19, 3, 81, 72, 50, 0, 32, 79, 28, 25, 12, 25, 29, 3, 37, 30, 30, 41, 84, 32, 31, 61, 32, 61, 7, 56, 54, 39, 33, 66, 29, 3, 14, 75, 75, 78, 84, 75, 84, 25, 54, 25, 25, 107, 78, 27, 73, 0, 49, 96, 53, 50, 21, 10, 73, 58, 65, 27, 3, 27, 18, 54, 45, 69, 29, 3, 65, 31, 71, 76, 56, 54, 76, 54, 13, 5, 18, 142, 17, 3, 37, 114, 41, 25, 56, 0, 23, 3, 41, 22, 22, 31, 18, 48, 31, 58, 37, 75, 88, 3, 56, 1, 95, 19, 73, 52, 52, 4, 75, 26, 1, 25, 10, 1, 70, 31, 31, 12, 10, 54, 46, 11, 74, 84, 74, 8, 58, 23, 74, 8, 36, 11, 16, 94, 76, 14, 57, 65, 8, 22, 10, 36, 36, 96, 62, 103, 6, 75, 103, 58, 10, 15, 41, 75, 125, 58, 15, 10, 34, 29, 34, 4, 16, 29, 18, 18, 28, 71, 28, 43, 77, 18, 41, 41, 41, 62, 29, 96, 15, 106, 43, 15, 3, 43, 61, 3, 18, 103, 77, 29, 103, 19, 58, 84, 58, 1, 146, 32, 3, 70, 52, 54, 29, 70, 69, 124, 62, 1, 26, 38, 26, 3, 16, 26, 5, 51, 120, 41, 16, 1, 43, 34, 34, 29, 37, 56, 29, 96, 86, 54, 25, 84, 50, 34, 34, 93, 84, 96, 29, 29, 50, 50, 6, 1, 105, 78, 15, 37, 19, 50, 71, 36, 6, 54, 8, 28, 54, 75, 75, 16, 75, 131, 5, 25, 16, 69, 17, 69, 6, 96, 53, 96, 41, 119, 6, 6, 88, 50, 88, 52, 37, 0, 124, 73, 73, 7, 14, 36, 69, 79, 6, 114, 40, 79, 17, 77, 24, 44, 37, 69, 27, 37, 29, 33, 37, 50, 31, 69, 29, 101, 7, 61, 45, 17, 73, 37, 34, 18, 94, 22, 22, 63, 3, 25, 25, 17, 3, 90, 34, 34, 41, 34, 41, 54, 41, 54, 41, 41, 41, 163, 143, 96, 18, 32, 39, 86, 104, 11, 17, 17, 11, 86, 104, 78, 70, 52, 78, 17, 73, 91, 62, 7, 128, 50, 124, 18, 101, 46, 10, 75, 104, 73, 58, 132, 34, 13, 4, 95, 88, 33, 76, 74, 54, 62, 113, 114, 103, 32, 103, 69, 54, 53, 3, 11, 72, 31, 53, 102, 37, 53, 11, 81, 41, 10, 164, 10, 41, 31, 36, 113, 82, 3, 125, 62, 16, 4, 41, 41, 4, 128, 49, 138, 128, 74, 103, 0, 6, 101, 41, 142, 171, 39, 105, 121, 81, 62, 41, 81, 37, 3, 81, 69, 62, 3, 69, 70, 21, 29, 4, 91, 87, 37, 79, 36, 21, 71, 37, 41, 75, 128, 128, 15, 25, 3, 108, 73, 91, 62, 114,

