Jurnal Riset Mahasiswa Matematika
Volume 3(1) (2023), Pages 10-25
p-ISSN: 2808-1552; e-ISSN: 2808-4926

Kriptografi Hibrida Cipher Block Chaining (CBC) dan Merkle-Hellman Knapsack untuk Pengamanan Pesan Teks

Chofifah Alfin Novianti*, Muhammad Khudzaifah, Muhammad Nafie Jauhari

Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang, Indonesia

chofifahalfin@gmail.com, khudzaifah@uin-malang.ac.id, nafie.jauhari@uin-malang.ac.id

Abstrak

Pesan rahasia merupakan pesan yang hanya boleh dilihat oleh orang yang berhak. Dalam pengirimannya diperlukan suatu prosedur agar pesan rahasia dapat terjaga yang disebut sebagai kriptografi. Pada penelitian ini menggunakan kriptografi hibrida *Cipher Block Chaining* (*CBC*) dan *Merkle-Hellman Knapsack*. Tujuan penelitian ini yaitu untuk mengetahui proses enkripsi dan dekripsi dari kriptografi hibrida *Cipher Block Chaining* (*CBC*) dan *Merkle-Hellman Knapsack*. Tahapan dalam penelitian ini menggunakan pendekatan kualitatif dengan metode kajian Pustaka. Pada proses enkripsi dengan *CBC*, plainteks dienkripsi terlebih dahulu dan menghasilkan cipherteks. Selanjutnya kunci dan *Initialization Vector* (*IV*) dari *CBC* dienkripsi menggunakan *Merkle-Hellman Knapsack* dengan membangkitkan kunci publik terlebih dahulu dan menghasilkan *cipherkey.* Kebalikan dari enkripsi, pada proses dekripsi *cipherkey* terlebih dahulu didekripsi menggunakan *Merkle-Hellman Knapsack* dengan menghitung balikan modulo $n \bmod m$. Proses dekripsi dilanjutkan dengan mendekripsi cipherteks menggunakan *CBC.* Hasil dari pesan yang diamankan menggunakan kriptografi hibrida *Cipher Block Chaining* (*CBC*) dan *Merkle-Hellman Knapsack* memiliki tingkat keamanan lebih tinggi dibandingkan dengan hanya menggunakan satu algoritma kriptografi saja. Adapun untuk kedepannya, penelitian ini dapat digunakan untuk memperluas pengetahuan mengenai pengamanan pesan teks menggunakan kriptografi hibrida algoritma *CBC* dan *Merkle-Hellman Knapsack*.

Kata kunci: Cipher Block Chaining (CBC); Merkle-Hellman Knapsack; Kriptografi

Abstract

A secret message is a message that can only be seen by those who are entitled. In its delivery, a procedure is needed to keep the secret message secure, which is called cryptography. This research uses hybrid cryptography Cipher Block Chaining (CBC) and Merkle-Hellman Knapsack. The purpose of this research is to find out the encryption and decryption process of hybrid cryptography Cipher Block Chaining (CBC) and Merkle-Hellman Knapsack. The stages in this research use a qualitative approach with the library research method. In the encryption process with CBC, plaintext is encrypted first and the result called ciphertext. Furthermore, the key and Initialization Vector (IV) from CBC are encrypted using Merkle-Hellman Knapsack by generating a public key first and producing cipherkey. In the decryption process cipherkey is first decrypted using Merkle-Hellman Knapsack by calculating the inverse modulo $n \bmod m$. The decryption process continues by decrypting ciphertext using CBC. The result of securing messages using hybrid cryptography Cipher Block Chaining (CBC) and Merkle-Hellman Knapsack has a higher level of security than using only one cryptographic algorithm. As for the future, this research can be used to expand knowledge about securing text messages using hybrid cryptography algorithm CBC and Merkle-Hellman Knapsack.

