

announcing the
ADVANCED ENCRYPTION STANDARD
(AES)
adaptation française du FIPS 197

turms_rep@yahoo.fr

Copyright (c) 2004 Vincent CONIN. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Table des matières

1	Préliminaires Mathématiques	5
1.1	Les opérations dans $GF(2^8)$	6
1.2	Les polynômes à coefficients dans $GF(2^8)$	7
2	Chiffrement	9
2.1	Notation, structure de données	10
2.2	Chiffrement	11
2.3	La transformation <code>SubBytes()</code>	11
2.4	La transformation <code>ShiftRows()</code>	12
2.5	La transformation <code>MixColumns()</code>	12
2.6	La transformation <code>AddRoundKey()</code>	12
2.7	Gestion de la clé <code>KeyExpansion()</code>	13
3	Déchiffrement	17
3.1	La transformation <code>InvSubBytes()</code>	18
3.2	La transformation <code>InvShiftRows()</code>	18
3.3	La transformation <code>InvMixColumns()</code>	19
3.4	Déchiffrement équivalent	19
4	Implémentation	21
4.1	Modes	23
4.2	Padding	24
A	GNU Free Documentation License	27

Chapitre 1

Préliminaires Mathématiques

1.1 Les opérations dans $GF(2^8)$

Les opérations effectuées par l'AES le sont dans le groupe fini $GF(2^8)$ (*Galois Fields*[GAL03])[19701][RD02].

Nous l'appellerons indifféremment $GF(2^8)$ ou $\frac{\mathbb{Z}_2[X]}{m(X)\mathbb{Z}_2[X]}$ dans ce qui suit.

$$GF(2^8) \Leftrightarrow \frac{\mathbb{Z}_2[X]}{m(X)\mathbb{Z}_2[X]} \quad (1.1)$$

Ceci signifie que les calculs sont effectués sur des polynômes de degré 7 à coefficients dans $\{0, 1\}$, *modulo* un polynôme $m(X)$ de degré 8 (valant dans l'AES $X^8 + X^4 + X^3 + X + 1$).

$$\mathbb{Z}_2[X] \Leftrightarrow \{0, 1\} \quad (1.2)$$

$$m(x) = X^8 + X^4 + X^3 + X + 1 \quad (1.3)$$

Cette représentation est très commode pour travailler sur l'espace des octets, en assimilant ces octets à des polynômes (01001101₂ à $X^6 + X^3 + X^2 + 1$ par exemple) on dispose des outils propres à ces derniers pour générer des transformations sur les données en machine (addition, multiplication).

$$m(X) = \begin{array}{l} X^6 + X^3 + X^2 + 1 \\ X^8 + X^4 + X^3 + X + 1 \end{array} \Leftrightarrow \begin{array}{l} 01001101_2 \\ 100011011_2 \end{array} \quad (1.4)$$

De plus, $m(X)$ étant irréductible, $\frac{\mathbb{Z}_2[X]}{m(X)\mathbb{Z}_2[X]}$ forme un corps à 2^8 éléments (les octets de 0 à 255), où l'addition de deux polynômes équivaut à un ou-exclusif bit à bit entre les octets les représentants,

$$\begin{array}{l} X^6 + X^3 + X^2 + 1 \\ \oplus X^7 + X^4 + X^3 + X + 1 \\ \hline = X^7 + X^6 + X^4 + X^2 + X \end{array} \Leftrightarrow \begin{array}{l} 01001101_2 \\ \oplus 10011011_2 \\ \hline = 11010110_2 \end{array} \quad (1.5)$$

où la multiplication de polynômes est une multiplication binaire entre octets *modulo* $m(X)$, où chaque élément hormis le polynôme nul possède un inverse¹ et où la multiplication par X est un décalage de l'octet d'un bit vers la gauche *modulo* $m(X)$. Pour calculer le reste d'un polynôme $p(X)$ *modulo* $m(X)$ il suffit d'appliquer l'algorithme suivant (ou une de ses variantes) :

$$\begin{array}{l} \mathbf{tant\ que\ } \text{degré}(p) \geq 8 \mathbf{\ faire} \\ p(X) \leftarrow p(X) \oplus m(X) \cdot X^{\text{degré}(p) - \text{degré}(m)} \\ \mathbf{Ftant\ que} \end{array} \quad (1.6)$$

l'exemple suivant résume ces propriétés :

$$\begin{array}{ll} 11101010 & X^7 + X^6 + X^5 + X^3 + X \\ \otimes 00000101 & X^2 + 1 \\ \hline 11101010 & \\ \oplus 1110101000 & \\ \hline = 1101000010 & \text{résultat brut} \\ \oplus 1000110110 & \text{modulo } X \cdot m(X) \\ \hline = 101110100 & \\ \oplus 100011011 & \text{modulo } m(X) \\ \hline = 01101111 & \text{résultat : } X^6 + X^5 + X^3 + X^2 + X + 1. \end{array} \quad (1.7)$$

¹que l'on calcul par l'algorithme de Bezout (algorithme d'Euclide étendu)

