Hachage cryptographique : MD5

pseudo-code

Les variables sont sur 32 bits

//Définir r comme suit :
var entier[64] r, k
$r[0..15] := \{7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22\}$
$r[16..31] := \{5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20\}$
$r[32..47] := \{4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23\}$
$r[48..63] := \{6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21\}$

//MD5 utilise des sinus d'entiers pour ses constantes :
pour i de 0 à 63 faire
    $k[i] := floor(abs(sin(i + 1)) \times 2^{32})$
fin pour

//Préparation des variables :
var entier $h0 := 0x67452301$
var entier $h1 := 0xEFCDAB89$
var entier $h2 := 0x98BADCFE$
var entier $h3 := 0x10325476$

//Préparation du message (padding) :
ajouter un bit "1" à la fin du message
ajouter des "0" jusqu'à ce que la taille du message en bits soit égale à 448 (mod 512)
ajouter la taille du message codée en 64-bit au message

//Découpage en blocs de 512 bits :
pour chaque bloc de 512 bits du message
    subdiviser en 16 mots de 32 bits $w[i]$, $0 \leq i \leq 15$

    //initialiser les valeurs de hachage :
    var entier $a := h0$
    var entier $b := h1$
    var entier $c := h2$
    var entier $d := h3$

    //Boucle principale :
    pour i de 0 à 63 faire
        si $0 \leq i \leq 15$ alors
            $f :=$ (b et c) ou ((non b) et d)
            $g := i$
        sinon si $16 \leq i \leq 31$ alors
            $f :=$ (d et b) ou ((non d) et c)
            $g := (5 \times i + 1) \mod 16$
        sinon si $32 \leq i \leq 47$ alors
            $f :=$ b xor c xor d

$g := (3 \times i + 5) \mod 16$
sinon si $48 \leq i \leq 63$ alors
$f := c$ xor (b ou (non d))
$g := (7 \times i) \mod 16$
fin si
var entier temp := d
$d := c$
$c := b$
$b := ((a + f + k[i] + w[g])$ leftrotate $r[i]) + b$
$a :=$ temp
fin pour

//ajouter le résultat au bloc précédent :
$h0 := h0 + a$
$h1 := h1 + b$
$h2 := h2 + c$
$h3 := h3 + d$
fin pour

var entier empreinte := $h0$ concaténer $h1$ concaténer $h2$ concaténer $h3$