Keywords: Cipher Block Chaining (CBC); Cryptography; Merkle-Hellman Knapsack

PENDAHULUAN

Kriptografi adalah ilmu yang menekuni metode-metode matematika yang bersangkutan dengan keamanan pesan [1]-[3]. Pesan yang akan dikirimkan terlebih dahulu disandikan oleh pengirim pesan. Terdapat dua proses yang digunakan untuk menyandikan pesan, yaitu enkripsi

Submitted: 8 Februari 2023 Reviewed: 20 Agustus 2023 Accepted: 21 September 2023
Doi: http://dx.doi.org/10.18860/jrmm.v3i1.22292

dan dekripsi pesan. Enkripsi adalah proses merubah pesan asli (plainteks) menjadi pesan yang tidak dapat dipahami (cipherteks). Sedangkan, dekripsi merupakan kebalikan dari enkripsi, yaitu pesan yang telah dienkripsi akan dikembalikan ke bentuk asal pesan (plainteks). Berdasarkan kuncinya, kriptografi dibedakan menjadi dua jenis yaitu, kriptografi simetris dan asimetris. Kriptografi simetris adalah kriptografi yang menggunakan satu kunci yang sama pada enkripsi dan dekripsinya. Sedangkan kriptografi asimetris berarti terdapat dua kunci yang berbeda saat proses enkripsi dan dekripsinya [4]-[5].

Salah satu algoritma yang termasuk kriptografi simetris adalah algoritma *Cipher Block Chaining* (*CBC*) selain itu, *CBC* juga termasuk dalam kriptografi modern. Algoritma ini menggunakan blok-blok bit sebagai plainteksya dan menerapkan mekanisme umpan-balik pada tiap bloknya [6]. Pada pengoperasian kriptografi modern, plainteks perlu dikonversi menjadi suatu digit biner, 0 dan 1. Schema encoding yang umum digunakan adalah ASCII (*American Standart Code for Information Interchange*). ASCII merupakan tabel karakter *printable* maupun *non-printable* [7]. Urutan bit yang akan mewakili plainteks kemudian dienkripsi untuk mendapatkan *ciphertext* dalam bentuk urutan bit. Binari string merupakan bagian operasi algoritma ini, maka pemahaman terhadap metode kombinasi dua bit yang disebut *Exclusive OR* diperlukan. *Exclusive OR* dinotasikan dengan $\oplus$. Operasi XOR mengkombinasikan dua bit string dengan panjang yang sama. Dengan penambahan modulo 2 dan digambarkan sebagai berikut $0 \oplus 0 = 0,\ 0 \oplus 1 = 1,\ 1 \oplus 0 = 1,\ 1 \oplus 1 = 0$ [8]-[9]. Penelitian yang telah dibahas pada [8], membahas mengenai algoritma *Cipher Block Chaining* (*CBC*) digunakan untuk proses enkripsi dan dekripsi pada pengamanan data, seperti bentuk pesan teks, dokumen, dan gambar, karena dalam penerapannya digunakan digit biner, sehingga pada proses enkripsi pesan tidak dapat terbaca serta waktu yang diperlukan cukup singkat.

Sedangkan *Merkle-Hellman Knapsack* termasuk dalam kriptografi asimetris. Algoritma ini merupakan modifikasi dari algoritma *Knapsack* yang ditemukan oleh Martin Hellman dan Ralph Merkle pada tahun 1978 [10]. Mekanisme pengerjaan *Merkle-Hellman Knapsack* adalah dengan menentukan *key generation* dari barisan *superincreasing.* Barisan *superincreasring* merupakan barisan di mana setiap nilai di dalam barisan lebih besar daripada jumlah semua nilai sebelumnya. Algoritma *Merkle-Hellman Knapsack* dilakukan dengan perhitungan aritmatika modulo [11]-[13]. Aritmatika modulo menghasilkan hasil sisa pembagian [14]. Pada penelitian sebelumnya [15], algoritma *Merkle-Hellman Knapsack* digunakan untuk proses dekripsi dan enkripsi teks. Penggunaan algoritma ini dinilai memiliki ukuran kunci yang lebih kecil dibandingkan dengan algoritma *Rivest Shamir Adleman* (RSA) dan memiliki kemampuan keamanan yang kuat dengan panjang kunci yang pendek.