Notons que la multiplication d'un polynôme $p(X)$ par 1 revient à réduire $p(X)$ modulo $m(X)$, que la multiplication de $p(X)$ par X (soit 10_2) revient à décaler l'octet $p(X)$ d'un bit sur la gauche et à le réduire modulo $m(X)$ et enfin que la multiplication par $X + 1$ (soit 11_2) revient à effectuer une multiplication par X et une addition.

$$\begin{array}{r} 10000010 \quad X^7 + X \\ \otimes \quad \quad \quad 01 \quad 1 \\ \hline = \quad 10000010 \quad X^7 + X \end{array} \quad (1.8)$$

$$\begin{array}{r} 10000010 \quad X^7 + X \\ \otimes \quad \quad \quad 10 \quad X \\ \hline = \quad 100000100 \\ \oplus \quad 100011011 \quad \text{modulo } m(X) \\ \hline = \quad 00011111 \quad X^4 + X^3 + X^2 + X + 1 \end{array} \quad (1.9)$$

$$\begin{array}{r} 10000010 \quad X^7 + X \\ \otimes \quad \quad \quad 11 \quad X + 1 \\ \hline = \quad 100000100 \quad X^7 + X \otimes X \\ \oplus \quad 100011011 \quad \text{modulo } m(X) \\ \hline = \quad 00011111 \quad X^4 + X^3 + X^2 + X + 1 \\ \oplus \quad 10000010 \quad X^7 + X \\ \hline = \quad 10011101 \quad X^7 + X^4 + X^3 + X^2 + 1 \end{array} \quad (1.10)$$

1.2 Les polynômes à coefficients dans $GF(2^8)$

Tout comme un octet peut être représenté par un polynôme de degré 7, un mot de 32 bits (4 octets) peut l'être par un polynôme de degré 3 à coefficients dans $\frac{\mathbb{Z}_2[X]}{m(X)\mathbb{Z}_2[X]}$, chaque coefficient représentant un octet du mot.

$$\begin{array}{l} \text{0x5b 0x99 0xb3 0xe7} \\ \Leftrightarrow 01011011_2 \ 10011001_2 \ 10110011_2 \ 11100111_2 \\ \Rightarrow (01011011_2)X^3 + (10011001_2)X^2 + (10110011_2)X + (11100111_2) \\ \Leftrightarrow a_1X^3 + a_2X^2 + a_1X + a_0 = a(X) \end{array} \quad (1.11)$$

L'addition de deux mots est alors égale à l'addition des polynômes les représentants, soit un ou-exclusif entre les coefficients de même degré.

$$a(X) + b(X) = (a_3 \oplus b_3)X^3 + (a_2 \oplus b_2)X^2 + (a_1 \oplus b_1)X + (a_0 \oplus b_0) \quad (1.12)$$

La multiplication de deux mots donne un polynôme de degré 3+3 dont on peut calculer les coefficients par la définition précédente et la définition générale du produit de polynômes.

$$a(X) \times b(X) = c(X) = c_6X^6 + c_5X^5 + c_4X^4 + c_3X^3 + c_2X^2 + c_1X + c_0 \quad (1.13)$$

$$\begin{array}{l} c_0 = a_0 \otimes b_0 \\ c_1 = a_1 \otimes b_0 \oplus a_0 \otimes b_1 \\ c_2 = a_2 \otimes b_0 \oplus a_1 \otimes b_1 \oplus a_0 \otimes b_2 \\ c_3 = a_3 \otimes b_0 \oplus a_2 \otimes b_1 \oplus a_1 \otimes b_2 \oplus a_0 \otimes b_3 \\ c_4 = a_3 \otimes b_1 \oplus a_2 \otimes b_2 \oplus a_1 \otimes b_3 \\ c_5 = a_3 \otimes b_2 \oplus a_2 \otimes b_3 \\ c_6 = a_3 \otimes b_3 \end{array} \quad (1.14)$$

Afin de rester dans l'espace des mots de 32 bits, on considère les polynômes $c(X)$ modulo un polynôme de degré 4 qui vaut pour l'AES $X^4 + 1$, formant un groupe que l'on pourrait écrire

$$\mathcal{A} = \frac{GF(2^8)[X]}{(X^4 + 1)GF(2^8)[X]} \quad (1.15)$$

où

$$\begin{aligned} & a(X) \otimes b(X) \\ &= c(X) \text{ modulo } (X^4 + 1) \\ &= d(X) \\ &= d_3X^3 + d_2X^2 + d_1X + d_0 \end{aligned} \quad (1.16)$$

on obtient par calcul² les coefficients d_i

$$\begin{aligned} d_0 &= a_0 \otimes b_0 \oplus a_3 \otimes b_1 \oplus a_2 \otimes b_2 \oplus a_1 \otimes b_3 \\ d_1 &= a_1 \otimes b_0 \oplus a_0 \otimes b_1 \oplus a_3 \otimes b_2 \oplus a_2 \otimes b_3 \\ d_2 &= a_2 \otimes b_0 \oplus a_1 \otimes b_1 \oplus a_0 \otimes b_2 \oplus a_3 \otimes b_3 \\ d_3 &= a_3 \otimes b_0 \oplus a_2 \otimes b_1 \oplus a_1 \otimes b_2 \oplus a_0 \otimes b_3 \end{aligned} \quad (1.17)$$

