PENERAPAN KODE REED MULLER PADA SISTEM KRIPTOGRAFI MCELIECE

Rizal Fahrur Rozi*[1], Putranto Hadi Utomo[2]
Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret
*Penulis Korespondensi (rizalfahrurrozi30@student.uns.ac.id)

Abstrak: Komputer kuantum memiliki keunggulan dalam memproses suatu perhitungan lebih cepat dibanding komputer yang ada saat ini. Keunggulan komputer kuantum menjadi ancaman bagi beberapa sistem kriptografi. Sistem kriptografi McEliece bertahan dari ancaman karena menggunakan kode koreksi kesalahan pada prosesnya. Pada sistem kriptografi McEliece, pesan yang dikirim dengan sengaja ditambah kesalahan sehingga tidak dapat dibaca penyadap. Kode Reed Muller merupakan salah satu kode tertua yang memiliki kemampuan untuk memperbaiki banyak kesalahan. Pada artikel ini akan dijelaskan penerapan kode Reed Muller pada sistem kriptografi McEliece.

Kata kunci: Kode Reed Muller, Komputer Kuantum, Sistem Kriptografi McEliece

Abstract: *Quantum computers have the advantage of processing calculations faster than existing computers. The advantages of quantum computers pose a threat to several cryptosystem. The McEliece cryptosystem survives threats because it uses error-correcting codes in its processes. In the McEliece cryptosystem, messages sent are deliberately added with errors so that eavesdroppers cannot read them. Reed Muller code is one of the oldest codes that has the ability to fix many errors. This article will explain the application of Reed Muller code to the McEliece cryptosystem.*

Keywords*: Reed Muller code, Quantum computers, the McEliece cryptosystem*

PENDAHULUAN

Kriptografi merupakan suatu cabang ilmu yang memperlajari tentang teknik pengamanan pesan. Menurut Wahyuni (2010), terdapat empat tujuan utama dari kriptografi yaitu kerahasian, integritas data, otentikasi, dan nirpenyangkalan. Menurut Sinaga (2017) dan Ilmiayah (2019), algoritma kriptografi dibedakan menjadi algoritma kunci simetris dan algoritma kunci asimetris. Algoritma kunci simetris menggunakan kunci yang sama pada proses enkripsi dan dekripsi. Pada algoritma kunci publik menggunakan dua kunci yaitu kunci publik pada proses enkrisi dan kunci privat pada proses dekripsi.

Proses enkripsi merupakan proses penyandian atau pengamanan pesan asli menjadi pesan yang tesamarkan. Pesan asli biasa disebut dengan *plaintext* dan pesan yang tersamarkan biasa disebut dengan *ciphertext*. Proses dekripsi merupakan proses mengubah *ciphertext* menjadi *plaintext* (Bruce, 1996).

Artikel ini disajikan dalam SENPIKA VI (Seminar Nasional Pendidikan Matematika) yang diselenggarakan oleh Prodi Pendidikan Matematika FKIP Universitas Lambung Mangkurat Banjarmasin pada 22 Juli 2023

Komputer kuantum adalah komputer yang memanfaatkan fenomena pada mekanika kuantum (Bernstein, 2009). Komputer kuantum lebih cepat dibanding komputer konvensional karena melakukan proses perhitungan secara simultan. Pada komputer konvensional perhitungan dilakukan secara linear (Saputra, 2009).

Keunggulan komputer kuantum menjadi ancaman bagi beberapa sistem kriptografi. Komputer kuantum dapat dengan mudah menembus sistem kriptografi yang keamanannya berdasarkan pada sukarnya memfaktorkan bilangan bulat yang besar. Shor (1999) dan Chen (2016) berpedapat bahwa sistem kriptografi RSA yang populer saat ini menjadi tidak aman dengan kedatangan komputer kuantum.

Pada tahun 1978, Robert McEliece merumuskan algoritma kunci asimetris memanfaatkan kode koreksi kesalahan yang kemudian dikenal dengan sistem kriptografi McEliece. Sistem kriptografi McEliece bertahan dari ancaman komputer kuantum karena menggunakan kode koreksi kesalahan pada prosesnya. Pada sistem kriptografi McEliece, pesan yang dikirim dengan sengaja ditambah kesalahan sehingga tidak dapat dibaca penyadap. Penambahan kesalahan tersebut membuat sistem kriptografi tidak terancam dengan adanya komputer kuantum.