METODE

Penelitian ini menggunakan jenis penelitian kualitatif atau studi literatur. Terdapat dua proses yang digunakan untuk penyandian pesan, yaitu enkripsi dan dekripsi. Langkah-langkah yang digunakan pada proses enkripsi yaitu:

1. Membangkitkan kunci *Merkle-Hellman Knapsack* dengan menentukan nilai barisan *superincreasing* (w), n, dan m secara acak. Dimana nilai $FPB(n, m) = 1$. Pembangkitan kunci publik menggunakan persamaan $\beta_i = w_i \cdot n \bmod m$. Barisan *superincreasing* akan menjadi kunci privatnya.
2. Menentukan pesan awal (plainteks), kunci dan *Initialization Vector* (IV) dalam bentuk karakter huruf.
3. Mengubah plainteks, kunci dan IV menjadi biner
4. Melakukan perhitungan terhadap plainteks menggunakan CBC dengan persamaan $C_i = E_K(P_i \oplus C_{i-1})$.
5. Mendapatkan cipherteks dari proses enkripsi CBC yang berupa heksadesimal.
6. Enkripsi kunci dan IV dari CBC menggunakan kunci publik *Merke-Hellman Knapsack*, sehingga didapatkan *cipherkey*.

Sedangkan langkah-langkah proses dekripsi yaitu:

1. Dekripsi *cipherkey* menggunakan *Merkle-Hellman Knapsack* dilakukan dengan menggunakan kunci privat.
2. Menghitung balikan n modulo m, sedemikian sehingga $n \cdot n^{-1} \equiv 1 \ (mod\ m)$
3. Kalikan setiap kriptogram dengan $n^{-1}\ mod\ m$, lalu nyatakan hasil kalinya sebagai penjumlahan elemen-elemen kunci privat untuk memperoleh plainteks dengan menggunakan algoritma pencarian solusi *superincreasing* knapsack,
4. Dihasilkan plainteks kunci dan IV CBC.
5. Dekripsi cipherteks menggunakan CBC dengan mengubah cipherteks menjadi bentuk biner.
6. Dekripsi dengan persamaan: $P_i = D_k(C_i) \oplus C_{i-1}$
7. Mendapatkan plainteks dalam bentuk biner yang selanjutnya diubah menjadi huruf alfabet.
8. Didapatkan kembali pesan awal (plainteks).

HASIL DAN PEMBAHASAN

A. Berikut adalah simulasi proses enkripsi pesan teks menggunakan CBC dan *Merkle-Hellman Knapsack*:

1. Pembangkitan kunci dilakukan dengan menentukan nilai barisan *superincreasing* (w), n, dan m secara acak. Di mana nilai $FPB(n, m) = 1$. Pada penelitian ini digunakan $w = \{2, 3, 6, 15,\ 27, 78, 131, 304\}$, $n = 97$, dan $m = 575$.
2. Melakukan pembangkitan kunci dengan persamaan $\beta_i = w_i \cdot n\ mod\ m$. Hasil dari perkalian akan menjadi kunci publik dan barisan *superincreasing* awal akan menjadi kunci privat.

   $\beta_i = w_i \cdot n\ mod\ m$

   β_1 : $2 \cdot 97\ mod\ 575 = 194$
   β_2 : $3 \cdot 97\ mod\ 575 = 291$
   β_3 : $6 \cdot 97\ mod\ 575 = 7$
   β_4 : $15 \cdot 97\ mod\ 575 = 305$
   β_5 : $27 \cdot 97\ mod\ 575 = 319$
   β_6 : $78 \cdot 97\ mod\ 575 = 91$
   β_7 : $131 \cdot 97\ mod\ 575 = 57$
   β_8 : $304 \cdot 97\ mod\ 575 = 163$

   Jadi, kunci publiknya adalah $\{194, 291, 7, 305, 319, 91, 57, 163\}$, sedangkan kunci privatnya adalah $\{2, 3, 6, 15,\ 27, 78, 131, 304\}$.
3. Menentukan karakter alfabet yang akan digunakan sebagai plainteks. Pada penelitian ini, digunakan kalimat "JaLaN#GAjAyana#50" sebagai plainteksnya.

Tabel 1 Hasil Perubahan Karakter Plainteks Menjadi Biner Menggunakan Tabel ASCII

Plainteks	Karakter	Biner
P_1	J	01001010
P_2	a	01100001
P_3	L	01001100
P_4	a	01100001
P_5	N	01001110

P_6	#	00100011
P_7	G	01000111
P_8	A	01000001
P_9	j	01101010
P_{10}	A	01000001
P_{11}	y	01111001
P_{12}	a	01100001
P_{13}	n	01101110
P_{14}	a	01100001
P_{15}	#	00100011
P_{16}	5	00110101
P_{17}	0	00110000