Ces opérations peuvent être mises sous forme matricielle.

$$\begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} \quad (1.18)$$

Comme $X^4 + 1$ n'est pas un polynôme irréductible, \mathcal{A} n'est pas un corps et ses éléments ne sont pas forcément inversibles, l'AES utilise alors pour ses calculs sur les mots le polynôme $a(X)$ inversible dans \mathcal{A} .

$$\begin{aligned} a(X) &= (0x03)X^3 + (0x01)X^2 + (0x02)X + (0x02) \\ a^{-1}(X) &= (0x0b)X^3 + (0x0d)X^2 + (0x09)X + (0x0e) \end{aligned} \quad (1.19)$$

²simplement en posant la division par $X^4 + 1$

Chapitre 2

Chiffrement

2.1 Notation, structure de données

Avant d'entrer dans le vif du sujet, voici les conventions de notation issues de la spécification du NIST [19701] :

AddRoundKeys()	:	transformation (pour le chiffrement et le déchiffrement) qui ajoute une clé de tour (<i>Round key</i>) au bloc courant (les clés de l'AES sont de 128 bits).
MixColumns()	:	transformation qui permute les colonnes d'un bloc lors du chiffrement.
RotWord()	:	fonction utilisée par la routine d'expansion de la clé qui applique à un mot de 4 octets une permutation circulaire.
ShiftRows()	:	transformation qui applique des permutations circulaires aux trois dernières lignes du bloc lors du chiffrement.
SubBytes()	:	transformation qui opère une substitution non linéaire de chaque octet du bloc en utilisant une table (S-box) lors du chiffrement.
SubWord()	:	fonction utilisée par la routine d'expansion de la clé qui prend un mot en entrée et applique à ses 4 octets une substitution non linéaire (S-box).
InvMixColumns()	:	transformation du déchiffrement qui est l'inverse de la transformation MixColumns() .
InvShiftRows()	:	transformation du déchiffrement qui est l'inverse de la transformation ShiftRows() .
InvSubBytes()	:	transformation du déchiffrement qui est l'inverse de la transformation SubBytes() .
K	:	la clé de chiffrement.
Nb	:	nombre de colonnes (mots de 32 bits) du bloc. Pour l'AES Nb=4 .
Nk	:	nombre de mots de 32 bits dans la clé de chiffrement. Pour l'AES Nk= 4,6 ou 8 .
Nr	:	nombre de tours, fonction de Nb et Nk . Pour l'AES Nr=10,12 ou 14 .
Rcon[]	:	table constante.

Le Rijndael opère sur des *blocs de données de 128, 192 ou 256 bits* (voir figure 1.1 page 14) en utilisant des *clés de chiffrement de 128, 192 ou 256 bits*. La taille de bloc retenue pour l'AES est de 128 bits, par souci de commodité semble-t-il.

Un bloc peut être représenté comme une matrice d'octets $4 \times \mathbf{Nb}$ (*the State*). Les octets lus en entrée y sont copiés colonne après colonne (chaque colonne

représente un mot lu).

2.2 Chiffrement

Le chiffrement transforme les données contenues dans le bloc en itérant 10, 12 ou 14 fois (ceci dépend de la longueur de la clé) quatre transformations sur les octets :

1. une substitution non linéaire
2. une permutation circulaire des octets au sein d'une même ligne
3. une multiplication dans $\frac{GF(2^8)[X]}{(X^4+1)GF(2^8)[X]}$ pour chaque colonne
4. une addition de clé

le dernier tour n'incluant pas l'addition de clé.

Le bloc chiffré est ensuite envoyé vers la sortie puis réinitialisé avec la suite des données. Le pseudo-code de la figure 1.2 décrit ces itérations, les transformations - **SubBytes()**, **ShiftRows()**, **MixColumns()** et **AddRoundKey()** - feront l'objet des sections suivantes. Le tableau $\mathbf{w}[]$ contient la clé telle que décrite section 2.7.

2.3 La transformation SubBytes()

La transformation **SubBytes()** est une substitution non linéaire d'octets, utilisant une table S (*S-box*). Cette table est construite en composant deux transformations :

1. prendre l'inverse de l'octet dans $\frac{\mathbb{Z}_2[X]}{m(X)\mathbb{Z}_2[X]}$ (voir section 1.1), l'octet 0x00 étant par convention son propre inverse.
2. lui appliquer la transformation affine suivante (dans $\frac{\mathbb{Z}_2[X]}{m(X)\mathbb{Z}_2[X]}$) :

$$\text{pour } 0 \leq i < 8 \quad (2.1)$$

$$b'_i = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$$

où b_i est le $i^{\text{ème}}$ bit de l'octet et c_i le $i^{\text{ème}}$ bit d'un octet c qui vaut 01100011₂ (0x63).

Cette transformation peut prendre la forme matricielle suivante :

$$\begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \quad (2.2)$$

La figure 1.3 page 15 illustre cette transformation. La table S issue du précalcul des valeurs de chacun des 256 polynômes de $\frac{\mathbb{Z}_2[X]}{m(X)\mathbb{Z}_2[X]}$ est indexée par les 4 bits

de poids fort et les 4 bits de poids faible de l'octet. Par exemple pour 0x53, la substitution aura lieu avec la valeur située à l'intersection de la ligne 5 et de la colonne 3 de la figure 1.4.

