Completed by Constantin Vigulear, group I1802.

**TEMA:** MESSAGE DIGEST 4 (MD)

**1.1 Partea teoretică:**

Algoritmul MD4 Message-Digest este o funcție de hash criptografică dezvoltată de Ronald Rivest în 1990. Lungimea digestiei este de 128 biți. Algoritmul a influențat proiectările ulterioare, precum algoritmii MD5, SHA-1 și RIPEMD. Initialismul "MD" inseamna "Message Digest".

O operațiune MD4: MD4 constă din 48 din aceste operațiuni, grupate în trei runde de 16 operații. F este o funcție neliniară; o funcție este utilizată în fiecare rundă. Mi denotă un bloc pe 32 de biți din intrarea mesajului, iar Ki denotă o constantă pe 32 de biți, diferită pentru fiecare operație.

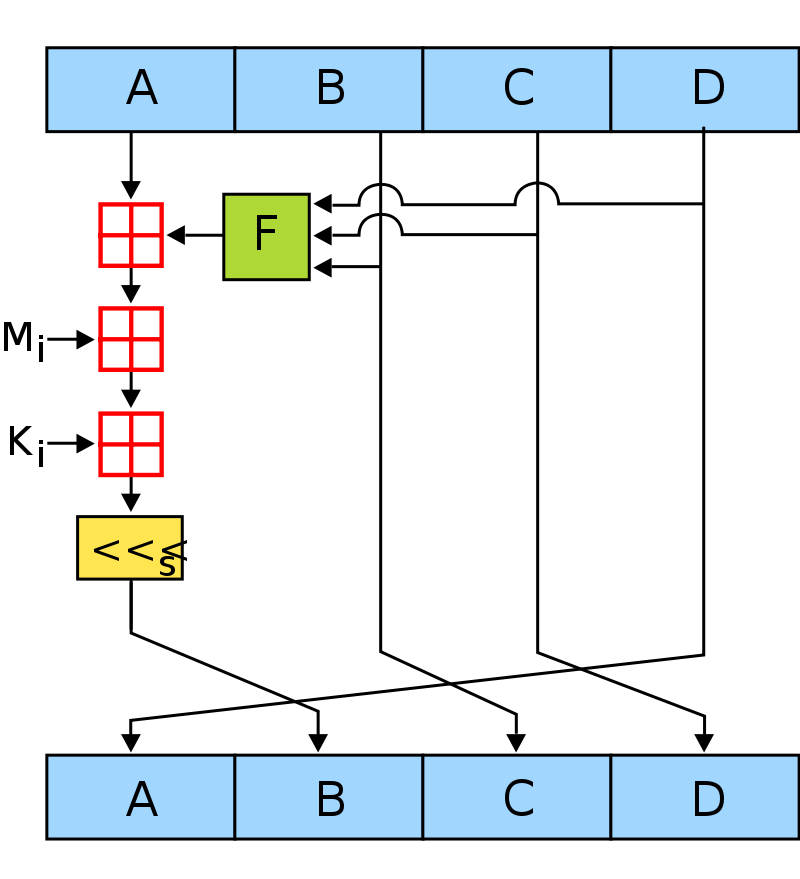
Securitatea MD4 a fost grav compromisă. Primul atac complet de coliziune împotriva MD4 a fost publicat în 1995 și mai multe atacuri mai noi au fost publicate de atunci. Începând cu 2007, un atac poate genera coliziuni în mai puțin de 2 operații de hash MD4. Există și un atac teoretic preimage.

O variantă de MD4 este utilizată în schema URI ed2k pentru a furniza un identificator unic pentru un fișier din rețelele populare eDonkey2000 / eMule P2P. MD4 a fost folosit și de protocolul rsync (anterior versiunii 3.0.0.). MD4 este utilizat pentru a calcula calculele cheie derivate de la parolă NTLM pe Microsoft Windows NT, XP, Vista, 7, 8 și 10.

