Implementasi Steganografi Pesan Rahasia Pada Sebuah File Teks

Dimas Wahyu Langkawi - 13518069 (*Author*)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13518069@std.stei.itb.ac.id

***Abstract*—Steganografi lebih dikenal implementasinya dalam menyembunyikan sebuah pesan rahasia dalam file audio, gambar maupun video, namun cara ini akan sulit diimplementasikan khususnya pada *file format* yang merupakan kompresi efisien seperti JPEG dan MPEG. Maka dari itu alternatif penyisipan pesan rahasia dengan metode lain dibutuhkan. Dengan menggunakan dokumen teks biasa, steganografi dapat dicapai dengan menyisipkan pesan rahasia ke dalam dokumen tersebut. Pada makalah ini diimplementasikan *whitespace steganography*.**

Keywords—steganography, steganografi, steganografi teks, whitespace steganography

I. Pendahuluan

Pada era digital ini dimana jutaan bahkan milyaran dokumen dapat diakses dengan mudah di internet dibutuhkan sebuah cara untuk mengamankan dan melindungi isi sebuah pesan penting. Steganografi adalah sebuah area populer untuk melindungi sebuah informasi. Lebih lengkapnya steganografi adalah ilmu dan seni menyembunyikan pesan rahasia dengan suatu cara sedemikian sehingga tidak seorangpun mengetahui keberadaan pesan tersebut. Berbeda dengan kriptografi yang memiliki tujuan menyembunyikan isi atau konten dari sebuah pesan, steganografi bertujuan untuk menyembunyikan keberadaan pesan itu sendiri sehingga pihak ketiga tidak akan curiga.

Steganografi sudah ada bahkan sebelum era digital, contoh steganografi sebelum era digital yaitu menyembunyikan pesan dengan menggunakan tinta khusus supaya tidak terlihat secara sekilas mata. Terinspirasi dari sini pada makalah ini diimplementasikan sebuah steganografi yang menyembunyikan sebuah pesan rahasia dalam sebuah file teks biasa sedemikian rupa sehingga hasil file steganografi teks tidak akan terlihat sekilas mata.

II. Landasan Teori

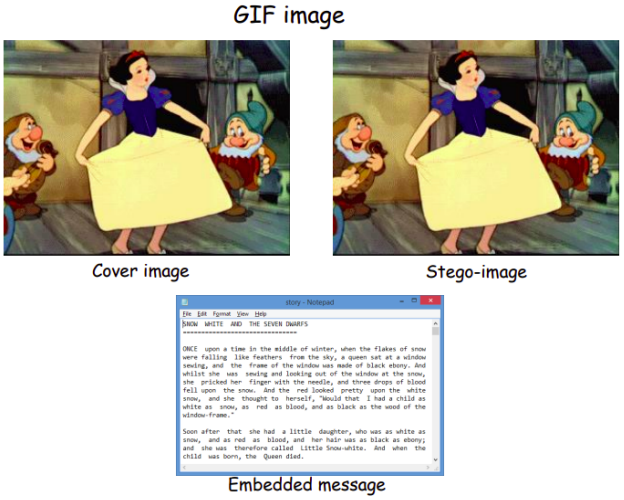
A. Steganografi

Steganografi berasal dari kata steganos yang artinya tersembunyi dan graphien yang artinya tulisan jadi steganografi adalah tulisan tersembunyi membuat kata steganografi berarti tulisan tersembunyi. Lebih lengkapnya lagi steganografi adalah ilmu dan seni menyembunyikan pesan rahasia dengan suatu cara sedemikian sehingga tidak seorangpun mengetahui keberadaan pesan tersebut[1].

Steganografi modern atau steganografi digital adalah sebuah cara untuk menyembunyikan pesan digital dalam dokumen digital lainnya. Dokumen yang digunakan sebagai media untuk menyembunyikan pesan rahasia disebut dengan *carrier file*. Umumnya steganografi modern menggunakan dokumen berformat audio, gambar atau bahkan video. Pada makalah ini steganografi akan lebih difokuskan dalam penyembunyian pesan rahasia pada sebuah dokumen teks biasa.