2.4 La transformation ShiftRows()

La transformation **ShiftRows()** applique une permutation circulaire sur les trois dernières lignes du bloc

$$\begin{aligned} & \text{pour } 0 < r < 4 \text{ et } 0 \leq c < \mathbf{Nb} \\ & s''_{r,c} = s'_{r,(c+\text{shift}(r,\mathbf{Nb}))\bmod \mathbf{Nb}} \end{aligned} \quad (2.3)$$

où (pour l'AES) :

$$\text{shift}(1,4) = 1; \text{shift}(2,4) = 2; \text{shift}(3,4) = 3. \quad (2.4)$$

La figure 1.5 illustre cette transformation.

2.5 La transformation MixColumns()

La transformation **MixColumns()** traite chaque colonne comme un polynôme de degré 3, on calcule dans \mathcal{A} (voir la section 1.2) le produit de ce polynôme avec un polynôme fixe $a(X)$.

$$a(X) = (0x03)X^3 + (0x01)X^2 + (0x02)X + (0x02) \quad (2.5)$$

Ces opérations peuvent être mises sous forme matricielle :

$$\begin{aligned} & \text{pour } 0 \leq c < \mathbf{Nb} \\ & \begin{pmatrix} s'''_{0,c} \\ s'''_{1,c} \\ s'''_{2,c} \\ s'''_{3,c} \end{pmatrix} = \begin{pmatrix} 0x02 & 0x03 & 0x01 & 0x01 \\ 0x01 & 0x02 & 0x03 & 0x01 \\ 0x01 & 0x01 & 0x02 & 0x03 \\ 0x03 & 0x01 & 0x01 & 0x02 \end{pmatrix} \begin{pmatrix} s''_{0,c} \\ s''_{1,c} \\ s''_{2,c} \\ s''_{3,c} \end{pmatrix} \end{aligned} \quad (2.6)$$

2.6 La transformation AddRoundKey()

La transformation **AddRoundKey()** addition au bloc une clé de la façon suivante :

1. une clé de tour est extraite de la fonction de gestion des clés (voir section 2.7), celle-ci est composée de \mathbf{Nb} mots de 4 octets.
2. les mots sont additionnés aux colonnes suivant la formule :

$$\text{pour } 0 \leq c < \mathbf{Nb} \\ [s^{iv}_{0,c}; s^{iv}_{1,c}; s^{iv}_{2,c}; s^{iv}_{3,c}] = [s'''_{0,c}; s'''_{1,c}; s'''_{2,c}; s'''_{3,c}] \oplus [w_{\text{round}*\mathbf{Nb}+c}] \quad (2.7)$$

où $w[i]$ représente le $i^{\text{ème}}$ mot de la clé de tour, cette opération est illustrée par la figure 1.6 page 16. Pour $\text{round} = 0$ on effectue l'addition de clé correspondant

à la ligne 5 du pseudo-code 1.2 page 14, pour $0 \leq \textit{round} < \mathbf{Nr} - 1$ celles de la ligne 11 et enfin pour $\textit{round} = \mathbf{Nr}$ celle de la ligne 16.

2.7 Gestion de la clé KeyExpansion()

La routine d'expansion de la clé (voir le pseudo-code 1.7) fournit les $\mathbf{Nb} \times (\mathbf{Nr} + 1)$ blocs de clés de tour nécessaires au chiffrement à partir de la clé secrète \mathbf{K} (rappelons qu'un ou-exclusif est effectué entre le clair et la clé avant d'entrer dans les rondes de chiffrement).

La routine **SubWord()** prend en entrée un mot de 4 octets et substitue à chaque octet sa valeur correspondante de la table S, la routine **RotWord()** prend aussi en entrée un mot de 4 octets $[a_0, a_1, a_2, a_3]$ et lui applique la permutation circulaire $[a_1, a_2, a_3, a_0]$, le tableau **Rcon[]** est construit ainsi (voir l'exponentiation dans $\text{GF}(2^8)$ section 1.1) :

$$\text{Rcon}[i] = [X^{i-1}, 0x00, 0x00, 0x00] \quad 1 \geq i \quad (X \equiv 0x02) \quad (2.8)$$

Notons que la clé \mathbf{K} forme les \mathbf{Nk} premiers blocs de la clé étendue et que la routine est légèrement différente lorsque la taille de la clé \mathbf{K} est de 256 bits ($\mathbf{Nk} = 8$), dans ce cas si $i - 4$ est un multiple de \mathbf{Nk} , **SubWord()** est appliquée avant le ou-exclusif avec $w[i-\mathbf{Nk}]$.