Kesalahan dapat terdeteksi dengan cara ditambahkan simbol biner pada pesan yang akan dikirim. Penambahan tersebut berfungsi sebagai bit-bit pendeteksi kesalahan. Bitbit tersebut memberikan gambaran mengenai kondisi pesan yang sesungguhnya sehingga kesalahan (*error*) yang terjadi dapat dideteksi dengan mudah karena terdapat keterkaitan antara pesan dengan bit-bit pedeteksi kesalahan yang dikonstruksi (Irawanto dan Widyaningsih, 2009).

Pada tahun 1954, kode Reed Muller ditemukan oleh David E. Muller dan menjadi salah satu kode tertua. Kode Reed Muller memiliki kemampuan yang cukup baik dalam mengoreksi kesalahan dan sering digunakan pada komunikasi luar angkasa (Meyer, 2021). Sidelnikov (1994) mengusulkan sistem kriptografi McEliece yang berdasarkan pada kode Reed Muller. Oleh karena itu, pada penelitian ini membahas mengenai penerapkan kode Reed Muller pada sistem kriptografi McEliece.

METODE

Metode yang digunakan pada penelitian ini adalah studi literature yang meliputi buku, artikel dalam jurnal, dan tesis terkait dengan sistem kriptografi McEliece dan kode Reed Muller. Selanjutnya dilakukan penerapan kode Reed Muller pada sistem kriptografi McEliece. Sistem kriptografi McEliece dibagi menjadi tiga proses utama yaitu pembentukan kunci, enkripsi pesan dan dekripsi pesan.

Pembentukan kunci sistem kriptografi McEliece

Alogritma pembentukan kunci:

1. Konstruksi matriks generator dari kode Reed Muller. Matriks generator ini berukuran $k \times n$ dan mampu memperbaiki *error* sebanyak t.
2. Konstruksi matriks nonsingular S berukuran $k \times k$ dengan elemen-elemen 0 dan 1. Matriks nonsingular merupakan matriks yang memiliki invers.
3. Konstruks matriks permutasi P berukuran $n \times n$. Matriks permutasi merupakan matriks yang mempunyai tepat satu bit tidak nol pada setiap baris dan kolom.
4. Hitung $G' = SGP$.

Enkripsi pesan sistem kriptografi McEliece

Algoritma enkripsi pada sistem kriptografi McEliece:

1. Pesan asli dikonversi ke bentuk biner menjadi m.
2. Hitung $y = mG' + e$ dimana e merupakan matriks *error* dengan bobot maksimum t.

Dekripsi pesan sistem kriptografi McEliece

Algoritma dekripsi pada sistem kriptografi McEliece:

1. Hitung invers dari P dan S.
2. Hitung $x = yP^{-1}$.
3. Menggunakan algoritma decoding untuk memperoleh m'.
4. Hitung $m'S^{-1}$

HASIL DAN PEMBAHASAN

Ling dan Xing (2004) membahas mengenai pembentukan kode Reed Muller. Pada tahun 2016, Saptharishi mengemukakan tentang *decoding* kode Reed Muller secara efisien dari kesalahan acak. Jaya (2017) meneliti proses *decoding* kode Reed Muller orde pertama menggunakan transformasi hadamard. Selanjutnya pada tahun 2020, Abbe membahas mengenai teori dan algoritma terkait kode Reed Muller.

Siim (2015) berpendapat bahwa hubungan sistem kriptografi McEliece dengan teori koding dapat dijadikan sebagai alternatif dari sistem kriptografi RSA. Ilmiyah (2019) telah melakukan kajian terhadap sistem kriptografi McEliece dalam menghadapi tantangan komputer kuantum di era revolusi industry 4.0. Pada penelitian kali ini membahas mengenai penerapan kode Reed Muller pada sistem kriptografi McEliece.

Proses pembentukan kunci

Pada penelitian kali ini diambil matriks generator dari kode Reed Muller orde pertama RM (1,3) yaitu

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

dengan panjang $n = 8$, dimensi $k = 4$, dan dapat mengoreksi kesalahan sebanyak $t = 1$.

Ambil matriks

$$S = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

dan matriks

$$P = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Menghitung $G' = SGP = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$.

Matriks G, matriks S, dan matriks P merupakan kunci privat sedangkan G' merupakan kunci publik dari kriptografi McEliece.