**1.2 PARTEA PRACTICĂ:**

Fie un mesaj aleatoriu care intentionam sa fie mesajul pentru parola.(primele doua simboluri trebuie sa fieinitialele student. Vigulear Constantin - VC)

Mesaj: VC**ElectronicaDig** (16 SIMBOLURI)---128 Biți



**Figura 1.1** Schema de criptare al algoritmului MD4

**1.2.1 CRIPTAREA MESAJULUI:**

01010110 01000011 01000101 01101100 01100101 01100011 01110100 01110010 01101111 01101110 01101001 01100011 01100001 01000100 01101001 01100111

A => 01010110 01000011 01000101 01101100 (1,447,249,260) 010101100100001  
B => 01100101 01100011 01110100 01110010 (1,701,016,690)

C => 01101111 01101110 01101001 01100011 (1,869,506,915)

D => 01100001 01000100 01101001 01100111 (1,631,873,383)

|  |
| --- |
| \*\*\* ROUND 1 \*\*\*  1. 01100101011000110111010001110010 xor 01101111011011100110100101100011 xor 01100001010001000110100101100111 = 1799976054 = 1101011010010010111010001110110  2. 01010110010000110100010101101100 + 1799976054 mod 2^32 = 3247225314 = 11000001100011001011100111100010  3. 3247225314 + 0 mod 2^32 = 3247225314 = 11000001100011001011100111100010  4. 3247225314 <<< 3 = 25977802512 = 11000001100011001011100111100010000  5. A = 01100001010001000110100101100111 (1631873383)  B = 11000001100011001011100111100010000 (25977802512)  C = 01100101011000110111010001110010 (1701016690)  D = 01101111011011100110100101100011 (1869506915) |
| \*\*\* ROUND 2 \*\*\*  1. 11000001100011001011100111100010000 xor 01100101011000110111010001110010 xor 01101111011011100110100101100011 = 25877336577 = 11000000110011010001101001000000001  2. 01100001010001000110100101100111 + 25877336577 mod 2^32 = 1739406184 = 1100111101011010011101101101000  3. 1739406184 + 1 mod 2^32 = 1739406185 = 1100111101011010011101101101001  4. 1739406185 <<< 7 = 222643991680 = 11001111010110100111011011010010000000  5. A = 01101111011011100110100101100011 (1869506915)  B = 11001111010110100111011011010010000000 (222643991680)  C = 11000001100011001011100111100010000 (25977802512)  D = 01100101011000110111010001110010 (1701016690) |
| \*\*\* ROUND 3 \*\*\*  1. 11001111010110100111011011010010000000 xor 11000001100011001011100111100010000 xor 01100101011000110111010001110010 = 230847877090 = 11010110111111100110110000111111100010  2. 01101111011011100110100101100011 + 230847877090 mod 2^32 = 789150021 = 101111000010010111100101000101  3. 789150021 + 2 mod 2^32 = 789150023 = 101111000010010111100101000111  4. 789150023 <<< 11 = 1616179247104 = 10111100001001011110010100011100000000000  5. A = 01100101011000110111010001110010 (1701016690)  B = 10111100001001011110010100011100000000000 (1616179247104)  C = 11001111010110100111011011010010000000 (222643991680)  D = 11000001100011001011100111100010000 (25977802512) |
| \*\*\* ROUND 4 \*\*\*  1. 10111100001001011110010100011100000000000 xor 11001111010110100111011011010010000000 xor 11000001100011001011100111100010000 = 1432660099984 = 10100110110010001001100100100001110010000  2. 01100101011000110111010001110010 + 1432660099984 mod 2^32 = 4137007106 = 11110110100101011011100000000010  3. 4137007106 + 3 mod 2^32 = 4137007109 = 11110110100101011011100000000101  4. 4137007109 <<< 19 = 2168983183163392 = 111101101001010110111000000001010000000000000000000  5. A = 11000001100011001011100111100010000 (25977802512)  B = 111101101001010110111000000001010000000000000000000 (2168983183163392)  C = 10111100001001011110010100011100000000000 (1616179247104)  D = 11001111010110100111011011010010000000 (222643991680) |