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

$s_{0,0}, s_{1,0}, s_{2,0}, s_{3,0}$ représente un mot de 32 bits w_0 .

$$w_0 = s_{0,0} s_{1,0} s_{2,0} s_{3,0} \quad w_1 = s_{0,1} s_{1,1} s_{2,1} s_{3,1}$$

$$w_2 = s_{0,2} s_{1,2} s_{2,2} s_{3,2} \quad w_3 = s_{0,3} s_{1,3} s_{2,3} s_{3,3}$$

FIG. 1.1: structure d'un bloc d'octets

```

Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
début
1   byte state[4,Nb]
2
3   state = in      // l'entrée est copiée dans le bloc
4
5   AddRoundKey(state, w[0, Nb-1])
6
7   pour round = 1 à Nr-1 par pas de 1 faire
8       SubBytes(state)
9       ShiftRows(state)
10      MixColumns(state)
11      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
12  finpour
13
14  SubBytes(state)
15  ShiftRows(state)
16  AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
17
18  out = state     // copie du bloc dans la sortie
fin

```

FIG. 1.2: pseudo-code - chiffrement

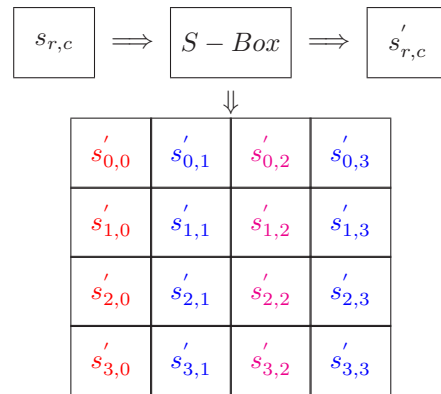
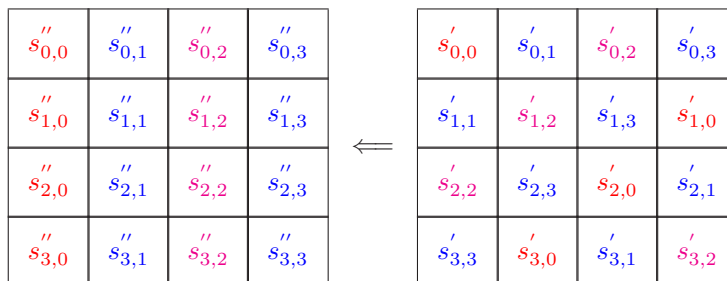


FIG. 1.3: substitution par table S

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

FIG. 1.4: la table S

FIG. 1.5: bloc après la transformation **ShiftRows()**

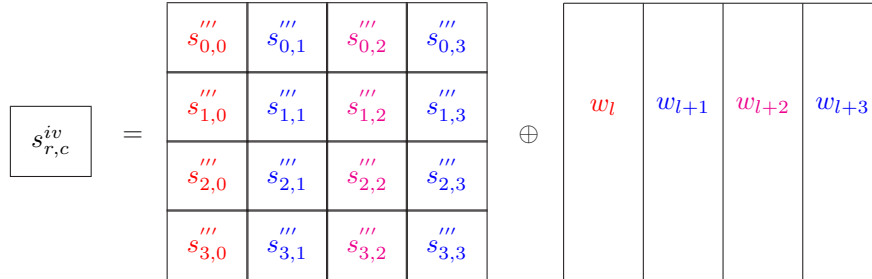


FIG. 1.6: **AddRoundKey()** : $l = round * Nb$

```

KeyExpansion (byte key[4*Nk], word w[Nb * (Nr+1)], Nk)
début
    word temp

    i = 0;

    tant que (i < Nk) faire
        w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
        i = i+1
    fin tant que

    i = Nk

    tant que (i < Nb * (Nr+1)) faire
        tmp = w[i-1]
        si (i mod Nk = 0) alors
            tmp = SubWord(RotWord(tmp)) Xor Rcon[i/Nk]
        sinon si (Nk > 6 et i mod Nk = 4) alors
            tmp = SubWord(tmp)
        fin si
        w[i] = w[i-Nk] Xor tmp
        i = i + 1
    fin tant que
fin
    
```

FIG. 1.7: pseudo-code - KeyExpansion

Chapitre 3

Déchiffrement

La routine de chiffrement (pseudo-code 1.2) peut être inversée et réordonnée pour produire une routine de déchiffrement (pseudo-code 1.8). Les transformations **InvSubBytes()**, **InvShiftRows()**, **InvMixColumns()**, inverses des opérations de chiffrement seront succinctement décrites dans la suite, puis nous donnerons une routine équivalente plus facile à implémenter (pseudo-code 1.10)

```

InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
début
1      byte state[4,Nb]
2
3      state = in      // l'entrée est copiée dans le bloc
4
5      AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
6
7      pour round = Nr-1 à 1 par pas de -1 faire
8          InvShiftRows(state)
9          InvSubBytes(state)
10         AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
11         InvMixColumns(state)
12     finpour
13
14     InvShiftRows(state)
15     InvSubBytes(state)
16     AddRoundKey(state, w[0, Nb-1])
17
18     out = state     // copie du bloc dans la sortie
fin

```

FIG. 1.8: pseudo-code - déchiffrement

3.1 La transformation **InvSubBytes()**

La transformation **InvSubBytes()** substitue à un octet sa valeur équivalente dans la table S_i (1.9), ces valeurs sont calculées en appliquant la transformation inverse de 2.2 puis en inversant le résultat par la *relation de Bezout* dans $\frac{\mathbb{Z}_2[X]}{m(X)\mathbb{Z}_2[X]}$.