Beberapa terminologi steganografi contohnya:

1. *Embedded message* atau *secret message* adalah pesan yang disembunyikan baik dalam bentuk teks, gambar, audio, video, dll.
2. *Cover object* atau *carrier file* adalah media digital yang digunakan untuk menyembunyikan *embedded message*
3. *Stego-object* (*stego-data*) adalah media yang sudah berisi pesan *embedded message*.
4. *Stego-key* adalah kunci yang digunakan untuk menyisipkan pesan dan mengekstraksi pesan dari stego-object.



Gambar 1. Contoh steganografi teks pada sebuah gambar[1].

Steganografi yang bagus memiliki kriteria kriteria berikut:

1. *Imperceptible*

   Keberadaan pesan rahasia tidak dapat dipersepsi secara visual atau secara audial

2. *Fidelity*

   Kualitas *cover-object* tidak jauh berubah akibat penyisipan pesan rahasia.

3. *Recovery*

   Pesan yang disembunyikan harus dapat diekstraksi kembali.

4. *Capacity*

   Ukuran pesan yang disembunyikan sedapat sebesar mungkin

Walaupun cara steganografi berbeda dengan kriptografi dalam penyembunyian pesan, steganografi bukanlah pengganti kriptografi namun keduanya dapat digabungkan. Keamanan dari pesan rahasia akan semakin tinggi jika steganografi dan kriptografi digabungkan. Pesan rahasia dapat dienkripsi terlebih dahulu dengan sebuah algoritma kriptografi dan kemudian disembunyikan dengan sebuah algoritma steganografi ke dalam sebuah media lain.

Secara umum ada 3 jenis steganografi:

1. *Pure steganography*

   Tidak membutuhkan kunci sama sekali. Keamanan steganografi seluruhnya bergantung pada algoritmanya.

2. *Secret* (*or symmetric*) *key Steganography*

   Menggunakan kunci yang sama untuk *embedding* dan *extraction*.

3. Public-key Steganography

   Menggunakan dua kunci: kunci publik untuk embedding dan kunci privat untuk extraction.

B. Whitespace steganography

Algoritma steganografi yang diimplementasikan pada makalah ini adalah algoritma *whitespace steganography* yang tergolong pada *pure steganography* karena tidak memerlukan kunci dan pesan rahasia langsung dipetakan dalam bentuk spasi dan tab.

Cara ini bergantung pada fakta bahwa spasi dan tab pada teks tidak akan terlihat secara sekilas mata pada file editor umum seperti notepad. Sehingga menambahkan spasi sebagai tanda bit 0 dan tab sebagai tanda bit 1 dapat menghasilkan kode dari pesan rahasia yang tidak terlihat secara kasat mata.

III. IMPLEMENTASI

Percobaan yang akan dilakukan untuk makalah ini yaitu implementasi dari algoritma *whitespace steganography*. Untuk mengimplementasikan algoritma ini dibutuhkan pemisah antara file teks asli dengan pesan rahasia yang telah diubah, hal ini dilakukan supaya dapat dilakukan ekstraksi dari pesan rahasia tersebut karena pesan rahasia dan teks sudah ditentukan pemisahnya. Pemisah yang diimplementasikan yaitu sebuah karakter garis baru dan tiga buah karakter tab. Semua karakter setelah empat karakter ini adalah pesan rahasia yang telah diubah oleh algoritma.

Proses perubahan pesan rahasia atau proses enkripsi yang diimplementasikan untuk makalah ini yaitu dengan mengubah pesan ke dalam bentuk byte dan diproses bit per bitnya. Bit 0 akan diubah menjadi karakter spasi dan bit 1 akan diubah menjadi karakter tab, hal ini dilakukan terus menerus dengan menyambungkan karakter yang baru diubah ke dalam string dan kemudian ditambahkan ke akhir file teks atau *carrier file* setelah menambahkan karakter karakter yang digunakan untuk menandai pemisah.

Proses berikutnya yaitu dekripsi atau mengekstrak pesan rahasia dari sebuah *carrier file* yang telah disisipkan pesan rahasia atau *stego-object*. Pesan rahasia yang telah dienkripsi dapat didapatkan dengan mencari karakter-karakter pemisah yang digunakan pada tahap enkripsi lalu mengubah kembali dari bentuk bit menjadi pesan asli dalam bentuk string.