62, 62, 36, 36, 15, 58, 114, 61, 114, 58, 105, 114, 41, 61, 176, 145, 46, 37, 30, 220, 77, 138, 15, 1, 128, 53, 50, 50, 58, 8, 91, 114, 105, 63, 91, 37, 37, 13, 169, 51, 102, 6, 102, 23, 105, 23, 58, 6, 29, 29, 19, 82, 29, 13, 36, 27, 29, 61, 12, 18, 127, 127, 12, 44, 102, 18, 4, 15, 206, 53, 127, 53, 17, 69, 69, 69, 29, 29, 109, 25, 102, 25, 53, 62, 99, 62, 62, 29, 62, 62, 45, 91, 125, 29, 29, 29, 4, 117, 72, 4, 30, 71, 71, 95, 79, 179, 71, 30, 53, 32, 32, 49, 25, 91, 25, 26, 26, 103, 123, 26, 41, 162, 78, 52, 103, 25, 6, 142, 94, 45, 45, 94, 127, 94, 94, 94, 47, 209, 138, 39, 39, 19, 154, 73, 67, 91, 27, 91, 84, 4, 84, 91, 12, 14, 165, 142, 54, 69, 192, 157, 185, 8, 95, 25, 62, 103, 103, 95, 71, 97, 62, 128, 0, 29, 51, 16, 94, 16, 16, 51, 0, 29, 85, 10, 105, 16, 29, 29, 13, 29, 4, 4, 132, 23, 95, 25, 54, 41, 29, 50, 70, 58, 142, 72, 70, 15, 72, 54, 29, 22, 145, 29, 127, 29, 85, 58, 101, 34, 165, 91, 46, 46, 25, 185, 25, 77, 128, 46, 128, 46, 188, 114, 46, 25, 45, 45, 114, 145, 114, 15, 102, 142, 8, 73, 31, 139, 157, 13, 79, 13, 114, 150, 8, 90, 91, 123, 69, 82, 132, 8, 18, 10, 102, 103, 114, 103, 8, 103, 13, 115, 55, 62, 3, 8, 154, 114, 99, 19, 8, 31, 73, 19, 99, 10, 6, 121, 32, 13, 32, 119, 32, 29, 145, 30, 13, 13, 114, 145, 32, 1, 123, 39, 29, 31, 69, 31, 140, 72, 72, 25, 25, 123, 25, 123, 8, 4, 85, 8, 25, 39, 25, 39, 85, 138, 25, 138, 25, 33, 102, 70, 25, 25, 31, 25, 25, 192, 69, 69, 114, 145, 120, 120, 8, 33, 98, 15, 212, 155, 8, 101, 8, 8, 98, 68, 155, 102, 132, 120, 30, 25, 123, 123, 101, 25, 123, 32, 24, 94, 145, 32, 24, 94, 118, 145, 101, 53, 53, 25, 128, 173, 142, 81, 81, 69, 33, 33, 125, 4, 1, 17, 27, 4, 17, 102, 27, 13, 25, 128, 71, 13, 39, 53, 13, 53, 47, 39, 23, 128, 53, 39, 47, 39, 135, 158, 136, 36, 36, 27, 157, 47, 76, 213, 47, 156, 25, 25, 53, 25, 53, 25, 86, 27, 159, 25, 62, 79, 39, 79, 25, 145, 49, 25, 143, 13, 114, 150, 130, 94, 102, 39, 4, 39, 61, 77, 228, 22, 25, 47, 119, 205, 122, 119, 205, 119, 22, 119, 258, 143, 22, 81, 179, 22, 22, 143, 25, 65, 53, 168, 36, 79, 175, 37, 79, 70, 79, 103, 70, 25, 175, 4, 96, 96, 49, 128, 138, 96, 22, 62, 47, 95, 105, 95, 62, 95, 62, 142, 103, 69, 103, 30, 103, 34, 173, 127, 70, 127, 132, 18, 85, 22, 71, 18, 206, 206, 18, 128, 145, 70, 193, 188, 8, 125, 114, 70, 128, 114, 145, 102, 25, 12, 108, 102, 94, 10, 102, 1, 102, 124, 22, 22, 118, 132, 22, 116, 75, 41, 63, 41, 189, 208, 55, 85, 69, 8, 71, 53, 71, 69, 102, 165, 41, 99, 69, 33, 33, 29, 156, 102, 13, 251, 102, 25, 13, 109, 102, 164, 102, 164, 102, 25, 29, 228, 29, 259, 179, 222, 95, 94, 30, 30, 30, 142, 55, 142, 72, 55, 102, 128, 17, 69, 164, 165, 3, 164, 36, 165, 27, 27, 45, 21, 21, 237, 113, 83, 231, 106, 13, 154, 13, 154, 128, 154, 148, 258, 25, 154, 128, 3, 27, 10, 145, 145, 21, 146, 25, 1, 185, 121, 0, 1, 95, 55, 95, 95, 30, 0, 27, 95, 0, 95, 8, 222, 27, 121, 30, 95, 121, 0, 98, 94, 131, 55, 95, 95, 30, 98, 30, 0, 91, 145, 66, 179, 66, 58, 175, 29, 0, 31, 173, 146, 160, 39, 53, 28, 123, 199, 123, 175, 146, 156, 54, 54, 149, 25, 70, 178, 128, 25, 70, 70, 94, 224, 54, 4, 54, 54, 25, 228, 160, 206, 165, 143, 206, 108, 220, 234, 160, 13, 169, 103, 103, 103, 91, 213, 222, 91, 103, 91, 103, 31, 30, 123, 13, 62, 103, 50, 106, 42, 13, 145, 114, 220, 65, 8, 8, 175, 11, 104, 94, 118, 132, 27, 118, 193, 27, 128, 127, 127, 183, 33, 30, 29, 103, 128, 61, 234, 165, 41, 29, 193, 33, 207, 41, 165, 165, 55, 81, 157, 157, 8, 81, 11, 27, 8, 8, 98, 96, 142, 145, 41, 179, 112, 62, 180, 206, 206, 165, 39, 241, 45, 151, 26, 197, 102, 192, 125, 128, 67, 128, 69, 128, 197, 33, 125, 102, 13, 103, 25, 30, 12, 30, 12, 30, 25, 77, 12, 25, 180, 27, 10, 69, 235, 228, 343, 118, 69, 41, 8, 69, 175, 25, 69, 25, 125, 41, 25, 41, 8, 155, 146, 155, 146, 155, 206, 168, 128, 157, 27, 273, 211, 211, 168, 11, 173, 154, 77, 173, 77, 102, 102, 102, 8, 85, 95, 102, 157, 28, 122, 234, 122, 157, 235, 222, 