$$s'_{r,c} = Si[s_{r,c}] \quad (3.1)$$

3.2 La transformation **InvShiftRows()**

La transformation **InvShiftRows()** est l'inverse de **ShiftRows()**, la permutation circulaire suivante est appliquée aux trois dernières lignes du bloc :

$$\text{pour } 0 < r < 4 \text{ et } 0 \leq c < \mathbf{Nb} \\ s''_{r,(c+shift(r,Nb)) \bmod Nb} = s'_{r,c} \quad (3.2)$$

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

FIG. 1.9: la table Si

3.3 La transformation `InvMixColumns()`

La transformation `invMixColumns()` est l'inverse de la transformation `MixColumns()`, traitant chaque colonne comme un polynôme de degré 3, on calcule dans \mathcal{A} (voir la section 1.2) le produit de ce polynôme avec un polynôme fixe $a^{-1}(X)$.

$$a^{-1}(X) = (0x0b)X^3 + (0x0d)X^2 + (0x09)X + (0x0e) \quad (3.3)$$

Ces opérations peuvent être mises sous forme matricielle :

$$\begin{pmatrix} s'''_{0,c} \\ s'''_{1,c} \\ s'''_{2,c} \\ s'''_{3,c} \end{pmatrix} = \begin{matrix} \text{pour } 0 \leq c < \mathbf{Nb} \\ \begin{pmatrix} 0x0e & 0x0b & 0x0d & 0x09 \\ 0x09 & 0x0e & 0x0b & 0x0d \\ 0x0d & 0x09 & 0x0e & 0x0b \\ 0x0b & 0x0d & 0x09 & 0x0e \end{pmatrix} \end{matrix} \begin{pmatrix} s''_{0,c} \\ s''_{1,c} \\ s''_{2,c} \\ s''_{3,c} \end{pmatrix} \quad (3.4)$$

3.4 Déchiffrement équivalent

Dans la version basique du déchiffrement (pseudo-code 1.2) la séquence de transformations diffère de celle du chiffrement (pseudo-code 1.8), le traitement de la clé restant inchangé. Certaines propriétés du Rijndael permettent d'implémenter une routine de déchiffrement qui respecte la séquence de transformations de la routine `Cipher()`, la structure de celle-ci étant la plus efficace on adoptera cette routine équivalente qui nécessite tout de même une modification de la gestion de la clé.

Les deux propriétés qui permettent ce changement sont les suivantes :

1. les transformations **SubBytes()** et **ShiftRows()** commutent (leurs inverses respectifs aussi).
2. les transformations **MixColumns()** et **InvMixColumns()** sont linéaires :

$$\text{InvMixColumns}(\text{state XOR Round Key}) = \\ \text{InvMixColumns}(\text{state}) \text{ XOR } \text{InvMixColumns}(\text{Round Key})$$

Ces propriétés permettent donc d'inverser l'ordre des transformations **InvSubBytes()** et **InvShiftRows()**. L'ordre des transformations **AddRoundKey()** et **InvMixColumns()** peut aussi être inversé si les colonnes (les mots de 32 bits) sont modifiées en utilisant **InvMixColumns()**. Les premiers et derniers **Nb** mots de la clé étendue *ne doivent pas être modifiés de cette manière*. Le pseudo-code du déchiffrement équivalent est donné en 1.10.

```

EqInvCipher(byte in[4*Nb], byte out[4*Nb], word dw[Nb*(Nr+1)])
début
1   byte state[4,Nb]
2
3   state = in      // l'entrée est copiée dans le bloc
4
5   AddRoundKey(state, dw[Nr*Nb, (Nr+1)*Nb-1])
6
7   pour round = Nr-1 à -1 par pas de -1 faire
8       InvSubBytes(state)
9       InvShiftRows(state)
10      InvMixColumns(state)
11      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
12  finpour
13
14  InvSubBytes(state)
15  InvShiftRows(state)
16  AddRoundKey(state, w[0, Nb-1])
17
18  out = state    // copie du bloc dans la sortie
fin

ceci implique l'aménagement suivant dans la routine d'expansion
de la clé :

    pour i = 0 à (Nr+1*Nb-1) par pas de 1 faire
        dw[i]=w[i]
    fin pour
    pour round = 1 à Nr-1 par pas de 1 faire
        InvMixColumns(dw[round*Nb, (round+1)*Nb-1])
    fin pour

```

FIG. 1.10: pseudo-code - déchiffrement équivalent

Chapitre 4

Implémentation

L'implantation de l'AES que nous allons mettre en oeuvre est dédiée à une machine 32 bits, Intel ou autre [Gla03].

Nous allons dans ce qui suit montrer comment les transformations appliquées aux octets du bloc à chaque itération de chiffrement par le Rijndael peuvent être combinées pour ne former qu'une unique transformation itérée, dont les résultats seront facilement stockables dans des tables [RD99].

