Pengiriman *File* dengan Menggunakan *Shamir's Secret Sharing Scheme*

Muhammad Cisco Zulfikar - 13518073
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
ciscozulfikar25@gmail.com, 13518073@std.stei.itb.ac.id

***Abstrak*—Pengiriman *file* melalui internet merupakan suatu hal yang mudah. Namun, seorang penyadap dapat dengan mudah mengambil *file* tersebut dan menyebarkan isi dari *file* tersebut. Terjadinya kejadian tersebut berbahaya. Untuk mengurangi peluang kemudahan dari pengambilan konten dari *file* tersebut, digunakanlah teori kriptografi. *Shamir's secret sharing scheme* merupakan salah satu skema algoritma yang mengenkripsikan suatu pesan kepada beberapa pihak sehingga konten dari psean tersebut sulit untuk dipecahkan. Makalah ini mencoba untuk membuat sistem pengiriman *file* dengan menggunakan skema tersebut.**

***Kata kunci—file, divide,* skema pembagian rahaisa Shamir**

I. PENDAHULUAN

Kriptografi adalah sebuah ilmu dan seni untuk menjaga keamanan sebuah pesan. Teknik yang umum digunakan pada kriptografi adalah dengan menggunakan enkripsi dan dekripsi pada pesan yang akan dikirimkan. Enkripsi melakukan penyembunyian informasi menjadi sebuah kode rahasia saat pesan akan dikirim. Dekripsi melakukan pengembalian bentuk pesan dari kode rahasia menjadi pesan dengan bentuk semula. Diperlukan kunci rahasia untuk melakukan dua proses di atas. Kunci tersebut harus juga disepakati oleh dua pihak, pihak pengirim dan pihak penerima.

Dengan berjalannya waktu, kriptografi mengalami perkembangan. Terdapat beberapa kemunculan teknik-teknik dalam kriptografi. Berkembang pula ilmu kriptanalisis, yaitu ilmu dalam memecahkan kode rahasia menjadi pesan yang dapat dibaca tanpa mengetahui kunci rahasia yang digunakan. Selain itu, diperkenalkan pula kriptografi dengan enkripsi asimetrik yang menggunakan kunci publik dan kunci privat.

Dengan menggunakan kriptografi publik, pengirim akan membuat dua buah kunci, kunci publik dan kunci privat. Kunci publik digunakan untuk mendekripsi pesan dan kunci privat digunakan untuk me-enkripsi pesan. Kunci publik dapat dibagikan kepada siapa saja, tetapi kunci privat harus tetap rahasia dan hanya pemilik kunci privat yang tahu.

Di beberapa situasi, kunci privat terkadang diinginkan untuk dijaga kerahasiaanya tidak hanya oleh satu orang, melainkan dijaga oleh lebih dari satu orang. Sebagai contoh, untuk melakukan sebuah peluncuran misil nuklir, diperlukan konfirmasi dan kunci peluncuran dari dua orang. Contoh lain yang dapat digunakan adalah penggunaan dua kunci untuk membuka sebuah brankas bank. Kunci tersebut dipegang oleh karyawan bank dan pemilik brangkas. Dari situasi-situasi yang telah disebutkan sebelumnya, telah diterapkan sebuah mekanisme yang dinamakan *Two-Man Rule*, yang pada dasarnya semua akses dan tindakan memerlukan kehadiran dua atau lebih orang yang berwenang pada setiap saat. Hal tersebut membuat pendistribusian kunci privat kepada beberapa orang membuat kerahasiaan informasi menjadi lebih aman karena menghindari peluang terjadinya *loss of key*. Selain untuk menyimpan informasi, mekanisme di atas dapat pula digunakan untuk mengirimkan sebuah pesan.

Berdasarkan informasi tersebut, *secret sharing scheme* merupakan salah satu cara untuk menyebarkan sebuah pesan rahasia kepada beberapa pihak tertentu dengan setiap pihak memegang sebuah bagian dari rahasia tersebut. Salah satu skema yang dikenal adalah *Shamir's secret sharing scheme*. Skema pembagian rahasia Shamir membagi sebuah pesan rahasia ke dalam beberapa bagian yang kemudian dibagikan kepada sejumlah pihak.