241, 10, 91, 179, 25, 13, 25, 41, 25, 206, 41, 6, 41, 158, 206, 206, 33, 296, 296, 33, 228, 69, 8, 114, 148, 33, 29, 66, 27, 27, 30, 233, 54, 173, 108, 106, 108, 108, 53, 103, 33, 33, 33, 176, 27, 27, 205, 164, 105, 237, 41, 27, 72, 165, 29, 29, 259, 132, 132, 132, 364, 71, 71, 27, 94, 160, 127, 51, 234, 55, 27, 95, 94, 165, 55, 55, 41, 0, 41, 128, 4, 123, 173, 6, 164, 157, 121, 121, 154, 86, 164, 164, 25, 93, 164, 25, 164, 210, 284, 62, 93, 30, 25, 25, 30, 30, 260, 130, 25, 125, 57, 53, 166, 166, 166, 185, 166, 158, 94, 113, 215, 159, 62, 99, 21, 172, 99, 184, 62, 259, 4, 21, 21, 77, 62, 173, 41, 146, 6, 41, 128, 121, 41, 11, 121, 103, 159, 164, 175, 206, 91, 103, 164, 72, 25, 129, 72, 206, 129, 33, 103, 102, 102, 29, 13, 11, 251, 234, 135, 31, 8, 123, 65, 91, 121, 129, 65, 243, 10, 91, 8, 65, 70, 228, 220, 243, 91, 10, 10, 30, 178, 91, 178, 33, 21, 25, 235, 165, 11, 161, 158, 27, 27, 30, 128, 75, 36, 30, 36, 36, 173, 25, 33, 178, 112, 162, 112, 112, 112, 162, 33, 33, 178, 123, 123, 39, 106, 91, 106, 106, 158, 106, 106, 284, 39, 230, 21, 228, 11, 21, 228, 159, 241, 62, 10, 62, 10, 68, 234, 39, 39, 138, 62, 22, 27, 183, 22, 215, 10, 175, 175, 353, 228, 42, 193, 175, 175, 27, 98, 27, 193, 150, 27, 173, 17, 233, 233, 25, 102, 123, 152, 242, 108, 4, 94, 176, 13, 41, 219, 17, 151, 22, 103, 103, 53, 128, 233, 284, 25, 265, 128, 39, 39, 138, 42, 39, 21, 86, 95, 127, 29, 91, 46, 103, 103, 215, 25, 123, 123, 230, 25, 193, 180, 30, 60, 30, 242, 136, 180, 193, 30, 206, 180, 60, 165, 206, 193, 165, 123, 164, 103, 68, 25, 70, 91, 25, 82, 53, 82, 186, 53, 82, 53, 25, 30, 282, 91, 13, 234, 160, 160, 126, 149, 36, 36, 160, 149, 178, 160, 39, 294, 149, 149, 160, 39, 95, 221, 186, 106, 178, 316, 267, 53, 53, 164, 159, 164, 165, 94, 228, 53, 52, 178, 183, 53, 294, 128, 55, 140, 294, 25, 95, 366, 15, 304, 13, 183, 77, 230, 6, 136, 235, 121, 311, 273, 36, 158, 235, 230, 98, 201, 165, 165, 165, 91, 175, 248, 39, 185, 128, 39, 39, 128, 313, 91, 36, 219, 130, 25, 130, 234, 234, 130, 234, 121, 205, 304, 94, 77, 64, 259, 60, 60, 60, 77, 242, 60, 145, 95, 270, 18, 91, 199, 159, 91, 235, 58, 249, 26, 123, 114, 29, 15, 191, 15, 30, 55, 55, 347, 4, 29, 15, 4, 341, 93, 7, 30, 23, 7, 121, 266, 178, 261, 70, 169, 25, 25, 158, 169, 25, 169, 270, 270, 13, 128, 327, 103, 55, 128, 103, 136, 159, 103, 327, 41, 32, 111, 111, 114, 173, 215, 173, 25, 173, 180, 114, 173, 173, 98, 93, 25, 160, 157, 159, 160, 159, 159, 160, 320, 35, 193, 221, 33, 36, 136, 248, 91, 215, 125, 215, 156, 68, 125, 125, 1, 287, 123, 94, 30, 184, 13, 30, 94, 123, 206, 12, 206, 289, 128, 122, 184, 128, 289, 178, 29, 26, 206, 178, 65, 206, 128, 192, 102, 197, 36, 94, 94, 155, 10, 36, 121, 280, 121, 368, 192, 121, 121, 179, 121, 36, 54, 192, 121, 192, 197, 118, 123, 224, 118, 10, 192, 10, 91, 269, 91, 49, 206, 184, 185, 62, 8, 49, 289, 30, 5, 55, 30, 42, 39, 220, 298, 42, 347, 42, 234, 42, 70, 42, 55, 321, 129, 172, 173, 172, 13, 98, 129, 325, 235, 284, 362, 129, 233, 345, 175, 261, 175, 60, 261, 58, 289, 99, 99, 99, 206, 99, 36, 175, 29, 25, 432, 125, 264, 168, 173, 69, 158, 273, 179, 164, 69, 158, 69, 8, 95, 192, 30, 164, 101, 44, 53, 273, 335, 273, 53, 45, 128, 45, 234, 123, 105, 103, 103, 224, 36, 90, 211, 282, 264, 91, 228, 91, 166, 264, 228, 398, 50, 101, 91, 264, 73, 36, 25, 73, 50, 50, 242, 36, 36, 58, 165, 204, 353, 165, 125, 320, 128, 298, 298, 180, 128, 60, 102, 30, 30, 53, 179, 234, 325, 234, 175, 21, 250, 215, 103, 21, 21, 250, 91, 211, 91, 313, 301, 323, 215, 228, 160, 29, 29, 81, 53, 180, 146, 248, 66, 159, 39, 98, 323, 98, 36, 95, 218, 234, 39, 82, 82, 230, 62, 13, 62, 230, 13, 30, 98, 0, 8, 98, 8, 98, 91, 267, 121, 197, 30, 78, 27, 78, 102, 27, 298, 160, 103, 264, 264, 264, 175, 17, 273, 273, 165, 31, 160, 17, 99, 17, 99, 234, 31, 17, 99, 36, 26, 128, 29, 214, 353, 264, 102, 36, 102, 264, 264, 273, 273, 4, 16, 138, 138, 264, 128,