Implementasi yang dilakukan dalam bahasa python 3.8.11 dengan kode sebagai berikut:

```
import re

class WhitespaceSteganography:
    def __init__(self) -> None:
        self.delimiter = '\n\t\t\t'
        self.character_set = [" ", "\t"]

    def encrypt(self, text, msg):
```

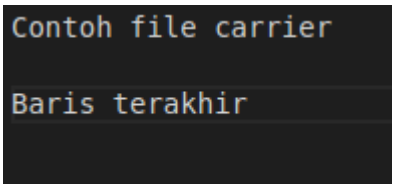
```
        code = ''
        msg_bytes = bytes(msg, 'utf-8')
        bit_mask = 0b00000001
        for by in msg_bytes:
            for _ in range(8):
                m_byte = by & bit_mask
                by = by >> 1
                if m_byte == 0:
                    code += self.character_set[0]
                elif m_byte == 1:
                    code += self.character_set[1]

        return text + self.delimiter + code

    def decrypt(self, text):
        regex = "{}(.+)$".format(self.delimiter)
        m = re.search(regex, text)
        code = m.groups()[0]
        msg_bytes = []

        for i in range(8, len(code)+1, 8):
            m_byte = 0b00000000
            encoded_byte = code[i-8:i]

            for j in range(8):
                m_byte >>= 1
                if encoded_byte[j] == self.character_set[1]:
                    m_byte |= 0b10000000
            msg_bytes.append(m_byte)

        return bytes(msg_bytes).decode('utf-8')
```

Implementasi dari blok kode diatas sesuai yang telah dijelaskan pada paragraf sebelumnya. Program yang menggunakan kelas ini yaitu sebagai berikut:

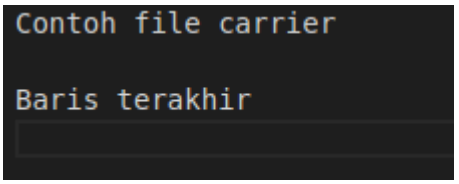
```
import algorithm

if __name__ == "__main__":
    whitepace_steganography =
algorithm.WhitespaceSteganography()
    with open("example_input") as file:
        encoded =
whitepace_steganography.encrypt(file.read(), "secret")

    with open("example_encoded", 'w') as file:
        file.write(encoded)
    decoded = whitepace_steganography.decrypt(encoded)
    print(decoded)
```

file input perlu disiapkan terlebih dahulu dalam file bernama “example_input” dan file ini yang akan dibaca oleh program dan akan disisipkan sebuah pesan rahasia dan dalam kasus ini pesan rahasianya berupa string berisikan “secret”.

IV. Hasil dan Analisis



Gambar 2 *Carrier file*

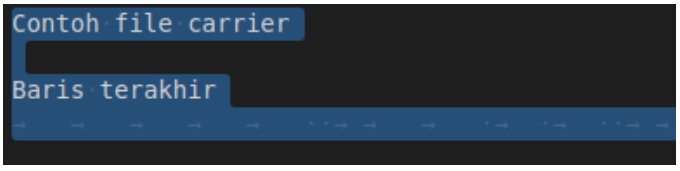


Gambar 3 *stego-object*

Kedua gambar diatas yaitu gambar 2 dan gambar 3 adalah contoh sebelum dan sesudah pesan rahasia disisipkan ke sebuah file teks biasa. Dapat dilihat bahwa secara kasat mata pesan rahasia tersembunyi dengan cukup baik. Pesan rahasia yang disisipkan adalah sebuah string “*secret*”.

Algoritma yang digunakan sudah cukup menjalankan tugasnya dalam menyembunyikan sebuah pesan rahasia. Namun masih banyak yang dapat dikembangkan dari algoritma ini. Setelah diamati secara lebih seksama ukuran *file* bertambah cukup banyak setelah disisipi pesan rahasia tersebut, salah satu faktornya adalah memang sedikitnya konten dari *file* teks yang digunakan sebagai contoh, jika *file* yang digunakan adalah *file* dokumen biasa berukuran beberapa megabyte kemungkinan pihak ketiga tidak akan menyadari perubahan ukuran file.