Proses enkripsi pesan

Pesan "coba" dikonversi ke bentuk *binary string* menjadi "01100011011011110110001001100001". Selanjutnya pesan diubah panjangnya menjadi $k = 4$ sehingga diperoleh

$m_1 = (0110)$,
$m_2 = (0011)$,
$m_3 = (0110)$,
$m_4 = (1111)$,
$m_5 = (0110)$,
$m_6 = (0010)$,
$m_7 = (0110)$, dan
$m_8 = (0001)$.

Ambil vektor biner $e = (00010000)$ dengan bobot maksimal $t = 1$. Melakukan enkripsi dengan menghitung

$$y_1 = m_1 G' + e = (0110)\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} + (00010000)$$

$= (10010111)$,
$y_2 = m_2 G' + e = (01011110)$,
$y_3 = m_3 G' + e = (10010111)$,
$y_4 = m_4 G' + e = (01101000)$,
$y_5 = m_5 G' + e = (10010111)$,
$y_6 = m_6 G' + e = (01110011)$,
$y_7 = m_7 G' + e = (10010111)$,
$y_8 = m_8 G' + e = (00111101)$.

Sehingga diperoleh pesan "coba" yang telah dienkripsi yaitu y =10010111010111101001011101101000100101110111001101110011001111101.

Proses dekripsi pesan

Pesan y diubah panjangnya menjadi

$y_1 = (10010111)$,
$y_2 = (01011110)$,
$y_3 = (10010111)$,
$y_4 = (01101000)$,
$y_5 = (10010111)$,
$y_6 = (01110011)$,
$y_7 = (10010111)$,
$y_8 = (00111101)$.

Menghitung P^{-1} dan S^{-1}

$$P^{-1} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$S^{-1} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Menghitung $x_1 = y_1 P^{-1} = (01111001)$.
Setelah diperoleh x_1, dilakukan decoding sehingga menghasilkan $m_1' = (0111)$.
Selanjutnya menghitung $m_1' S^{-1} = (0110) = m_1$.

Prosedur yang sama dilakukan pada $y_2, y_3, y_4, y_5, y_6, y_7,$ dan y_8 sehingga diperoleh $m_2, m_3, m_4, m_5, m_6, m_7,$ dan m_8 . Sehingga hasil dekripsi pesan adalah $m = 01100011011011110110001001100001$ dan apabila dikonversi menjadi "coba".

Implementasi menggunakan python

SageMath merupakan *software* dan *library* matematika untuk menangani fungsi matematika yang rumit. SageMath menggunakan lisensi GPL dan termasuk *open source* sehingga dapat digunakan oleh semua orang. SageMath memuat kumpulan library matematika dan statistika seperti NumPy, SciPy, matplotlib, Sympy, Maxima, GA, FLINT, R dan masih banyak lainnya. *SageMath* dikembangkan menggunakan bahasa pemrograman Python.

Pembentukan kunci dilakukan dengan menggunakan kunci yang sama pada perhitungan untuk mengetahui program berjalan sesuai dengan yang diharapkan. *Syntax* pembentukan kunci sebagai berikut:

```
C = codes.BinaryReedMullerCode(1, 3)
G = C.generator_matrix()
S = Matrix(GF(2), [
   [0,0,1,1],
   [0,1,1,0],
   [0,0,0,1],
   [1,0,1,1]
])
P = Matrix(GF(2), [
   [0,0,0,0,1,0,0,0],
   [0,0,0,0,0,1,0,0],
   [1,0,0,0,0,0,0,0],
   [0,0,0,1,0,0,0,0],
   [0,0,0,0,0,0,1,0],
   [0,1,0,0,0,0,0,0],
   [0,0,1,0,0,0,0,0],
   [0,0,0,0,0,0,0,1]
])
G2 = S*G*P
```

Program sederhana dibuat untuk enkripsi dan dekripsi pesan. Pengujian dilakukan dengan menggunakan dua pesan berbeda. Pesan pertama merupakan "coba" untuk mengetahui program berjalan dengan baik. Pesan kedua menggunakan kalimat yang lebih panjang untuk mengetahui program mampu memproses pesan yang lebih panjang.

```
pesan_1 = "coba"
pesan_2 = "Kode Reed Muller"
```

Enkripsi dilakukan ke pesan pertama dengan menjalankan **pesan_tersamar_1 = enkrip (pesan_1)** dan dihasilkan pesan tersamarkan pertama berupa '1001011101011110100101110110100010010111011100111001011100111101'.