Remarquons d'abord que la substitution par table S (**SubBytes(s)**) implique que les octets du bloc peuvent être directement identifiés à leur entrée dans la table S :

$$s_{i,j} \leftarrow S[s_{i,j}] \quad (4.1)$$

ainsi une colonne sera représentée par :

$$\begin{bmatrix} S[s_{0,j}] \\ S[s_{1,j}] \\ S[s_{2,j}] \\ S[s_{3,j}] \end{bmatrix} \quad (4.2)$$

ensuite la permutation circulaire **ShiftRows(s)** donne pour chaque colonne ¹ :

$$\begin{bmatrix} S[s_{0,j}] \\ S[s_{1,j}] - C1 \\ S[s_{2,j}] - C2 \\ S[s_{3,j}] - C3 \end{bmatrix} \quad (4.3)$$

ce qui est identique à :

$$\begin{bmatrix} S[s_{0,j}] \\ S[s_{1,j} - C1] \\ S[s_{2,j} - C2] \\ S[s_{3,j} - C3] \end{bmatrix} \quad (4.4)$$

chaque colonne est ensuite multipliée par la matrice de **MixColumns(s)** :

$$\begin{pmatrix} 0x02 & 0x03 & 0x01 & 0x01 \\ 0x01 & 0x02 & 0x03 & 0x01 \\ 0x01 & 0x01 & 0x02 & 0x03 \\ 0x03 & 0x01 & 0x01 & 0x02 \end{pmatrix} \begin{bmatrix} S[s_{0,j}] \\ S[s_{1,j} - C1] \\ S[s_{2,j} - C2] \\ S[s_{3,j} - C3] \end{bmatrix} \quad (4.5)$$

ce que l'on peut décomposer en une combinaison linéaire de vecteurs :

$$S[s_{0,j}] \begin{bmatrix} 2 \\ 1 \\ 1 \\ 3 \end{bmatrix} \oplus S[s_{1,j} - C1] \begin{bmatrix} 3 \\ 2 \\ 1 \\ 1 \end{bmatrix} \oplus S[s_{2,j} - C2] \begin{bmatrix} 1 \\ 3 \\ 2 \\ 1 \end{bmatrix} \oplus S[s_{3,j} - C3] \begin{bmatrix} 1 \\ 1 \\ 3 \\ 2 \end{bmatrix} \quad (4.6)$$

à quoi l'on ajoute la clé de tour par la transformation finale **AddRound-**

¹C1, C2 et C3 traduisent les décalages de 1, 2 ou 3 colonnes exercés sur les lignes du bloc par **ShiftRows(s)**

Key(s) :

$$S[s_{0,j}] \begin{bmatrix} 2 \\ 1 \\ 1 \\ 3 \end{bmatrix} \oplus S[s_{1,j}-C1] \begin{bmatrix} 3 \\ 2 \\ 1 \\ 1 \end{bmatrix} \oplus S[s_{2,j}-C2] \begin{bmatrix} 1 \\ 3 \\ 2 \\ 1 \end{bmatrix} \oplus S[s_{3,j}-C3] \begin{bmatrix} 1 \\ 1 \\ 3 \\ 2 \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix} \quad (4.7)$$

définissons les tables T_0 à T_3 telles que :

$$T_0[s] = \begin{bmatrix} S[s] \otimes 2 \\ S[s] \\ S[s] \\ S[s] \otimes 3 \end{bmatrix} \quad T_1[s] = \begin{bmatrix} S[s] \otimes 3 \\ S[s] \otimes 2 \\ S[s] \\ S[s] \end{bmatrix} \quad T_2[s] = \begin{bmatrix} S[s] \\ S[s] \otimes 3 \\ S[s] \otimes 2 \\ S[s] \end{bmatrix} \quad T_3[s] = \begin{bmatrix} S[s] \\ S[s] \\ S[s] \otimes 3 \\ S[s] \otimes 2 \end{bmatrix} \quad (4.8)$$

un tour de chiffrement peut maintenant s'exprimer ainsi :

$$s_j \leftarrow T_0[s_{0,j}] \oplus T_1[s_{1,j}-C1] \oplus T_2[s_{2,j}-C2] \oplus T_3[s_{3,j}-C3] \oplus k_j. \quad (4.9)$$

Il suffit ainsi de stocker 8 tables de 256×4 octets (en prenant en compte les tables $T_{i,x}$, inverses des $T_{x,i}$), ce qui représente 4Ko de mémoire pour le chiffrement et autant pour le déchiffrement, soit deux pages mémoire sur une architecture 80x86.

4.1 Modes

Nous décrivons ici deux des modes habituellement utilisés avec les algorithmes de chiffrement par bloc (voir [MOV97][Sch01] pour plus de précisions) :

- **le mode ECB (Electronic CodeBook)** (voir figure 4.1) : Ce mode est la méthode la plus simple pour utiliser un algorithme de chiffrement. On chiffre chaque bloc du message clair avec la clé, un bloc de texte clair sera ainsi toujours chiffré en un même bloc de texte chiffré (pour une clé donnée), offrant la possibilité (relativement théorique avec des blocs de 128 bits) à un attaquant de constituer un *cahier de codage de texte clair* (*Electronic CodeBook*), particulièrement efficace sur des messages à *entête stéréotypés* comme les fichiers zippés. Ce mode est rarement utilisé dans des applications cryptographiques, on lui préfère un mode plus élaboré.
- **le mode CBC (Cipher Block Chaining)** (voir figure 4.2) : Ce mode est réputé sûr. Comme pour le mode ECB, on chiffre chaque bloc du message clair avec la clé, mais avant de chiffrer le bloc clair courant on effectue entre celui-ci et le bloc précédemment chiffré un ou-exclusif, entraînant ainsi un chiffrement différent pour deux blocs identiques. Le premier ou-exclusif est effectué entre le premier bloc clair et un *vecteur d'initialisation* aléatoire qui sera transmis avec le message chiffré. Cette précaution rend l'attaque par dictionnaire irréalisable.