Dalam makalah ini, akan dibahas bagaimana menyebarkan sebuah *file* dokumen dengan menggunakan *Shamir's secret sharing scheme*. Dengan begitu, pihak pengirim dan penerima dapat saling mengetahui konten dari *file* yang dikirimkan tersebut secara rahasia.

II. DASAR TEORI

A. Kriptografi

Kriptografi adalah sebuah ilmu dan seni untuk menjaga keamanan sebuah pesan. Secara bahasa, kriptografi berasal dari Bahasa Yunani berupa kata *cryptos* yang berarti rahasia dan *graphein* yang berarti tulisan. Dengan menggabungkan dua kata tersebut, kriptografi dapat diartikan secara literal sebagai tulisan rahasia. Secara lebih umum, kriptografi adalah tentang membangun dan menganalisis protokol yang mencegah pihak ketiga atau publik membaca pesan privat [1].

Pada kriptografi modern, dibahas peran dan penggunaan pada informasi. Pengetahuan pada kriptografi modern berpotongan dengan disiplin ilmu matematika, ilmu komputer, teknik elektro, ilmu komunikasi, dan fisika.

Terdapat empat aspek yang berkaitan dengan kriptografi dan keamanan informasi, yaitu *confidentiality* (kerahasiaan), *data integrity* (kebenaran dari data), *authentication* (otentikasi), dan *non-repudiation* (tidak dapat disangkal) [2].

Pada konteks kriptografi terdapat beberapa istilah yang perlu dimengerti dan lazim dipakai untuk menggambarkan sebuah objek informasi.

1. Pesan

Pesan adalah sebuah bentuk dasar dari sebuah informasi. Sebuah pesan dapat dipahami dengan mudah. Sebuah pesan dapat berbentuk apapun, baik itu dalam bentuk teks, gambar, audio, maupun video.

2. Plainteks

Plainteks adalah bentuk awal dari sebuah pesan sebelum dilakukan proses enkripsi. Plainteks juga dapat dihasilkan melalui proses dekripsi.

3. Cipherteks

Cipherteks adalah bentuk dari sebuah pesan yang tidak dapat dipahami dengan mudah. Cipherteks merupakan hasil enkripsi dari sebuah plainteks. Cipherteks biasanya dikirimkan melalui pihak ketiga sehingga cipherteks dibuat sedemikian rupa agar isinya tidak dengan mudah diketahui oleh pihak ketiga tersebut.

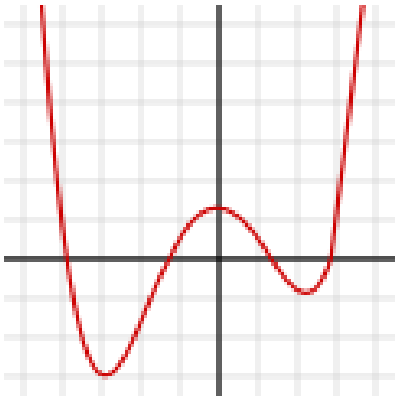
B. *Polinomial*

Dalam matematika, sebuah polinomal atau suku banyak didefinisikan sebagai sebagai ekspresi yang terdiri dari variabel dan koefisien, yang melibatkan operasi penjumlahan, pengurangan, perkalian, perpangkatan bilangan bulat nonnegatif.

Fungsi polinomial adalah sebuah fungsi yang dapat definisikan dengan mengevaluasi sebuah polinomial. Sebuah fungsi polinomial biasanya dinotasikan sebagai berikut.

$$a_n x^n + a_{n-1} x^{n-1} + a_2 x^2 + a_1 x + a_0 \qquad (1)$$

Dalam notasi di atas, n merupakan bilangan nonnegatif, a merupakan koefisien konstanta, dan x merupakan salah satu variabel. Derajat polinomial mengacu kepada nilai pangkat tertinggi yang mempunyai koefisien tidak nol. Oleh karena itu, notasi di atas mempunyai nilai derajat polinomial berupa n.