| \*\*\* ROUND 5 \*\*\*  1. 111101101001010110111000000001010000000000000000000 xor 10111100001001011110010100011100000000000 xor 11001111010110100111011011010010000000 = 2170325852720256 = 111101101011110011001011101011111111000110010000000  2. 11000001100011001011100111100010000 + 2170325852720256 mod 2^32 = 1776638864 = 1101001111001010101101110010000  3. 1776638864 + 4 mod 2^32 = 1776638868 = 1101001111001010101101110010100  4. 1776638868 <<< 3 = 14213110944 = 1101001111001010101101110010100000  5. A = 11001111010110100111011011010010000000 (222643991680)  B = 1101001111001010101101110010100000 (14213110944)  C = 111101101001010110111000000001010000000000000000000 (2168983183163392)  D = 10111100001001011110010100011100000000000 (1616179247104) |
| \*\*\* ROUND 6 \*\*\*  1. 1101001111001010101101110010100000 xor 111101101001010110111000000001010000000000000000000 xor 10111100001001011110010100011100000000000 = 2170258866103456 = 111101101011101011011000100110010001110010010100000  2. 11001111010110100111011011010010000000 + 2170258866103456 mod 2^32 = 2607192352 = 10011011011001101001100100100000  3. 2607192352 + 5 mod 2^32 = 2607192357 = 10011011011001101001100100100101  4. 2607192357 <<< 7 = 333720621696 = 100110110110011010011001001001010000000  5. A = 10111100001001011110010100011100000000000 (1616179247104)  B = 100110110110011010011001001001010000000 (333720621696)  C = 1101001111001010101101110010100000 (14213110944)  D = 111101101001010110111000000001010000000000000000000 (2168983183163392) |
| \*\*\* ROUND 7 \*\*\*  1. 100110110110011010011001001001010000000 xor 1101001111001010101101110010100000 xor 111101101001010110111000000001010000000000000000000 = 2169212899315232 = 111101101001110001100111100010011100100111000100000  2. 10111100001001011110010100011100000000000 + 2169212899315232 mod 2^32 = 2283308576 = 10001000000110001000011000100000  3. 2283308576 + 6 mod 2^32 = 2283308582 = 10001000000110001000011000100110  4. 2283308582 <<< 11 = 4676215975936 = 1000100000011000100001100010011000000000000  5. A = 111101101001010110111000000001010000000000000000000 (2168983183163392)  B = 1000100000011000100001100010011000000000000 (4676215975936)  C = 100110110110011010011001001001010000000 (333720621696)  D = 1101001111001010101101110010100000 (14213110944) |
| \*\*\* ROUND 8 \*\*\*  1. 1000100000011000100001100010011000000000000 xor 100110110110011010011001001001010000000 xor 1101001111001010101101110010100000 = 4459121311264 = 1000000111000111000010101110111111000100000  2. 111101101001010110111000000001010000000000000000000 + 4459121311264 mod 2^32 = 4169104928 = 11111000011111110111111000100000  3. 4169104928 + 7 mod 2^32 = 4169104935 = 11111000011111110111111000100111  4. 4169104935 <<< 19 = 2185811688161280 = 111110000111111101111110001001110000000000000000000  5. A = 1101001111001010101101110010100000 (14213110944)  B = 111110000111111101111110001001110000000000000000000 (2185811688161280)  C = 1000100000011000100001100010011000000000000 (4676215975936)  D = 100110110110011010011001001001010000000 (333720621696) |