FIG. 4.1 – le mode ECB

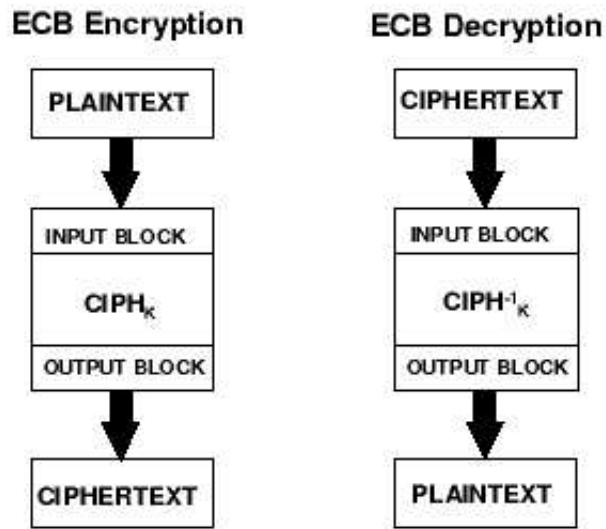
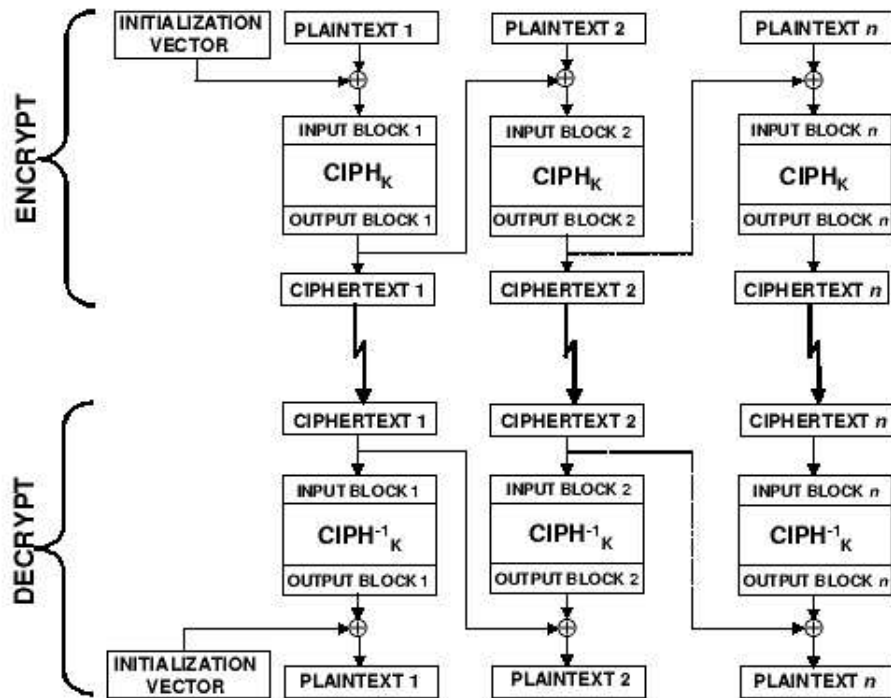


FIG. 4.2 – le mode CBC



4.2 Padding

Les modes ECB et CBC utilisés pour l'AES nécessitent des entrées de **Nb** mots. Si la taille des données à chiffrer est inférieure à **Nb** × octets, les octets vides doivent être remplis (*padded*) et ce remplissage doit faire l'objet d'une notification pour que le programme du destinataire en tienne compte. Plusieurs méthodes existent, notamment dans [80001].

Bibliographie

- [19701] Federal Information Processing Standard Publication 197. Specifications for the Advanced Encryption Standard, 2001.
- [80001] Federal Information Processing Standard Special Publication 800-xx. Recommendation for Block Cipher Modes of Operation, 2001.
- [GAL03] les génies de la science : Evariste galois. Pour la Science, 2003.
- [Gla03] Brian Gladman. A specification for Rijndael, the AES algorithm, 2003.
- [MOV97] A. Menezes, P. Van Oorschot, and S. Vanstone. *Handbook of applied cryptography*. CRC Press, 1997.
- [RD99] Vincent Rijmen and Joan Daemen. AES proposal : Rijndael, 1999.
- [RD02] Vincent Rijmen and Joan Daemen. Un nouvel algorithme de chiffrement. Pour la Science hors serie : l'art du secret, 2002.
- [Sch01] Bruce Schneier. *Cryptographie appliquée : algorithmes, protocoles et codes source en C*. Vuibert, 2001.

Annexe A

GNU Free Documentation License

GNU Free Documentation License
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML

or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover

Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.

- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or

by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights

of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.