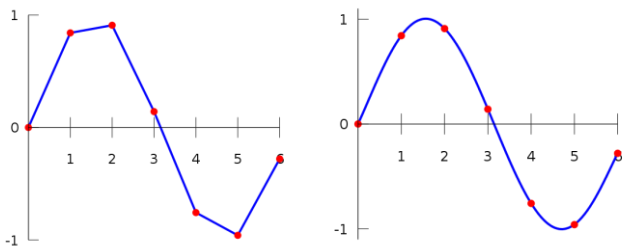


Gambar 1. Sebuah grafik dari fungsi polinomial berderajat 4

C. *Interpolasi*

Dalam matematika, interpolasi didefinisikan sebagai sebuah cara untuk mengestimasi, sebuah metode untuk mengkonstruksi titik data yang baru yang masuk ke dalam jarak set diskrit berisi data yang diketahui [3].

Terdapat beberapa cara untuk menginterpolasi sebuah data. Salah satu metode yang sederhana adalah dengan menggunakan interpolasi linear yang menghubungkan semua titik dengan sebuah garis. Metode lainnya adalah interpolasi polinomial dengan mencari satu fungsi ppolinomial dengan derajat $n-1$ jika terdapat sejumlah n titik yang diketahui.



Gambar 2. Sebuah grafik dari interpolasi linear (kiri) dan interpolasi polinomial (kanan) dengan menggunakan data berupa tujuh titik

D. *Polinomial Lagrange*

Polinomial Lagrange secara umum dapat membantu permasalahan interpolasi dalam matematika. Pertama kali ditemukan oleh Edward Waring pada tahun 1779, metode ini lebih dikenal saat Joseph-Louis Lagrange memublikasikannya pada tahun 1795.

Polinomial Lagrange didefinisikan sebagai berikut. Diberikan sebuah set yang terdiri dari $k+1$ titik data

$$(x_0, y_0), \ldots, (x_j, y_j), \ldots, (x_k, y_k)$$

dengan tidak ada dua x_j yang sama, interpolasi polinomial dalam bentuk Lagrange adalah *linear combination*

$$L(x) := \sum_{j=0}^{k} y_j \ell_j(x) \qquad (2)$$

dari basis polinomial Lagrange

$$\ell_j(x) := \prod_{\substack{0 \le m \le k \\ m \ne j}} \frac{x - x_m}{x_j - x_m} = \frac{(x - x_0)}{(x_j - x_0)} \cdots \frac{(x - x_{j-1})}{(x_j - x_{j-1})} \frac{(x - x_{j+1})}{(x_j - x_{j+1})} \cdots \frac{(x - x_k)}{(x_j - x_k)} \quad (3)$$

dengan $0 \le j \le k$.

E. Shamir's Secret Sharing Scheme

Skema pembagian rahasia Shamir adalah sebuah algoritma pembagian rahasia yang ditemukan oleh Adi Shamir, orang yang juga menemukan algoritma RSA. Pada dasarnya, algoritma ini mempunyai ide untuk memecahkan suatu data menjadi *n* banyak *file* dengan nilai $n > 1$. Lalu, perlu disepakati *threshold* atau batas berupa *k* dengan nilai $0 < k \le n$ sebagai banyaknya *file* ang diperlukan untuk mengkonstruksi ulang rahasia yang sebelumnya dibagi.

Dengan menggunakan skema *threshold* (*k, n*) untuk membagi rahasia *S* dengan nilai $S < P$ dan *P* merupakan bilangan prima, pilih $k-1$ bilangan positif dengan $a_i < P$ dan $a_0 = S$. Polinomial yang dibangun menjadi polinomial $f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + ... + a_{k-1}x^{k-1}$. Ambil titik *n* apapun dari polinomial tersebut, sebagai contoh pada set $i = 1, \ldots, n$ untuk mengambil $(i, f(i))$, setiap pihak diberikan sebuah titik berupa bilangan bulat tdak nol yang kemudian dimodulokan dengan *P*. Untuk merekontruksi ulang rahasia dapat digunakan, dapat digunakan interpolasi polinomial Lagrange.