313, 25, 420, 60, 10, 280, 264, 60, 60, 103, 178, 125, 178, 29, 327, 29, 36, 30, 36, 4, 52, 183, 183, 173, 52, 31, 173, 31, 158, 31, 158, 31, 9, 31, 31, 353, 31, 353, 173, 415, 9, 17, 222, 31, 103, 31, 165, 27, 31, 31, 165, 27, 27, 206, 31, 31, 4, 4, 30, 4, 4, 264, 185, 159, 310, 273, 310, 173, 40, 4, 173, 4, 173, 4, 250, 250, 62, 188, 119, 250, 233, 62, 121, 105, 105, 54, 103, 111, 291, 236, 236, 103, 297, 36, 26, 316, 69, 183, 158, 206, 129, 160, 129, 184, 55, 179, 279, 11, 179, 347, 160, 184, 129, 179, 351, 179, 353, 179, 129, 129, 351, 11, 111, 93, 93, 235, 103, 173, 53, 93, 50, 111, 86, 123, 94, 36, 183, 60, 55, 55, 178, 219, 253, 321, 178, 235, 235, 183, 183, 204, 321, 219, 160, 193, 335, 121, 70, 69, 295, 159, 297, 231, 121, 231, 136, 353, 136, 121, 279, 215, 366, 215, 353, 159, 353, 353, 103, 31, 31, 298, 298, 30, 30, 165, 273, 25, 219, 35, 165, 259, 54, 36, 54, 54, 165, 71, 250, 327, 13, 289, 165, 196, 165, 165, 94, 233, 165, 94, 60, 165, 96, 220, 166, 271, 158, 397, 122, 53, 53, 137, 280, 272, 62, 30, 30, 105, 102, 67, 140, 8, 67, 21, 270, 298, 69, 173, 298, 91, 179, 327, 86, 179, 88, 179, 179, 55, 123, 220, 233, 94, 94, 175, 13, 53, 13, 154, 191, 74, 83, 83, 325, 207, 83, 74, 83, 325, 74, 316, 388, 55, 55, 364, 55, 183, 434, 273, 273, 273, 164, 213, 11, 213, 327, 321, 21, 352, 185, 103, 13, 13, 55, 30, 323, 123, 178, 435, 178, 30, 175, 175, 30, 481, 527, 175, 125, 232, 306, 232, 206, 306, 364, 206, 270, 206, 232, 10, 30, 130, 160, 130, 347, 240, 30, 136, 130, 347, 136, 279, 298, 206, 30, 103, 273, 241, 70, 206, 306, 434, 206, 94, 94, 156, 161, 321, 321, 64, 161, 13, 183, 183, 83, 161, 13, 169, 13, 159, 36, 173, 159, 36, 36, 230, 235, 235, 159, 159, 335, 312, 42, 342, 264, 39, 39, 39, 34, 298, 36, 36, 252, 164, 29, 493, 29, 387, 387, 435, 493, 132, 273, 105, 132, 74, 73, 206, 234, 273, 206, 95, 15, 280, 280, 280, 280, 397, 273, 273, 242, 397, 280, 397, 397, 397, 273, 397, 280, 230, 137, 353, 67, 81, 137, 137, 353, 259, 312, 114, 164, 164, 25, 77, 21, 77, 165, 30, 30, 231, 234, 121, 234, 312, 121, 364, 136, 123, 123, 136, 123, 136, 150, 264, 285, 30, 166, 93, 30, 39, 224, 136, 39, 355, 355, 397, 67, 67, 25, 67, 25, 298, 11, 67, 264, 374, 99, 150, 321, 67, 70, 67, 295, 150, 29, 321, 150, 70, 29, 142, 355, 311, 173, 13, 253, 103, 114, 114, 70, 192, 22, 128, 128, 183, 184, 70, 77, 215, 102, 292, 30, 123, 279, 292, 142, 33, 215, 102, 468, 123, 468, 473, 30, 292, 215, 30, 213, 443, 473, 215, 234, 279, 279, 279, 279, 265, 443, 206, 66, 313, 34, 30, 206, 30, 51, 15, 206, 41, 434, 41, 398, 67, 30, 301, 67, 36, 3, 285, 437, 136, 136, 22, 136, 145, 365, 323, 323, 145, 136, 22, 453, 99, 323, 353, 9, 258, 323, 231, 128, 231, 382, 150, 420, 39, 94, 29, 29, 353, 22, 22, 347, 353, 39, 29, 22, 183, 8, 284, 355, 388, 284, 60, 64, 99, 60, 64, 150, 95, 150, 364, 150, 95, 150, 6, 236, 383, 544, 81, 206, 388, 206, 58, 159, 99, 231, 228, 363, 363, 121, 99, 121, 121, 99, 422, 544, 273, 173, 121, 427, 102, 121, 235, 284, 179, 25, 197, 25, 179, 511, 70, 368, 70, 25, 388, 123, 368, 159, 213, 410, 159, 236, 127, 159, 21, 373, 184, 424, 327, 250, 176, 176, 175, 284, 316, 176, 284, 327, 111, 250, 284, 175, 175, 264, 111, 176, 219, 111, 427, 427, 176, 284, 427, 353, 428, 55, 184, 493, 158, 136, 99, 287, 264, 334, 264, 213, 213, 292, 481, 93, 264, 292, 295, 295, 6, 367, 279, 173, 308, 285, 158, 308, 335, 299, 137, 137, 572, 41, 137, 137, 41, 94, 335, 220, 36, 224, 420, 36, 265, 265, 91, 91, 71, 123, 264, 91, 91, 123, 107, 30, 22, 292, 35, 241, 356, 298, 14, 298, 441, 35, 121, 71, 63, 130, 63, 488, 363, 71, 63, 307, 194, 71, 71, 220, 121, 125, 71, 220, 71, 71, 71, 71, 235, 265, 353, 128, 155, 128, 420, 400, 130, 173, 183, 183, 184, 130, 173, 183, 13, 183, 130, 130, 183, 183, 353, 353, 183, 242, 183, 183, 306, 324, 324, 321, 306, 