Kemudian dilakukan enkripsi ke pesan kedua dengan menjalankan **pesan_tersamar_2 = enkrip (pesan_2)** dan dihasilkan pesan tersamarkan kedua berupa '1111010010001100100101110110100010010111111101001001011111011001011100110001000011011001011100111001011111101100110010111110110011001011111111010001110011000100000111101000000101110111010110110011001011100100110100101111001100101111010011001101001011111011001101110100111001'.

Dekripsi dilakukan ke pesan yang telah tersamarkan. Dekripsi pesan tersamarkan yang pertama dengan menjalakan **dekrip (pesan_tersamar_1)** dan dihasilkan 'coba'. Dekripsi pesan tersamarkan yang kedua dilakukan dengan menjalankan **dekrip (pesan_tersamar_2)** dan dihasilkan 'Kode Reed Muller'.

PENUTUP

Berdasarkan hasil dan pembahasan dapat disimpulkan bahwa kode Reed Muller dapat diterapkan pada sistem kriptografi McEliece. Error yang ditambahkan pada saat enkripsi pesan dapat diperbaiki dengan algoritma *decoding* Reed Muller pada proses dekripsi pesan. Penerapan kode Reed Muller pada sistem kriptografi McEliece dapat diimplementasikan pada bahasa pemograman Python. Enkripsi pesan pertama dapat dilakukan dan dihasilkan *output* yang sama dengan perhitungan. Enkripsi pesan kedua juga dapat dilakukan dengan menggunkan pesan yang lebih panjang dibandingkan pesan pertama. Dekripsi pesan tersamarkan pertama dan kedua dihasilkan *output* yang sama dengan pesan aslinya. Sistem kriptografi masih aman untuk diterapkan pada era pasca komputer kuantum. Kelemahan dari sistem kriptografi McEliece yaitu ukuran kunci yang cukup besar.

DAFTAR RUJUKAN

Abbe, E., Shpilka, A., & Ye, M. (2020). Reed–Muller codes: Theory and algorithms. IEEE Transactions on Information Theory, 67(6), 3251-3277.

Bernstein, D. J. (2009). Introduction to post-quantum cryptography. In Post-quantum cryptography (pp. 1-14). Berlin, Heidelberg: Springer Berlin Heidelberg.

Bruce, S. (1996). Applied Cryptography: Protocols, Algorthms, and Source Code in C.-2nd.

Chen, L., Chen, L., Jordan, S., Liu, Y. K., Moody, D., Peralta, R., ... & Smith-Tone, D. (2016). Report on post-quantum cryptography (Vol. 12). Gaithersburg, MD, USA: US Department of Commerce, National Institute of Standards and Technology.

Ilmiyah, N. F. (2019, May). Kajian Tentang Kriptosistem Mceliece Dalam Menghadapi Tantangan Komputer Kuantum Di Era Revolusi Industri 4.0. In Prosiding Seminar Nasional MIPA Kolaborasi (Vol. 1, No. 1, pp. 216-226).

Irawanto, B., & Widyaningsih, S. (2009). Deteksi dan Koreksi Error Pada Pesan Digital Dengan Kode Hamming. Jurnal Sains dan Matematika, 17(3), 127-130.

Jaya, A. K. (2017). Proses Decoding Kode Reed Muller Orde Pertama Menggunakan Transformasi Hadamard. Jurnal Matematika, Statistika dan Komputasi, 13(2), 122-127.

Ling, S., & Xing, C. (2004). Coding theory: a first course. Cambridge University Press.

McEliece, R. J. (1978). A public-key cryptosystem based on algebraic. Coding Thv, 4244, 114-116.

Meyer, L. (2021). Coding and Decoding of Reed-Muller Codes.

Saptharishi, R., Shpilka, A., & Volk, B. L. (2016, June). Efficiently decoding Reed-Muller codes from random errors. In Proceedings of the forty-eighth annual ACM symposium on Theory of Computing (pp. 227-235).

Saputra, H. (2009). Kajian tentang komputer kuantum sebagai pengganti komputer konvensional di masa depan. Generic, 4(2), 15-18.

Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review, 41(2), 303-332.

Sidelnikov, V. M. (1994). A public-key cryptosystem based on binary Reed-Muller codes.

Siim, S., & May, V. S. (2015, May). Study of McEliece cryptosystem. The MTAT. 07.022 Research Seminar in Cryptography, Spring.

Sinaga, M. C. (2017). Kriptografi Python. Matius Celcius Sinaga.

Wahyuni, A. (2010). Aplikasi Kriptosistem dengan Algoritma Mc Elliece. Majalah Ilmiah INFORMATIKA, 1(1).