| \*\*\* ROUND 9 \*\*\*  1. 111110000111111101111110001001110000000000000000000 xor 1000100000011000100001100010011000000000000 xor 100110110110011010011001001001010000000 = 2190186465567360 = 111110001111111011010000110010001011010001010000000  2. 1101001111001010101101110010100000 + 2190186465567360 mod 2^32 = 3580919584 = 11010101011100000111111100100000  3. 3580919584 + 8 mod 2^32 = 3580919592 = 11010101011100000111111100101000  4. 3580919592 <<< 3 = 28647356736 = 11010101011100000111111100101000000  5. A = 100110110110011010011001001001010000000 (333720621696)  B = 11010101011100000111111100101000000 (28647356736)  C = 111110000111111101111110001001110000000000000000000 (2185811688161280)  D = 1000100000011000100001100010011000000000000 (4676215975936) |
| \*\*\* ROUND 10 \*\*\*  1. 11010101011100000111111100101000000 xor 111110000111111101111110001001110000000000000000000 xor 1000100000011000100001100010011000000000000 = 2189942059616576 = 111110001111011110110011110100010101100100101000000  2. 100110110110011010011001001001010000000 + 2189942059616576 mod 2^32 = 1373068224 = 1010001110101110101101111000000  3. 1373068224 + 9 mod 2^32 = 1373068233 = 1010001110101110101101111001001  4. 1373068233 <<< 7 = 175752733824 = 10100011101011101011011110010010000000  5. A = 1000100000011000100001100010011000000000000 (4676215975936)  B = 10100011101011101011011110010010000000 (175752733824)  C = 11010101011100000111111100101000000 (28647356736)  D = 111110000111111101111110001001110000000000000000000 (2185811688161280) |
| \*\*\* ROUND 11 \*\*\*  1. 10100011101011101011011110010010000000 xor 11010101011100000111111100101000000 xor 111110000111111101111110001001110000000000000000000 = 2185647403441600 = 111110000111101010110110001000101100001110111000000  2. 1000100000011000100001100010011000000000000 + 2185647403441600 mod 2^32 = 1967607232 = 1110101010001110100110111000000  3. 1967607232 + 10 mod 2^32 = 1967607242 = 1110101010001110100110111001010  4. 1967607242 <<< 11 = 4029659631616 = 111010101000111010011011100101000000000000  5. A = 111110000111111101111110001001110000000000000000000 (2185811688161280)  B = 111010101000111010011011100101000000000000 (4029659631616)  C = 10100011101011101011011110010010000000 (175752733824)  D = 11010101011100000111111100101000000 (28647356736) |
| \*\*\* ROUND 12 \*\*\*  1. 111010101000111010011011100101000000000000 xor 10100011101011101011011110010010000000 xor 11010101011100000111111100101000000 = 3867521600960 = 111000010001111010010000000100110111000000  2. 111110000111111101111110001001110000000000000000000 + 3867521600960 mod 2^32 = 1803046336 = 1101011011110000100110111000000  3. 1803046336 + 11 mod 2^32 = 1803046347 = 1101011011110000100110111001011  4. 1803046347 <<< 19 = 945315563175936 = 11010110111100001001101110010110000000000000000000  5. A = 11010101011100000111111100101000000 (28647356736)  B = 11010110111100001001101110010110000000000000000000 (945315563175936)  C = 111010101000111010011011100101000000000000 (4029659631616)  D = 10100011101011101011011110010010000000 (175752733824) B=>C=>01001110 110101011 1101010 11100001  C=>D=>10100010 11100111 01001000 01110010 |