4. Menentukan kunci CBC dan *Initialization Vector* yang akan menjadi C_0 secara acak.

Tabel 2 Hasil Perubahan IV dan Kunci Menjadi Biner Menggunakan Tabel ASCII

Jenis	Karakter	Biner
Initialization Vector (C_0)	P	01010000
Kunci (K)	R	01010010

5. Melakukan proses enkripsi plainteks dengan CBC. Kunci dan *Initialization Vector* akan digunakan dalan proses ini. Adapun proses enkripsi sebagai berikut:
$C_i = E_K(P_i \oplus C_{i-1})$

P_1 : 01001010

IV : 01010000 $\oplus$

00011010

K 01010010 $\oplus$

01001000

Geser 1-bit ke kiri: $C_1 = 10010000$

P_2 : 01100001

C_1 : 10010000 $\oplus$

11110001

K 01010010 $\oplus$

10100011

Geser 1-bit ke kiri: $C_2 = 01000111$

P_3 : 01001100

C_2 : 01000111 $\oplus$

00001011

K 01010010 $\oplus$

01011001

Geser 1-bit ke kiri: $C_3 = 10110010$

P_4 : 01100001

C_3 : 10110010 $\oplus$

11010011

K 01010010 $\oplus$

10000001

Geser 1-bit ke kiri: $C_4 = 00000011$

P_5 : 01001110

C_4 : 00000011 $\oplus$

01001101

K 01010010 $\oplus$

00011111

Geser 1-bit ke kiri: $C_5 = 00111110$

P_6 : 00100011

C_5 : 00111110 $\oplus$

00011101

K 01010010 $\oplus$

01001111

Geser 1-bit ke kiri: $C_6 = 10011110$

P_7 : 01000111

C_6 : 10011110 $\oplus$

11011001

K $\quad 01010010 \quad \oplus$

10001011

Geser 1-bit ke kiri: $C_7 = 00010111$

P_8 : 01000001

C_7 : $00010111 \quad \oplus$

01010110

K $\quad 01010010 \quad \oplus$

00000100

Geser 1-bit ke kiri: $C_8 = 00001000$

P_9 : 01101010

C_8 : $00001000 \quad \oplus$

01100010

K $\quad 01010010 \quad \oplus$

00110000

Geser 1-bit ke kiri: $C_9 = 01100000$

P_{10} : 01000001

C_9 : $01100000 \quad \oplus$

00100001

K $\quad 01010010 \quad \oplus$

01110011

Geser 1-bit ke kiri: $C_{10} = 11100110$

P_{11} : 01111001

C_{10} : $11100110 \quad \oplus$

10011111

K $\quad 01010010 \quad \oplus$

11001101

Geser 1-bit ke kiri: $C_{11} = 10011011$

P_{12} : 01100001

C_{11} : $10011011 \quad \oplus$

$$\begin{array}{lll} & 11111010 & \\ \text{K} & 01010010 & \oplus \\ \hline & 10101000 & \end{array}$$

Geser 1-bit ke kiri: $C_{12} = 01010001$

$$\begin{array}{lll} P_{13} : & 01101110 & \\ C_{12} : & 01010001 & \oplus \\ \hline & 00111111 & \\ \text{K} & 01010010 & \oplus \\ \hline & 01101101 & \end{array}$$

Geser 1-bit ke kiri: $C_{13} = 11011010$

$$\begin{array}{lll} P_{14} : & 01100001 & \\ C_{13} : & 11011010 & \oplus \\ \hline & 10111011 & \\ \text{K} & 01010010 & \oplus \\ \hline & 11101001 & \end{array}$$

Geser 1-bit ke kiri: $C_{14} = 11010011$

$$\begin{array}{lll} P_{15} : & 00100011 & \\ C_{14} : & 11010011 & \oplus \\ \hline & 11110000 & \\ \text{K} & 01010010 & \oplus \\ \hline & 10100010 & \end{array}$$

Geser 1-bit ke kiri: $C_{15} = 01000101$

$$\begin{array}{lll} P_{16} : & 00110101 & \\ C_{15} : & 01000101 & \oplus \\ \hline & 01110000 & \\ \text{K} & 01010010 & \oplus \\ \hline & 00100010 & \end{array}$$

Geser 1-bit ke kiri: $C_{16} = 01000100$

P_{17} : 00110000

C_{16} : 01000100 $\oplus$

01110100

K 01010010 $\oplus$

00100110

Geser 1-bit ke kiri: $C_{17} = 01001100$

6. Ubah hasil enkripsi pesan menjadi heksadesimal sehingga menghasilkan cipherteks yang akan diterima oleh penerima pesan.