321, 6, 6, 128, 306, 242, 242, 306, 183, 183, 6, 183, 321, 486, 183, 164, 30, 78, 138, 158, 138, 34, 206, 362, 55, 70, 67, 21, 375, 136, 298, 81, 298, 298, 298, 230, 121, 30, 230, 311, 240, 311, 311, 158, 204, 136, 136, 184, 136, 264, 311, 311, 312, 312, 72, 311, 175, 264, 91, 175, 264, 121, 461, 312, 312, 238, 475, 350, 512, 350, 312, 313, 350, 312, 366, 294, 30, 253, 253, 253, 388, 158, 388, 22, 388, 22, 388, 103, 321, 321, 253, 7, 437, 103, 114, 242, 114, 114, 242, 114, 114, 242, 242, 242, 306, 242, 114, 7, 353, 335, 27, 241, 299, 312, 364, 506, 409, 94, 462, 230, 462, 243, 230, 175, 175, 462, 461, 230, 428, 426, 175, 175, 165, 175, 175, 372, 183, 572, 102, 85, 102, 538, 206, 376, 85, 85, 284, 85, 85, 284, 398, 83, 160, 265, 308, 398, 310, 583, 289, 279, 273, 285, 490, 490, 211, 292, 292, 158, 398, 30, 220, 169, 368, 368, 368, 169, 159, 368, 93, 368, 368, 93, 169, 368, 368, 443, 368, 298, 443, 368, 298, 538, 345, 345, 311, 178, 54, 311, 215, 178, 175, 222, 264, 475, 264, 264, 475, 478, 289, 63, 236, 63, 299, 231, 296, 397, 299, 158, 36, 164, 164, 21, 492, 21, 164, 21, 164, 403, 26, 26, 588, 179, 234, 169, 465, 295, 67, 41, 353, 295, 538, 161, 185, 306, 323, 68, 420, 323, 82, 241, 241, 36, 53, 493, 301, 292, 241, 250, 63, 63, 103, 442, 353, 185, 353, 321, 353, 185, 353, 353, 185, 409, 353, 589, 34, 271, 271, 34, 86, 34, 34, 353, 353, 39, 414, 4, 95, 95, 4, 225, 95, 4, 121, 30, 552, 136, 159, 159, 514, 159, 159, 54, 514, 206, 136, 206, 159, 74, 235, 235, 312, 54, 312, 42, 156, 422, 629, 54, 465, 265, 165, 250, 35, 165, 175, 659, 175, 175, 8, 8, 8, 8, 206, 206, 206, 50, 435, 206, 432, 230, 230, 234, 230, 94, 299, 299, 285, 184, 41, 93, 299, 299, 285, 41, 285, 158, 285, 206, 299, 41, 36, 396, 364, 364, 120, 396, 514, 91, 382, 538, 807, 717, 22, 93, 412, 54, 215, 54, 298, 308, 148, 298, 148, 298, 308, 102, 656, 6, 148, 745, 128, 298, 64, 407, 273, 41, 172, 64, 234, 250, 398, 181, 445, 95, 236, 441, 477, 504, 102, 196, 137, 364, 60, 453, 137, 364, 367, 334, 364, 299, 196, 397, 630, 589, 589, 196, 646, 337, 235, 128, 128, 343, 289, 235, 324, 427, 324, 58, 215, 215, 461, 425, 461, 387, 440, 285, 440, 440, 285, 387, 632, 325, 325, 440, 461, 425, 425, 387, 627, 191, 285, 440, 308, 55, 219, 280, 308, 265, 538, 183, 121, 30, 236, 206, 30, 455, 236, 30, 30, 705, 83, 228, 280, 468, 132, 8, 132, 132, 128, 409, 173, 353, 132, 409, 35, 128, 450, 137, 398, 67, 432, 423, 235, 235, 388, 306, 93, 93, 452, 300, 190, 13, 452, 388, 30, 452, 13, 30, 13, 30, 306, 362, 234, 721, 635, 809, 784, 67, 498, 498, 67, 353, 635, 67, 183, 159, 445, 285, 183, 53, 183, 445, 265, 432, 57, 420, 432, 420, 477, 327, 55, 60, 105, 183, 218, 104, 104, 475, 239, 582, 151, 239, 104, 732, 41, 26, 784, 86, 300, 215, 36, 64, 86, 86, 675, 294, 64, 86, 528, 550, 493, 565, 298, 230, 312, 295, 538, 298, 295, 230, 54, 374, 516, 441, 54, 54, 323, 401, 401, 382, 159, 837, 159, 54, 401, 592, 159, 401, 417, 610, 264, 150, 323, 452, 185, 323, 323, 185, 403, 185, 423, 165, 425, 219, 407, 270, 231, 99, 93, 231, 631, 756, 71, 364, 434, 213, 86, 102, 434, 102, 86, 23, 71, 335, 164, 323, 409, 381, 4, 124, 41, 424, 206, 41, 124, 41, 41, 703, 635, 124, 493, 41, 41, 487, 492, 124, 175, 124, 261, 600, 488, 261, 488, 261, 206, 677, 261, 308, 723, 908, 704, 691, 723, 488, 488, 441, 136, 476, 312, 136, 550, 572, 728, 550, 22, 312, 312, 22, 55, 413, 183, 280, 593, 191, 36, 36, 427, 36, 695, 592, 19, 544, 13, 468, 13, 544, 72, 437, 321, 266, 461, 266, 441, 230, 409, 93, 521, 521, 345, 235, 22, 142, 150, 102, 569, 235, 264, 91, 521, 264, 7, 102, 7, 498, 521, 235, 537, 235, 6, 241, 420, 420, 631, 41, 527, 103, 67, 337, 62, 264, 527, 131, 67, 174, 263, 264, 36, 36, 263, 581, 253, 465, 160, 286, 91, 160, 55, 4, 4, 631, 631, 608, 365, 465, 294, 427, 427, 335, 669, 669, 129, 93, 93,