Keuntungan dengan menggunakan skema pembagian rahasia Shamir adalah sebagai berikut.

1. Aman

Pesan *Information theoretic security* menurut teori kriptosistem, dengan maksud aman walaupun dengan menggunakan daya komputasi tinggi dan waktu tidak.

2. *Minimal*

Setiap hasil *share* yang dihasilkan memiliki *size* yang tidak melebihi *file* aslinya.

3. Extensible

Ketika jumlah *k* telah ditetapkan, jumlah *share* dapat ditambah maupun dikurangi tanpa mempengaruhi *shares* lainnya.

4. Dinamis

Keamanan dapat ditingkatkan tanpa mengganti rahasianya dengan cara mengganti nilai polinomial secara berkala dan membuat sejumlah *share* baru kepada semua pihak.

5. Fleksibel

Dengan menggunakan contoh sebuah organisasi yang memiliki tingkat hierarki, ketua dapat membaca rahasia tersebut sendiri, sedangkan dibutuhkan 3 jumlah *staf* untuk membaca rahasia tersebut.

III. IMPLEMENTASI

Dalam makalah ini, terdapat dua buah pendekatan yang digunakan penulis untuk membagi sebuah *file* berbentuk teks menjadi sejumlah *share*, yaitu pembagian sederhana (*simple divide)* dan pembagian dalam bentuk blok (*block divide*).

A. Simple Divide

Pada pembagian *simple divide*, dokumen *file* teks dibagi kedalam *n* buah *shares* dengan *threshold k* ditentukan secara bersamaan. Kemudian, *file* tersebut disimpan ke dalam *cloud* dengan folder yang berbeda untuk setiap *share*-nya.

Sebagi contoh, sebuah *file* dengan format *PDF* hanya memiliki konten berupa teks. Kemudian, konten tersebut diekstrak dan diubah sedemikian rupa agar dapat dibagikan menjadi sejumlah *n* buah *share* dan disaat yang bersamaan ditentukan nilai *k* untuk mengetahui berapa yang dibutuhkan untuk dapat direkonstruksi ulang. Pada akhirnya, masing-masing *share* diunggah ke dalam sebuah folder yang berbeda dengan menggunakan *cloud*.

B. Block Divide

Mirip dengan pembagian *simple divide*, pada pembagian *block divide*, dokumen *file* teks dibagi kedalam *n* buah *shares* dengan *threshold k* ditentukan secara bersamaan. Namun, konten dari *file* tersebut dibagi terlebih dahulu menjadi beberapa blok dengan isi masing-masing blok berjumlah *b* karakter. Kemudian, *file* tersebut disimpan ke dalam *cloud* dengan folder yang berbeda untuk setiap *share*-nya.

Misalnya, sebuah *file* dengan format *PDF* hanya memiliki konten berupa teks. Kemudian, konten tersebut diekstrak, lalu dibagi menjadi beberapa blok sejumlah *b* buah.. Masing-masing blok diubah sedemikian rupa agar dapat dibagikan menjadi sejumlah *n* buah *share* dan disaat yang bersamaan ditentukan nilai *k* untuk mengetahui berapa yang dibutuhkan untuk dapat direkonstruksi ulang. Pada akhirnya, masing-masing *share* diunggah ke dalam sebuah folder yang berbeda dengan menggunakan *cloud*.

IV. HASIL PENGUJIAN DAN ANALISIS

Pada pengujian diberikan sebuah *file* yang hanya berisikan teks dengan format *file* berupa *.pdf*. Di dalam *file* tersebut, berisikan sebuah teks yang bertuliskan "`Hello!`".