Tabel 3 Hasil Enkripsi Plainteks Perubahan dari Biner ke Heksadesimal Menggunakan Tabel ASCII

	Biner	**Heksadesimal**
C_1	10010000	90
C_2	01000111	47
C_3	10110010	B2
C_4	00000011	03
C_5	00111110	3E
C_6	10011110	9E
C_7	00010111	17
C_8	00001000	08
C_9	01100000	60
C_{10}	11100110	E6
C_{11}	10011011	B5
C_{12}	01010001	1D
C_{13}	11011010	AD
C_{14}	11010011	34
C_{15}	01000101	54
C_{16}	01000100	44
C_{17}	01001100	4C

Hasil enkripsi plainteks adalah:
9047B2033E9E170860E69B51DAD345444C
Penggunaan heksadesimal disebabkan karena penulisan lebih praktis, mudah dibaca, dan memiliki kemungkinan timbul kesalahan yang lebih kecil. Selain itu, tidak digunakannya karakter ASCII sebagai hasil akhir cipherteks karena tidak semua simbol karakter yang digunakan dapat dituliskan secara jelas.

7. Selanjutnya, akan dilakukan proses enkripsi kunci CBC dan *Initialization Vector* (IV) menggunakan *Merkle-Hellman Knapsack*. Hasil enkripsi disebut sebagai *cipherkey*. Berikut plainteks yang akan digunakan:

Tabel 4 Plainteks IV dan Kunci CBC

			Plainteks
1.	*Initialization Vector*	:	01010000
2.	Kunci CBC	:	01010010

8. Plainteks dipecah menjadi blok bit yang panjangnya sama dengan barisan kunci publik. Mengalikan tiap bit di dalam blok dengan elemen di kunci publik.

   Plainteks 1 (IV) : 01010000

   Kunci publik : 194, 291, 7, 305, 319, 91, 57, 163

   Kriptogram : $(0 \times 194) + (1 \times 291) + (0 \times 7) + (1 \times 305) + (0 \times 319) + (0 \times 91) + (0 \times 57) + (0 \times 163) = 0 + 291 + 0 + 305 + 0 + 0 + 0 + 0 = 596$

   Plainteks 2 (Kunci) : 01010010

   Kunci publik : 194, 291, 7, 305, 319, 91, 57, 163

   Kriptogram : $(0 \times 194) + (1 \times 291) + (0 \times 7) + (1 \times 305) + (0 \times 319) + (0 \times 91) + (1 \times 57) + (0 \times 163) = 0 + 291 + 0 + 305 + 0 + 0 + 57 + 0 = 653$

9. Mendapatkan *cipherkey* untuk IV yaitu 596 dan kunci, yaitu 653.

B. Berikut adalah simulasi proses dekripsi pesan teks menggunakan CBC dan *Merkle-Hellman Knapsack*:

1. Melakukan dekripsi *cipherkey* menggunakan kunci publik dengan *Merkle-Hellman Knapsack*. Mula-mula hitung n^{-1}, yaitu balikan dari $n \bmod m$. Akan digunakan persamaan:

   $n \cdot n^{-1} \equiv 1(mod\ m)$
   $97 \cdot n^{-1} \equiv 1(mod\ 575)$
   $n^{-1} \equiv \frac{1+575k}{97}, k = 0, 1, 2, 3, \ldots..$
   $n^{-1} = \frac{1+14 \cdot 575}{97}$
   $n^{-1} = 83$

2. Mengalikan setiap kriptogram dengan $n^{-1} \bmod m$, lalu menyatakan hasil kalinya sebagai penjumlahan elemen-elemen kunci privat untuk memperoleh kunci dan IV asli menggunakan algoritma pencarian solusi *superincreasing knapsack*.