| \*\*\* ROUND 13 \*\*\*  1. 11010110111100001001101110010110000000000000000000 xor 111010101000111010011011100101000000000000 xor 10100011101011101011011110010010000000 = 941460045935744 = 11010110000100000010111111100110111011010010000000  2. 11010101011100000111111100101000000 + 941460045935744 mod 2^32 = 1797238208 = 1101011000111111010110111000000  3. 1797238208 + 12 mod 2^32 = 1797238220 = 1101011000111111010110111001100  4. 1797238220 <<< 3 = 14377905760 = 1101011000111111010110111001100000  5. A = 10100011101011101011011110010010000000 (175752733824)  B = 1101011000111111010110111001100000 (14377905760)  C = 11010110111100001001101110010110000000000000000000 (945315563175936)  D = 111010101000111010011011100101000000000000 (4029659631616) |
| \*\*\* ROUND 14 \*\*\*  1. 1101011000111111010110111001100000 xor 11010110111100001001101110010110000000000000000000 xor 111010101000111010011011100101000000000000 = 941641729523296 = 11010110000110101100001100110010110011111001100000  2. 10100011101011101011011110010010000000 + 941641729523296 mod 2^32 = 4168688352 = 11111000011110010010001011100000  3. 4168688352 + 13 mod 2^32 = 4168688365 = 11111000011110010010001011101101  4. 4168688365 <<< 7 = 533592110720 = 111110000111100100100010111011010000000  5. A = 111010101000111010011011100101000000000000 (4029659631616)  B = 111110000111100100100010111011010000000 (533592110720)  C = 1101011000111111010110111001100000 (14377905760)  D = 11010110111100001001101110010110000000000000000000 (945315563175936) |
| \*\*\* ROUND 15 \*\*\*  1. 111110000111100100100010111011010000000 xor 1101011000111111010110111001100000 xor 11010110111100001001101110010110000000000000000000 = 945292408264928 = 11010110111011110100001010001101000001100011100000  2. 111010101000111010011011100101000000000000 + 945292408264928 mod 2^32 = 1151494368 = 1000100101000100110100011100000  3. 1151494368 + 14 mod 2^32 = 1151494382 = 1000100101000100110100011101110  4. 1151494382 <<< 11 = 2358260494336 = 100010010100010011010001110111000000000000  5. A = 11010110111100001001101110010110000000000000000000 (945315563175936)  B = 100010010100010011010001110111000000000000 (2358260494336)  C = 111110000111100100100010111011010000000 (533592110720)  D = 1101011000111111010110111001100000 (14377905760) |
| \*\*\* ROUND 16 \*\*\*  1. 100010010100010011010001110111000000000000 xor 111110000111100100100010111011010000000 xor 1101011000111111010110111001100000 = 2587569645792 = 100101101001110111001010110110100011100000  2. 11010110111100001001101110010110000000000000000000 + 2587569645792 mod 2^32 = 3850594528 = 11100101100000110110100011100000  3. 3850594528 + 15 mod 2^32 = 3850594543 = 11100101100000110110100011101111  4. 3850594543 <<< 19 = 2018820511760384 = 111001011000001101101000111011110000000000000000000  5. A = 1101011000111111010110111001100000 (14377905760)  B = 111001011000001101101000111011110000000000000000000 (2018820511760384)  C = 100010010100010011010001110111000000000000 (2358260494336)  D = 111110000111100100100010111011010000000 (533592110720) |