Kelebihan algoritma ini dibanding steganografi lainnya yaitu ukuran pesan yang dapat disembunyikan relatif tidak ada batasnya karena berbeda dengan steganografi gambar yang menyisipkan bit pada LSB (*least significant bit*) algoritma ini dapat menyisipkan pesan rahasia yang telah diproses dimanapun namun semakin banyak pesan rahasia yang ingin disembunyikan maka semakin besar pula *stego-object* yang akan dihasilkan. Cara lain untuk menangani masalah ukuran pesan rahasia yang ingin disisipkan ini yaitu dengan melakukan kompresi dari pesan rahasia terlebih dahulu.



Gambar 4. *stego-object* yang di *highlight*

Meskipun secara sekilas mata file teks tidak berubah namun algoritma ini memiliki kelemahan seperti jika teks

editor yang digunakan menunjukkan karakter karakter yang digunakan pada algoritma maka pihak ketiga dapat dengan mudah mencurigai *stego-object* ini, atau jika pihak ketiga kebetulan melakukan *highlight* seperti pada gambar 4 maka pihak ketiga juga dapat menyadari kejanggalan file teks tersebut dan timbullah sebuah kecurigaan karena karakter spasi dan tab dapat terlihat secara langsung.

V. Kesimpulan dan Saran

Algoritma yang digunakan hanya algoritma *pure steganography* biasa tanpa ditambahkan apapun, algoritma ini jelas dapat ditingkatkan lagi dengan melakukan enkripsi terlebih dahulu terhadap pesan rahasia sehingga meskipun pihak ketiga melihat kejanggalan dari file teks tersebut dan berhasil mendekripsikannya kembali ke dalam string, pihak ketiga tersebut tetap tidak akan mengetahui konten dari pesan rahasia.

Pada makalah ini hanya diimplementasikan sebuah algoritma penyisipan pesan rahasia dalam *file* teks yaitu dengan *whitespace steganography*. Selain algoritma ini ada beberapa algoritma lain yang dapat menjadi alternatif. Contohnya adalah algoritma steganografi *Zero Width Character* yang memanfaatkan standar unicode untuk menggunakan karakter non-cetak atau *zero width* seperti "\u200b" spasi *zero width*, yang tidak akan terlihat secara kasat mata pada kebanyakan *browser* dan *text editor* standar seperti notepad. Dengan menggunakan karakter ini pesan rahasia dapat disisipkan kedalam sebuah *file* teks tanpa menimbulkan kecurigaan.

Ucapan Terima Kasih

Ucapan terima kasih sebesar besarnya penulis berikan kepada Allah SWT yang telah memberikan nikmat sehat baik jasmani maupun rohani serta rezeki untuk menyelesaikan makalah ini. Dan tentu tak lupa pula ucapan terima kasih penulis ucapkan untuk semua pihak yang telah membantu dan mendukung dibuatnya makalah ini. Mudah-mudahan makalah ini dapat menambah ilmu bagi pembaca.

Referensi

[1] R. Munir, "Steganografi Bagian 1," 2021, [Online], Diakses pada 18 Desember 2021, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Steganografi-Bagian1-2020.pdf

[2] R. Munir, "Steganografi Bagian 2," 2021, [Online], Diakses pada 18 Desember 2021, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Steganografi-Bagian2-2020.pdf

[3] M. Shaifizat, D. Roshidi, dan S. Azman "Analysis of Natural Language Steganography", Diakses pada 19 Desember 2021, dari https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.227.7622&rep=rep1&type=pdf

[4] L. Y. Por, T. F. Ang, dan B. Delina "WhiteSteg: A New Scheme in Information Hiding Using Text Steganography", Diakses pada 20 Desember 2021, dari https://www.researchgate.net/profile/Yee-Por/publication/228672143_WhiteSteg_A_new_scheme_in_information_hiding_using_text_steganography/links/0deec531ae4e4dd4a9000000/WhiteSteg-A-new-scheme-in-information-hiding-using-text-steganography.pdf

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Desember 2021

Dimas Wahyu Langkawi

13518069