757, 73, 818, 23, 73, 563, 952, 262, 563, 137, 262, 1022, 952, 137, 1273, 442, 952, 604, 137, 308, 384, 913, 235, 325, 695, 398, 95, 668, 776, 713, 309, 691, 22, 10, 364, 682, 682, 578, 481, 1252, 1072, 1252, 825, 578, 825, 1072, 1149, 592, 273, 387, 273, 427, 155, 1204, 50, 452, 50, 1142, 50, 367, 452, 1142, 611, 367, 50, 50, 367, 50, 1675, 99, 367, 50, 1501, 1099, 830, 681, 689, 917, 1089, 453, 425, 235, 918, 538, 550, 335, 161, 387, 859, 324, 21, 838, 859, 1123, 21, 723, 21, 335, 335, 206, 21, 364, 1426, 21, 838, 838, 335, 364, 21, 21, 859, 920, 838, 838, 397, 81, 639, 397, 397, 588, 933, 933, 784, 222, 830, 36, 36, 222, 1251, 266, 36, 146, 266, 366, 581, 605, 366, 22, 966, 681, 681, 433, 730, 1013, 550, 21, 21, 938, 488, 516, 21, 21, 656, 420, 323, 323, 323, 327, 323, 918, 581, 581, 830, 361, 830, 364, 259, 364, 496, 496, 364, 691, 705, 691, 475, 427, 1145, 600, 179, 427, 527, 749, 869, 689, 335, 347, 220, 298, 689, 1426, 183, 554, 55, 832, 550, 550, 550, 165, 770, 957, 67, 1386, 219, 683, 683, 355, 683, 355, 738, 355, 842, 931, 266, 325, 349, 256, 1113, 256, 423, 960, 554, 554, 325, 554, 508, 22, 142, 22, 508, 916, 767, 55, 1529, 767, 55, 1286, 93, 972, 550, 931, 1286, 1286, 972, 93, 1286, 1392, 890, 93, 1286, 93, 1286, 972, 374, 931, 890, 808, 779, 975, 975, 175, 173, 4, 681, 383, 1367, 173, 383, 1367, 383, 173, 175, 69, 238, 146, 238, 36, 148, 888, 238, 173, 238, 148, 238, 888, 185, 925, 925, 797, 925, 815, 925, 469, 784, 289, 784, 925, 797, 925, 925, 1093, 925, 925, 1163, 797, 797, 815, 925, 1093, 784, 636, 663, 925, 187, 922, 316, 1380, 709, 916, 916, 187, 355, 948, 916, 187, 916, 916, 948, 948, 916, 355, 316, 316, 334, 300, 1461, 36, 583, 1179, 699, 235, 858, 583, 699, 858, 699, 1189, 1256, 1189, 699, 797, 699, 699, 699, 699, 427, 488, 427, 488, 175, 815, 656, 656, 150, 322, 465, 322, 870, 465, 1099, 582, 665, 767, 749, 635, 749, 600, 1448, 36, 502, 235, 502, 355, 502, 355, 355, 355, 172, 355, 355, 95, 866, 425, 393, 1165, 42, 42, 42, 393, 939, 909, 909, 836, 552, 424, 1333, 852, 897, 1426, 1333, 1446, 1426, 997, 1011, 852, 1198, 55, 32, 239, 588, 681, 681, 239, 1401, 32, 588, 239, 462, 286, 1260, 984, 1160, 960, 960, 486, 828, 462, 960, 1199, 581, 850, 663, 581, 751, 581, 581, 1571, 252, 252, 1283, 264, 430, 264, 430, 430, 842, 252, 745, 21, 307, 681, 1592, 488, 857, 857, 1161, 857, 857, 857, 138, 374, 374, 1196, 374, 1903, 1782, 1626, 414, 112, 1477, 1040, 356, 775, 414, 414, 112, 356, 775, 435, 338, 1066, 689, 689, 1501, 689, 1249, 205, 689, 765, 220, 308, 917, 308, 308, 220, 327, 387, 838, 917, 917, 917, 220, 662, 308, 220, 387, 387, 220, 220, 308, 308, 308, 387, 1009, 1745, 822, 279, 554, 1129, 543, 383, 870, 1425, 241, 870, 241, 383, 716, 592, 21, 21, 592, 425, 550, 550, 550, 427, 230, 57, 483, 784, 860, 57, 308, 57, 486, 870, 447, 486, 433, 433, 870, 433, 997, 486, 443, 433, 433, 997, 486, 1292, 47, 708, 81, 895, 394, 81, 935, 81, 81, 81, 374, 986, 916, 1103, 1095, 465, 495, 916, 667, 1745, 518, 220, 1338, 220, 734, 1294, 741, 166, 828, 741, 741, 1165, 1371, 1371, 471, 1371, 647, 1142, 1878, 1878, 1371, 1371, 822, 66, 327, 158, 427, 427, 465, 465, 676, 676, 30, 30, 676, 676, 893, 1592, 93, 455, 308, 582, 695, 582, 629, 582, 85, 1179, 85, 85, 1592, 1179, 280, 1027, 681, 398, 1027, 398, 295, 784, 740, 509, 425, 968, 509, 46, 833, 842, 401, 184, 401, 464, 6, 1501, 1501, 550, 538, 883, 538, 883, 883, 883, 1129, 550, 550, 333, 689, 948, 21, 21, 241, 2557, 2094, 273, 308, 58, 863, 893, 1086, 409, 136, 1086, 592, 592, 830, 830, 883, 830, 277, 68, 689, 902, 277, 453, 507, 129, 689, 630, 664, 550, 128, 1626, 1626, 128, 902, 312, 589, 755, 755, 589, 755, 407, 1782, 589, 784, 1516, 1118, 407, 407, 1447, 589, 235, 755, 1191, 235, 235, 407, 128, 589, 1118, 21, 383, 1331, 691, 481, 383, 1129, 1129, 1261, 1104, 1378, 1129, 784, 1129, 1261, 1129, 947, 1129, 784, 784, 1129, 1129, 35, 1104, 35, 866, 1129, 1129, 64, 481, 730, 1260, 481, 970, 481, 481, 481, 481, 863, 481, 681, 699, 863, 486, 681, 481, 481, 55, 55, 235, 1364, 944, 632, 822, 401, 822, 952, 822, 822, 99, 550, 2240, 550, 70, 891, 860, 860, 550, 550, 916, 1176, 1530, 425, 1530, 916, 628, 1583, 916, 628, 916, 628, 916, 628, 425, 916, 1062, 1265, 916, 916, 916, 280, 461, 916, 916, 1583, 628, 1062, 916, 916, 677, 1297, 924, 1260, 83, 1260, 482, 433, 234, 462, 323, 1656, 997, 323, 323, 931, 838, 931, 1933, 1391, 367, 323, 931, 1391, 1391, 103, 1116, 1116, 1116, 769, 1195, 1218, 312, 791, 312, 741, 791, 997, 312, 334, 334, 312, 287, 287, 633, 1397, 1426, 605, 1431, 327, 592, 705, 1194, 592, 1097, 1118, 1503, 1267, 1267, 1267, 618, 1229, 734, 1089, 785, 1089, 1129, 1148, 1148, 1089, 915, 1148, 1129, 1148, 1011, 1011, 1229, 871, 1560, 1560, 1560, 563, 1537, 1009, 1560, 632, 985, 592, 1308, 592, 882, 145, 145, 397, 837, 383, 592, 592, 832, 36, 2714, 2107, 1588, 1347, 36, 36, 1443, 1453, 334, 2230, 1588, 1169, 650, 1169, 2107, 425, 425, 891, 891, 425, 2532, 679, 274, 274, 274, 325, 274, 1297, 194, 1297, 627, 314, 917, 314, 314, 1501, 414, 1490, 1036, 592, 1036, 1025, 901, 1218, 1025, 901, 280, 592, 592, 901, 1461, 159, 159, 159, 2076, 1066, 1176, 1176, 516, 327, 516, 1179, 1176, 899, 1176, 1176, 323, 1187, 1229, 663, 1229, 504, 1229, 916, 1229, 916, 1661, 41, 36, 278, 1027, 648, 648, 648, 1626, 648, 646, 1179, 1580, 1061, 1514, 1008, 1741, 2076, 1514, 1008, 952, 1089, 427, 952, 427, 1083, 425, 427, 1089, 1083, 425, 427, 425, 230, 920, 1678, 920, 1678, 189, 189, 953, 189, 133, 189, 1075, 189, 189, 133, 1264, 725, 189, 1629, 189, 808, 230, 230, 2179, 770, 230, 770, 230, 21, 21, 784, 1118, 230, 230, 230, 770, 1118, 986, 808, 916, 30, 327, 918, 679, 414, 916, 1165, 1355, 916, 755, 733, 433, 1490, 433, 433, 433, 605, 433, 433, 433, 1446, 679, 206, 433, 21, 2452, 206, 206, 433, 1894, 206, 822, 206, 2073, 206, 206, 21, 822, 21, 206, 206, 21, 383, 1513, 375, 1347, 432, 1589, 172, 954, 242, 1256, 1256, 1248, 1256, 1256, 1248, 1248, 1256, 842, 13, 592, 13, 842, 1291, 592, 21, 175, 13, 592, 13, 13, 1426, 13, 1541, 445, 808, 808, 863, 647, 219, 1592, 1029, 1225, 917, 1963, 1129, 555, 1313, 550, 660, 550, 220, 660, 552, 663, 220, 533, 220, 383, 550, 1278, 1495, 636, 842, 1036, 425, 842, 425, 1537, 1278, 842, 554, 1508, 636, 554, 301, 842, 792, 1392, 1021, 284, 1172, 997, 1021, 103, 1316, 308, 1210, 848, 848, 1089, 1089, 848, 848, 67, 1029, 827, 1029, 2078, 827, 1312, 1029, 827, 590, 872, 1312, 427, 67, 67, 67, 67, 872, 827, 872, 2126, 1436, 26, 2126, 67, 1072, 2126, 1610, 872, 1620, 883, 883, 1397, 1189, 555, 555, 563, 1189, 555, 640, 555, 640, 1089, 1089, 610, 610, 1585, 610, 1355, 610, 1015, 616, 925, 1015, 482, 230, 707, 231, 888, 1355, 589, 1379, 151, 931, 1486, 1486, 393, 235, 960, 590, 235, 960, 422, 142, 285, 285, 327, 327, 442, 2009, 822, 445, 822, 567, 888, 2611, 1537, 323, 55, 1537, 323, 888, 2611, 323, 1537, 323, 58, 445, 593, 2045, 593, 58, 47, 770, 842, 47, 47, 842, 842, 648, 2557, 173, 689, 2291, 1446, 2085, 2557, 2557, 2291, 1780, 1535, 2291, 2391, 808, 691, 1295, 1165, 983, 948, 2000, 948, 983, 983, 2225, 2000, 983, 983, 705, 948, 2000, 1795, 1592, 478, 592, 1795, 1795, 663, 478, 1790, 478, 592, 1592, 173, 901, 312, 4, 1606, 173, 838, 754, 754, 128, 550, 1166, 551, 1480, 550, 550, 1875, 1957, 1166, 902, 1875, 550, 550, 551, 2632, 551, 1875, 1875, 551, 2891, 2159, 2632, 3231, 551, 815, 150, 1654, 1059, 1059, 734, 770, 555, 1592, 555, 2059, 770, 770, 1803, 627, 627,