   Cipherkey 1 : 596

   (IV) $= 596 \cdot 83\ mod\ 575 = 18$

   Bandingkan hasil dari perhitungan *cipherkey* 1 dengan bobot koresponden dimulai dari terbesar ke terkecil

   Koresponden : 2, 3, 6, 15, 27, 78, 131, 304

   1. Bandingkan 18 dengan 304. Karena $304 > 18$, maka bernilai 0
   2. Bandingkan 18 dengan 131. Karena $131 > 18$, maka bernilai 0
   3. Bandingkan 18 dengan 78. Karena $78 > 18$, maka bernilai 0
   4. Bandingkan 18 dengan 27. Karena $27 > 18$, maka bernilai 0
   5. Bandingkan 18 dengan 15. Karena $15 \leq 18$, maka bernilai 1

6. Bobot total sekarang menjadi $18 - 15 = 3$. Bandingkan 3 dengan 6. Karena $6 > 3$, maka bernilai 0
7. Bandingkan 3 dengan 3. Karena $3 \leq 3$, maka bernilai 1
8. Bobot total sekarang menjadi $3 - 3 = 0$. Bandingkan 0 dengan 2. Karena $2 > 0$, maka bernilai 0

Sehingga dihasilkan biner = 01010000

Cipherkey 2 : 653

(Kunci) $= 653 \cdot 83 \; mod \; 575 = 149$

Bandingkan hasil dari perhitungan *cipherkey* 2 dengan bobot koresponden dimulai dari yang terbesar ke terkecil

Koresponden : 2, 3, 6, 15, 27, 78, 131, 304

1. Bandingkan 149 dengan 304. Karena $304 > 149$, maka bernilai 0
2. Bandingkan 149 dengan 131. Karena $131 \leq 149$, maka bernilai 1
3. Bobot total sekarang menjadi $149 - 131 = 18$. Bandingkan 18 dengan 78. Karena $78 > 18$, maka bernilai 0
4. Bandingkan 18 dengan 27. Karena $27 > 18$, maka bernilai 0
5. Bandingkan 18 dengan 15. Karena $15 \leq 18$, maka bernilai 1
6. Bobot total sekarang menjadi $18 - 15 = 3$. Bandingkan 3 dengan 6. Karena $6 > 3$, maka bernilai 0
7. Bandingkan 3 dengan 3. Karena $3 \leq 3$, maka bernilai 1
8. Bobot total sekarang menjadi $3 - 3 = 0$. Bandingkan 0 dengan 2. Karena $2 > 0$, maka bernilai 0

Sehingga didapatkan biner = 01010010

3. Didapatkan hasil dekripsi IV yaitu 01010000 dan kunci CBC yaitu 01010010.
4. Melakukan dekripsi cipherteks menggunakan CBC. Mula-mula ubah cipherteks yang berupa bilangan heksadesimal menjadi bilangan biner.

Tabel 5 Hasil Perubahan Cipherteks Menjadi Biner Menjadi Biner Menggunakan Tabel ASCII

	Heksadesimal	**Biner**
C_1	90	10010000
C_2	47	01000111
C_3	B2	10110010
C_4	03	00000011
C_5	3E	00111110
C_6	9E	10011110
C_7	17	00010111
C_8	08	00001000
C_9	60	01100000
C_{10}	E6	11100110
C_{11}	B5	10011011