Rezultat:

"VCElectronicaDig" encrypted = 1101011000111111010110111001100000111001011000001101101000111011110000000000000000000100010010100010011010001110111000000000000111110000111100100100010111011010000000=

"Ö~nÁà(4î |yE´"

**CODE**

package Vigulear.Attestation2;  
  
import java.math.\*;  
  
public class MD4 {  
  
 public static String stringToBinary (String message) {  
 byte[] bytes = message.getBytes();  
 StringBuilder binary = new StringBuilder();  
 for (byte b : bytes)  
 {  
 int val = b;  
 for (int i = 0; i < 8; i++)  
 {  
 binary.append((val & 128) == 0 ? 0 : 1);  
 val <<= 1;  
 }  
 //binary.append(' ');  
 }  
 return binary.toString();  
 }  
  
 public static String binaryToAsciiToText(String message) {  
 StringBuilder s2 = new StringBuilder();  
  
 for (int i = 0; i < message.length()/8; i++) {  
  
 int a = Integer.*parseInt*(message.substring(8\*i,(i+1)\*8),2);  
 s2.append((char)(a));  
 }  
 return s2.toString();  
}

public static void main(String[] args) {  
 String m = "MDaElectronicaDig";  
 System.*out*.println("Message: " + m);  
  
 String stringInBinary = *stringToBinary*(m);  
  
 String a = "";  
 BigInteger aInt;  
 String b = "";  
 BigInteger bInt;  
 String c = "";  
 BigInteger cInt;  
 String d = "";  
 BigInteger dInt;  
  
 // setting A, B, C, D  
 int i = 0;  
 while (i < stringInBinary.length()) {  
 a = stringInBinary.substring(i, i+32);  
 i += 32;  
 b = stringInBinary.substring(i, i+32);  
 i += 32;  
 c = stringInBinary.substring(i, i+32);  
 i += 32;  
 d = stringInBinary.substring(i, i+32);  
 i += 32;  
 }  
  
 // cast A, B, C, and D to BigInteger  
 aInt = BigInteger.*valueOf*(Integer.*parseInt*(a, 2));  
 bInt = BigInteger.*valueOf*(Integer.*parseInt*(b, 2));  
 cInt = BigInteger.*valueOf*(Integer.*parseInt*(c, 2));  
 dInt = BigInteger.*valueOf*(Integer.*parseInt*(d, 2));  
  
 // BigInteger for 2^32  
 BigInteger modulus = BigInteger.*valueOf*(2).pow(32);  
  
 int[] shifts = {3, 7, 11, 19};  
  
 // Algorithm  
 for (int mi = 0; mi < 16; mi++) {  
  
  
 BigInteger p1 = bInt.xor(cInt).xor(dInt);  
 System.*out*.println("\n\*\*\* ROUND " + (mi + 1) +" \*\*\*\n1. " + b + " xor " + c + " xor " + d + " = " + p1 + " = " + p1.toString(2));  
  
 BigInteger p2 = p1.add(aInt).mod(modulus);  
 System.*out*.println("2. " + a + " + " + p1 + " mod 2^32 = " + p2 + " = " + p2.toString(2));  
  
 BigInteger p3 = p2.add(BigInteger.*valueOf*(mi)).mod(modulus);  
 System.*out*.println("3. " + p2 + " + " + mi + " mod 2^32 = " + p3 + " = " + p3.toString(2));  
  
 BigInteger p4 = p3.shiftLeft(shifts[mi%4]);  
 System.*out*.println("4. " + p3 + " <<< " + shifts[mi%4] + " = " + p4 + " = " + p4.toString(2));  
  
 a = d;  
 d = c;  
 c = b;  
 b = p4.toString(2);  
  
 aInt = dInt;  
 dInt = cInt;  
 cInt = bInt;  
 bInt = p4;  
  
 System.*out*.println("5. A = " + a + " (" + aInt + ") " + "\n\tB = " + b + " (" + bInt + ") "+  
 "\n\tC = " + c + " (" + cInt + ") " + "\n\tD = " + d + " (" + dInt + ") ");  
 }  
 System.*out*.println("\"" + m + "\" encrypted = \"" + *binaryToAsciiToText*(a.concat(b).concat(c).concat(d)) + "\"");  
 }  
}

**Concluzii:**

În acesta lucrare am utilizat algoritmul MD4 pentru a cripta un mesaj din 16 simboluri. Mesajul inițial de criptare a fost luat numele și prenumele unei persoane în cazul în care nu ajung simboluri se adauga unele simboluri la dorinta. Mesaj initial ,, VC**ElectronicaDig**’’.

În urma efectuării operațiilor de criptare la al 16 pas am primit mesajul criptat:

1101011000111111010110111001100000111001011000001101101000111011110000000000000000000100010010100010011010001110111000000000000111110000111100100100010111011010000000

Result:

**"VCElectronicaDig" encrypted = "Ö?[9`Ú;À J&àðòEÚ"**