627, 2059, 931, 1272, 427, 1606, 1272, 1606, 1187, 1204, 397, 822, 21, 1645, 263, 263, 822, 263, 1645, 280, 263, 605, 1645, 2014, 21, 21, 1029, 263, 1916, 2291, 397, 397, 496, 270, 270, 1319, 264, 1638, 264, 986, 1278, 1397, 1278, 1191, 409, 1191, 740, 1191, 754, 754, 387, 63, 948, 666, 666, 1198, 548, 63, 1248, 285, 1248, 169, 1248, 1248, 285, 918, 224, 285, 1426, 1671, 514, 514, 717, 514, 51, 1521, 1745, 51, 605, 1191, 51, 128, 1191, 51, 51, 1521, 267, 513, 952, 966, 1671, 897, 51, 71, 592, 986, 986, 1121, 592, 280, 2000, 2000, 1165, 1165, 1165, 1818, 222, 1818, 1165, 1252, 506, 327, 443, 432, 1291, 1291, 2755, 1413, 520, 1318, 227, 1047, 828, 520, 347, 1364, 136, 136, 452, 457, 457, 132, 457, 488, 1087, 1013, 2225, 32, 1571, 2009, 483, 67, 483, 740, 740, 1013, 2854, 866, 32, 2861, 866, 887, 32, 2444, 740, 32, 32, 866, 2225, 866, 32, 1571, 2627, 32, 850, 1675, 569, 1158, 32, 1158, 1797, 2641, 1565, 1158, 569, 1797, 1158, 1797, 55, 1703, 42, 55, 2562, 675, 1703, 42, 55, 749, 488, 488, 347, 1206, 1286, 1286, 488, 488, 1206, 1286, 1206, 1286, 550, 550, 1790, 860, 550, 2452, 550, 550, 2765, 1089, 1633, 797, 2244, 1313, 194, 2129, 194, 194, 194, 818, 32, 194, 450, 1313, 2387, 194, 1227, 2387, 308, 2232, 526, 476, 278, 830, 830, 194, 830, 194, 278, 194, 714, 476, 830, 714, 830, 278, 830, 2532, 1218, 1759, 1446, 960, 1747, 187, 1446, 1759, 960, 105, 1446, 1446, 1271, 1446, 960, 960, 1218, 1446, 1446, 105, 1446, 960, 488, 1446, 427, 534, 842, 1969, 2460, 1969, 842, 842, 1969, 427, 941, 2160, 427, 230, 938, 2075, 1675, 1675, 895, 1675, 34, 129, 1811, 239, 749, 1957, 2271, 749, 1908, 129, 239, 239, 129, 129, 2271, 2426, 1355, 1756, 194, 1583, 194, 194, 1583, 194, 1355, 194, 1628, 2221, 1269, 2425, 1756, 1355, 1355, 1583, 1033, 427, 582, 30, 582, 582, 935, 1444, 1962, 915, 733, 915, 938, 1962, 767, 353, 1630, 1962, 1962, 563, 733, 563, 733, 353, 822, 1630, 740, 2076, 2076, 2076, 589, 589, 2636, 866, 589, 947, 1528, 125, 273, 1058, 1058, 1161, 1635, 1355, 1161, 1161, 1355, 1355, 650, 1206, 1206, 784, 784, 784, 784, 412, 461, 412, 2240, 412, 679, 891, 461, 679, 679, 189, 189, 1933, 1651, 2515, 189, 1386, 538, 1386, 1386, 1187, 1386, 2423, 2601, 2285, 175, 175, 2331, 194, 3079, 384, 538, 2365, 2294, 538, 2166, 1841, 3326, 1256, 3923, 976, 85, 550, 550, 1295, 863, 863, 550, 1249, 550, 1759, 146, 1069, 920, 2633, 885, 885, 1514, 1489, 166, 1514, 2041, 885, 2456, 885, 2041, 1081, 1948, 362, 550, 94, 324, 2308, 94, 2386, 94, 550, 874, 1329, 1759, 2280, 1487, 493, 493, 2099, 2599, 1431, 1086, 1514, 1086, 2099, 1858, 368, 1330, 2599, 1858, 2846, 2846, 2907, 2846, 713, 713, 1854, 1123, 713, 713, 3010, 1123, 3010, 538, 713, 1123, 447, 822, 555, 2011, 493, 508, 2292, 555, 1736, 2135, 2704, 555, 2814, 555, 2000, 555, 555, 822, 914, 327, 679, 327, 648, 537, 2263, 931, 1496, 537, 1296, 1745, 1592, 1658, 1795, 650, 1592, 1745, 1745, 1658, 1592, 1745, 1592, 1745, 1658, 1338, 2124, 1592, 1745, 1745, 1745, 837, 1726, 2897, 1118, 1118, 230, 1118, 1118, 1118, 1388, 1748, 514, 128, 1165, 931, 514, 2974, 2041, 2387, 2041, 979, 185, 36, 1269, 550, 173, 812, 36, 1165, 2676, 2562, 1473, 2885, 1982, 1578, 1578, 383, 383, 2360, 383, 1578, 2360, 1584, 1982, 1578, 1578, 1578, 2019, 1036, 355, 724, 2023, 205, 303, 355, 1036, 1966, 355, 1036, 401, 401, 401, 830, 401, 849, 578, 401, 849, 849, 578, 1776, 1123, 552, 2632, 808, 1446, 1120, 373, 1529, 1483, 1057, 893, 1284, 1430, 1529, 1529, 2632, 1352, 2063, 1606, 1352, 1606, 2291, 3079, 2291, 1529, 506, 838, 1606, 1606, 1352, 1529, 1529, 1483, 1529, 1606, 1529, 259, 902, 259, 902, 612, 612, 284, 398, 2991, 1534, 1118, 1118, 1118, 1118, 734, 284, 2224, 398, 734, 284, 734, 398, 3031, 398, 734, 1707, 2643, 1344, 1477, 475, 1818, 194, 1894, 691, 1528, 1184, 1207, 1501, 6, 2069, 871, 2069, 3548, 1443, 2069, 2685, 3265, 1350, 3265, 2069, 2069, 128, 1313, 128, 663, 414, 1313, 414, 2000, 128, 2000, 663, 1313, 699, 1797, 550, 327, 550, 1526, 699, 327, 1797, 1526, 550, 550, 327, 550, 1426, 1426, 2285, 1123, 890, 728, 1707, 728, 728, 327, 253, 1187, 1281, 1364, 1571, 2170, 755, 3232, 925, 1496, 2170, 2170, 1125, 443, 902, 902, 925, 755, 2078, 2457, 902, 2059, 2170, 1643, 1129, 902, 902, 1643, 1129, 606, 36, 103, 338, 338, 1089, 338, 338, 338, 1089, 338, 36, 340, 1206, 1176, 2041, 833, 1854, 1916, 1916, 1501, 2132, 1736, 3065, 367, 1934, 833, 833, 833, 2041, 3017, 2147, 818, 1397, 828, 2147, 398, 828, 818, 1158, 818, 689, 327, 36, 1745, 2132, 582, 1475, 189, 582, 2132, 1191, 582, 2132, 1176, 1176, 516, 2610, 2230, 2230, 64, 1501, 537, 1501, 173, 2230, 2988, 1501, 2694, 2694, 537, 537, 173, 173, 1501, 537, 64, 173, 173, 64, 2230, 537, 2230, 537, 2230, 2230, 2069, 3142, 1645, 689, 1165, 1165, 1963, 514, 488, 1963, 1145, 235, 1145, 1078, 1145, 231, 2405, 552, 21, 57, 57, 57, 1297, 1455, 1988, 2310, 1885, 2854, 2014, 734, 1705, 734, 2854, 734, 677, 1988, 1660, 734, 677, 734, 677, 677, 734, 2854, 1355, 677, 1397, 2947, 2386, 1698, 128, 1698, 3028, 2386, 2437, 2947, 2386, 2643, 2386, 2804, 1188, 335, 746, 1187, 1187, 861, 2519, 1917, 2842, 1917, 675, 1308, 234, 1917, 314, 314, 2339, 2339, 2592, 2576, 902, 916, 2339, 916, 2339, 916, 2339, 916, 1089, 1089, 2644, 1221, 1221, 2446, 308, 308, 2225, 2225, 3192, 2225, 555, 1592, 1592, 555, 893, 555, 550, 770, 3622, 2291, 2291, 3419, 465, 250, 2842, 2291, 2291, 2291, 935, 160, 1271, 308, 325, 935, 1799, 1799, 1891, 2227, 1799, 1598, 112, 1415, 1840, 2014, 1822, 2014, 677, 1822, 1415, 1415, 1822, 2014, 2386, 2159, 1822, 1415, 1822, 179, 1976, 1033, 179, 1840, 2014, 1415, 1970, 1970, 1501, 563, 563, 563, 462, 563, 1970, 1158, 563, 563, 1541, 1238, 383, 235, 1158, 383, 1278, 383, 1898, 2938, 21, 2938, 1313, 2201, 2059, 423, 2059, 1313, 872, 1313, 2044, 89, 173, 3327, 1660, 2044, 1623, 173, 1114, 1114, 1592, 1868, 1651, 1811, 383, 3469, 1811, 1651, 869, 383, 383, 1651, 1651, 3223, 2166, 3469, 767, 383, 1811, 767, 2323, 3355, 1457, 3341, 2640, 2976, 2323, 3341, 2323, 2640, 103, 103, 1161, 1080, 2429, 370, 2018, 2854, 2429, 2166, 2429, 2094, 2207, 871, 1963, 1963, 2023, 2023, 2336, 663, 2893, 1580, 691, 663, 705, 2046, 2599, 409, 2295, 1118, 2494, 1118, 1950, 549, 2494, 2453, 2046, 2494, 2453, 2046, 2453, 2046, 409, 1118, 4952, 2291, 2225, 1894, 1423, 2498, 567, 4129, 1475, 1501, 795, 463, 2084, 828, 828, 232, 828, 232, 232, 1818, 1818, 666, 463, 232, 220, 220, 2162, 2162, 833, 4336, 913, 35, 913, 21, 2927, 886, 3037, 383, 886, 876, 1747, 383, 916, 916, 916, 2927, 916, 1747, 837, 1894, 717, 423, 481, 1894, 1059, 2262, 3206, 4700, 1059, 3304, 2262, 871, 1831, 871, 3304, 1059, 1158, 1934, 1158, 756, 1511, 41, 978, 1934, 2603, 720, 41, 756, 41, 325, 2611, 1158, 173, 1123, 1934, 1934, 1511, 2045, 2045, 2045, 1423, 3206, 3691, 2512, 3206, 2512, 2000, 1811, 2504, 2504, 2611, 2437, 2437, 2437, 1455, 893, 150, 2665, 1966, 605, 398, 2331, 1177, 516, 1962, 4241, 94, 1252, 760, 1292, 1962, 1373, 2000, 1990, 3684, 42, 1868, 3779, 1811, 1811, 2041, 3010, 5436, 1780, 2041, 1868, 1811, 1780, 1811, 1868, 1811, 2041, 1868, 1811, 5627, 4274, 1811, 1868, 4602, 1811, 1811, 1474, 2665, 235, 1474, 2665