C_{12}	1D	01010001
C_{13}	AD	11011010
C_{14}	34	11010011
C_{15}	54	01000101
C_{16}	44	01000100
C_{17}	4C	01001100

5. Dekripsi pesan dilakukan dengan persamaan:
   $P_i = D_k(C_i) \oplus C_{i-1}$

   C_1 : 10010000

   Geser 1-bit ke kanan: $C_1' = 01001000$

   C_1' : 01001000

   K : 01010010 $\oplus$

   00011010

   IV 01010000 $\oplus$

   P_1 : 01001010

   C_2 : 01000111

   Geser 1-bit ke kanan: $C_2' = 10100011$

   C_2' : 10100011

   K : 01010010 $\oplus$

   11110001

   C_1 10010000 $\oplus$

   P_2 : 01100001

   C_3 : 10110010

   Geser 1-bit ke kanan: $C_3' = 01011001$

   C_3' : 01011001

   K : 01010010 $\oplus$

   00001011

   C_2 01000111 $\oplus$

   P_3 : 01001100

   C_4 : 00000011

   Geser 1-bit ke kanan: $C_4' = 10000001$

   C_4' : 10000001

K : 01010010 ⊕
11010011
C_3 10110010 ⊕
P_4 : 01100001

C_5 : 00111110

Geser 1-bit ke kanan: $C_5' = 00011111$

C_5' : 00011111
K : 01010010 ⊕
01001101
C_4 00000011 ⊕
P_5 : 01001110

C_6 : 10011110

Geser 1-bit ke kanan: $C_6' = 01001111$

C_6' : 01001111
K : 01010010 ⊕
00011101
C_5 00111110 ⊕
P_6 : 00100011
C_7 : 00010111

Geser 1-bit ke kanan: $C_7' = 10001011$

C_7' : 10001011
K : 01010010 ⊕
11011001
C_6 10011110 ⊕
P_7 : 01000111

C_8 : 00001000

Geser 1-bit ke kanan: $C_8' = 00000100$

C_8' : 00000100
K : 01010010 ⊕
01010110

C_7 00010111 $\oplus$

P_8 : 01000001

C_9 : 01100000

Geser 1-bit ke kanan: $C_9' = 00110000$

C_9' : 00110000

K : 01010010 $\oplus$

01100010

C_8 00001000 $\oplus$

P_9 : 01101010

C_{10} : 11100110

Geser 1-bit ke kanan: $C_{10}' = 01110011$

C_{10}' : 01110011

K : 01010010 $\oplus$

00100001

C_9 01100000 $\oplus$

P_{10} : 01000001

C_{11} : 10011011

Geser 1-bit ke kanan: $C_{11}' = 11001101$

C_{11}' : 11001101

K : 01010010 $\oplus$

10011111

C_{10} 11100110 $\oplus$

P_{11} : 01111001

C_{12} : 01010001

Geser 1-bit ke kanan: $C_{12}' = 10101000$

C_{12}' : 10101000

K : 01010010 $\oplus$

11111010

C_{11} 10011011 $\oplus$

P_{12} : 01100001

C_{13} : 11011010

Geser 1-bit ke kanan: $C'_{13} = 01101101$

$C_{13}{}'$: 01101101

K : 01010010 $\oplus$

00111111

C_{12} 01010001 $\oplus$

P_{13} : 01101110

C_{14} : 11010011

Geser 1-bit ke kanan: $C'_{14} = 11101001$

$C_{14}{}'$: 11101001

K : 01010010 $\oplus$

10111011

C_{13} 11011010 $\oplus$

P_{14} : 01100001

C_{15} : 01000101

Geser 1-bit ke kanan: $C'_{15} = 10100010$

$C_{15}{}'$: 10100010

K : 01010010 $\oplus$

11110000

C_{14} 11010011 $\oplus$

P_{15} : 00100011

C_{16} : 01000100

Geser 1-bit ke kanan: $C'_{16} = 00100010$

$C_{16}{}'$: 00100010

K : 01010010 $\oplus$

01110000

C_{15} 01000101 $\oplus$

P_{16} : 00110101

C_{17} : 01001100

Geser 1-bit ke kanan: $C_{17}' = 00100110$

C_{17}' : 00100110

K : 01010010 $\oplus$

01110100

C_{16} 01000100 $\oplus$

P_{17} : 00110000

Hasil dekripsi cipherteks adalah:
01001010 01100001 01001100 01100001 01001110 00100011 01000111 01000001 01101010 01000001 01111001 01100001 01101110 01100001 00100011 00110101 00110000

6. Mengubah biner menjadi alfabet menggunakan tabel ASCII

Tabel 6 Hasil dekripsi cipherteks menggunakan CBC

Plainteks	Biner	Karakter
P_1	01001010	J
P_2	01100001	a
P_3	01001100	L
P_4	01100001	a
P_5	01001110	N
P_6	00100011	#
P_7	01000111	G
P_8	01000001	A
P_9	01101010	j
P_{10}	01000001	A
P_{11}	01111001	y
P_{12}	01100001	a
P_{13}	01101110	n
P_{14}	01100001	a
P_{15}	00100011	#
P_{16}	00110101	5
P_{17}	00110000	0

7. Mendapatkan plainteks, yaitu “JaLaN#GAjAyana#50”.

KESIMPULAN

Berdasarkan pembahasan yang dilakukan di atas, maka didaapatkan kesimpulan sebagai berikut:

1. Terdapat dua proses enkripsi pesan menggunakan kriptografi hibrida CBC dan *Merkle-Hellman Knapsack*, yaitu enkripsi plainteks menggunakan CBC dan enkripsi IV dan kunci CBC menggunakan *Merkle-Hellman Knapsack*. Dalam proses enkripsi plainteks, digunakan persamaan $C_i = E_K(P_i \oplus C_{i-1})$. Selanjutnya pada proses enkripsi IV dan kunci CBC menggunakan M*erkle-Hellman Knapsack*. Pertama, menentukan barisan *superincreasing* yang nantinya akan menjadi kunci privat, nilai n, dan m secara acak. Melakukan pembangkitan kunci dengan persamaan $\beta_i = w_i \cdot n \bmod m$. Hasil dari perkalian akan

menjadi kunci publik. Kemudian, IV dan kunci CBC dipecah menjadi blok bit yang panjangnya sama dengan barisan kunci publik. Mengalikan tiap bit di dalam blok dengan elemen di kunci publik. Tahap ini akan menghasilkan *cipherkey* yang akan dikirim kepada penerima pesan.