## 6. Considérations sur la sécurité

La livraison des données peut être sujette à des attaques de déni de service par des attaquants qui envoient des paquets corrompus acceptés comme légitimes par les receveurs. Ceci est un problème particulièrement pour la livraison en diffusion groupée parce que un paquet corrompu peut être injecté dans la session à proximité de la racine de l'arborescence de diffusion groupée, et dans ce cas, le paquet corrompu va arriver chez de nombreux receveurs. Ceci est particulièrement un problème quand le code décrit dans le présent document est utilisé, parce que l'utilisation de même un paquet corrompu contenant des données de codage peut résulter en ce que le décodage d'un objet soit complètement corrompu et inutilisable. Il est donc RECOMMANDÉ que l'authentification de la source et la vérification de l'intégrité soient appliquées aux objets décodés avant de livrer les objets à une application. Par exemple, un hachage SHA-1 [SHA1] d'un objet peut être ajouté avant transmission, et le hachage SHA-1 est calculé et vérifié après que l'objet est décodé mais avant qu'il soit livré à une application. L'authentification de la source DEVRAIT être fournie, par exemple, en incluant une signature numérique vérifiable par le receveur, calculée par dessus la valeur du hachage. Il est aussi RECOMMANDÉ qu'un protocole d'authentification de paquet, comme TESLA [RFC4082], soit utilisé pour détecter et éliminer les paquets corrompus à l'arrivée. Cette méthode peut aussi être utilisée pour fournir l'authentification de la source. De plus, il est RECOMMANDÉ que la vérification de transmission sur le chemin inverse soit activée dans tous les routeurs et commutateurs du réseau le long du chemin de l'expéditeur aux receveurs pour limiter la possibilité qu'un agent malveillant réussisse à injecter un paquet corrompu dans le chemin des données de l'arborescence de diffusion groupée.

Un autre souci de sécurité est que certaines informations de FEC peuvent être obtenues hors bande par les receveurs dans une description de session, et si la description de session est falsifiée ou corrompue, alors les receveurs ne vont pas utiliser le protocole correct pour décoder le contenu des paquets reçus. Pour éviter ces problèmes, il est RECOMMANDÉ que des mesures soient prises pour empêcher les receveurs d'accepter des descriptions de session incorrectes, par exemple, en utilisant l'authentification de la source pour s'assurer que les receveurs acceptent seulement des descriptions de session légitimes provenant d'expéditeurs autorisés.

## 7. Considérations relatives à l'IANA

Les valeurs des identifiants de codage de FEC et des identifiants d'instance de FEC sont soumises à enregistrement par l'IANA. Pour les lignes directrices générales sur les considérations relatives à l'IANA telles qu'elles s'appliquent au présent document, voir la [RFC5052]. Le présent document alloue l'identifiant de codage de FEC pleinement spécifié de 1 sous `ietf:rmt:fec:encoding-name-space` à "code Raptor".

## 8. Remerciements

De nombreuses améliorations rédactionnelles et précisions ont été apportées à la présente spécification durant le processus de relecture au sein du 3GPP. Des remerciements sont dus pour cela aux membres du groupe de travail 4 du groupe de spécification technique SA du 3GPP.

## 9. Références

### 9.1 Références normatives

- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (MàJ par [RFC8174](#))
- [RFC4082] A. Perrig et autres, "[Authentification de flux tolérante aux pertes](#) en temps efficace (TESLA) : Introduction à la transformation d'authentification de source de diffusion groupée", juin 2005. (*Information*)
- [RFC5052] M. Watson et autres, "[Bloc de construction de la correction](#) d'erreur directe (FEC)", août 2007. (*Remplace RFC3452*) (P.S.)

### 9.2 Références pour information

- [CCNC] Luby, M., Watson, M., Gasiba, T., Stockhammer, T., and W. Xu, "Raptor Codes for Reliable Download Delivery in Wireless Broadcast Systems", CCNC 2006, Las Vegas, NV, janvier 2006.

[MBMS] 3GPP TS 26.346 6.1.0, "Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs", juin 2005.

[Raptor] Shokrollahi, A., "Raptor Codes", IEEE Transactions on Information Theory n° 6, juin 2006.

[RFC3453] M. Luby et autres, "[Utilisation de la correction d'erreur directe](#) (FEC) en diffusion groupée fiable", décembre 2002. (*Info.*)

[SHA1] FIPS PUB 180-1, "Secure Hash Standard", Federal Information Processing Standards Publication, avril 2005.

## Adresse des auteurs

Michael Luby  
Digital Fountain  
9141 Civic Center Drive  
Suite 300  
Fremont, CA 94538  
U.S.A.  
mél : [luby@digitalfountain.com](mailto:luby@digitalfountain.com)

Amin Shokrollahi  
EPFL  
Laboratory of Algorithmic Mathematics  
IC-IIF-ALGO PSE-A  
Lausanne 1015  
Switzerland  
mél : [amin.shokrollahi@epfl.ch](mailto:amin.shokrollahi@epfl.ch)

Mark Watson  
Digital Fountain  
39141 Civic Center Drive  
Suite 300  
Fremont, CA 94538  
U.S.A.  
mél : [mark@digitalfountain.com](mailto:mark@digitalfountain.com)

Thomas Stockhammer  
Nomor Research  
Brecherspitzstrasse 8  
Munich 81541  
Germany  
mél : [stockhammer@nomor.de](mailto:stockhammer@nomor.de)

## Déclaration complète de droits de reproduction

Copyright (C) The IETF Trust (2007)

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations contenues sont fournis sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY, le IETF TRUST et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations encloses ne viole aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

### Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur le répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).