2. Pada proses dekripsi, pertama-tama dilakukan dekripsi terhadap cipherkey dengan *Merkle-Hellman Knapsack* dalam tahap ini akan dilakukan pencarian n^{-1} menggunakan persamaan $n \cdot n^{-1} \equiv 1(mod\ m)$. Setelah mendapatkan n^{-1} masing-masing kriptogram akan dikalikan dengan persamaan $n^{-1}\ mod\ m$. Didapatkan kembali IV dan kunci CBC. Selanjutnya, dekripsi cipherteks menggunakan persamaan $P_i = D_k(C_i) \oplus C_{i-1}$. Dengan demikian, penerima pesan dapat mengetahui isi dari plainteks.

DAFTAR PUSTAKA

[1] Ariyus, D. (2008). *Pengantar Ilmu Kriptografi.* Yogyakarta: Penerbit ANDI.

[2] Munir, R. (2019). *Kriptografi.* Bandung: Informatika Bandung.

[3] Sadikin, A. (2012). *Kriptografi untuk Keamanan Jaringan.* Yogyakarta: Penerbit ANDI.

[4] Ariyus, D. (2006). *Kriptografi: Keamanan Data dan Komunikasi.* Yogyakarta: Graha Ilmu.

[5] Sukamto, R. A. (2010). *Algoritma dan Pemograman .* Bogor: Program Ilmu Komputer Universitas Pendidikan Indonesia.

[6] Rosmala, D., & Aprian, R. (2012). Implementasi Mode Operasi Cipher Block Chaining (CBC) pada Pengamanan Data. *Jurnal Infromatika*, 55-65.

[7] Budiasa, R. M. (2010). Analisis Kriptografi dalam penentuan Cipherteks kode ASCII melalui metode Aljabar Boolean. *MAKALAH IF3058 KRIPTOGRAFI*.

[8] Sugiartowo, & Ambo, S. N. (2018). SIMULASI RANGKAIAN KOMBINASIONAL SEBAGAI MEDIA PEMBELAJARAN SISTEM DIGITAL PADA FAKULTAS TEKNIK UNIVERSITAS MUHAMMADIYAH JAKARTA. *Seminar Nasional Sains dan Teknologi*, 1-11.

[9] Saputro, P. H. (2023). Implementasi Algoritma Exclusive OR (XOR) Dalam Pengembangan Aplikasi Chat Berbasis Android. *Informatika*, 71-76.

[10] Sulaiman, O. K. (2019). Hybrid Cryptosystem Menggunakan XOR Cipher dan Merkle-Hellman Knapsack untuk Menjaga Kerahasiaan Pesan Digital. *Teknologi Informasi*, 169-173.

[11] Hidayat, A., Akmal, & Rosyadi, R. (2016). Cryptography Asymmetris Merkle-Hellman Knapsack Digunakan untuk Enkripsi dan Dekripsi Teks. *Prosiding Seminar Nasional MIPA 2016*, 66-68.

[12] Hidayat, A., Rosyadi, R., & Paulus, E. (2016). Aplikasi Merkle-Hellman Knapsack untuk Kriptografi File Teks. *SENTER 2016: Seminar Nasional Teknik Elektro 2016*, 194-200.

[13] Munir, R. (2016). *Matematika Diskrit.* Bandung: Informatika Bandung.

[14] Irawan, W. H., Hijriyah, N., & Habibi, A. R. (2014). *Pengantar Teori Bilangan.* Malang: UIN Maliki Press.

[15] Sidik, A. P., & Mayasari, N. (2019). Rancangan Model Algoritma Hybrid Teknik Enkripsi XOR dengan Kombinasi Mode Block Cipher CBC - ECB 512-Bits dan Algoritma RSA. *Teknik dan Informatika*, 1-7.