Untuk mengekstrak teks tersebut, dibuatlah sebuah kode untuk mengambil teks tersebut hingga menjadi sebuah masukan berbentuk *string*. Berikut potongan kode tersebut beserta hasilnya.

```
# Menggunakan library PyPDF2
import PyPDF2

# Lokasi file PDF (dapat diganti)
pdf_location = r'D:\ITB\Semester
7\Kriptografi\Hello.pdf'
fhandle = open(pdf_location, 'rb')

# Membaca isi file
pdfReader = PyPDF2.PdfFileReader(fhandle)
pagehandle = pdfReader.getPage(0)

# Ekstrak isi file
extracted_string = pagehandle.extractText()
extracted_string = extracted_string.strip()
print(extracted_string)

>> Hello!
```

A. Simple Divide

String di atas kemudian diubah kedalam format ASCII. Hal tersebut akan membuat tahap selanjutnya menjadi lebih mudah. Berikut potongan kode tersebut beserta hasilnya.

```
# Mengubah karakter ke dalam bentuk ASCII
ascii_values = []
for character in extracted_string:
    ascii_values.append(ord(character))
print(ascii_values)

>> [72, 101, 108, 108, 111, 33]
```

Setelah diubah ke dalam bentuk ASCII, dibuatlah karakter ASCII tersebut menjadi sebuah *string* yang lalu diubah menjadi sebuah *integer*. Berikut potongan kode beserta hasilnya.

```
# Mengubah string ASCII menjadi integer
list_to_str = ' '.join([str(elem) for elem
in ascii_values])
list_to_str = list_to_str.replace(' ','')
converted_text = int(list_to_str)
print(converted_text)

>> 7210110810811133
```

Setelah sebuah *integer* yang berasal dari teks telah dihasilkan, dalam kasus ini S = **7210110810811133**, saatnya dimulai untuk membagikan pesan tersebut dengan menggunakan skema pembagian rahasia Shamir.

Terdapat beberapa hal yang perlu dilakukan untuk membagi pesan rahasia S tersebut dengan menggunakan skema rahasia pembagian Shamir. Perlu dicari terlebih dahulu sebuah P sembarang dengan nilai $S < P$. Berikut potongan kode beserta hasilnya.

```
# Menggunakan library 'random'
import random

# Nilai terbesar adalah (2**127) - 1
MAX_INT = (2**127) - 1

# Fungsi pencarian bilangan prima dari
range x hingga y
def primes_in_range(x, y):
    prime_list = []
    for n in range(x, y):
        is_prime = True
        for num in range(2, n):
            if n % num == 0:
                is_prime = False
        if is_prime:
            prime_list.append(n)
    return prime_list

# Nilai prima
prime_list =
primes_in_range(converted_text, MAX_INT)
random_prime = random.choice(prime_list)
print(random_prime)

>> 10109134959066001
```

Setelah mendapatkan nilai P = **10109134959066001**, ditentukanlah berapa jumlah *share* yang akan dihasilkan serta berapa *threshold* yang akan menjadi patokannya. Untuk kasus ini, jumlah share yang akan dibuat sebanyak $n = 5$ dengan nilai *threshold* $k = 3$. Artinya, dibutuhkan sebanyak $k-1$ atau dua buah bilangan acak untuk membentuk sebuah polinomial. Berikut potongan kode beserta hasilnya.

```
# Threshold = 3
k = 3

def generate_coeff():
      coeff_list = []
      for i in range(0, k-1):
          n = random.randint(0, MAX_INT)
          coeff_list.append(n)
      return coeff_list

print(coeff_list)

>> [540732101132748, 891092712178493]
```

Setelah itu, dibuatlah sebuah polinomial *f(x)* berdasarkan nilai S, P, dan dua koefisien yang telah diketahui. Untuk kasus ini,

$$f(x) \equiv 7210110810811133 + 540732101132748x + 891092712178493x^2 \pmod{10109134959066001}$$

Dengan begitu, tahap selanjuatnya adalah membuat *share* sebanyak *n* buah. Untuk mendapatkan nilai untuk masing-masing *share*, nilai *x* diiterasi sebanyak *n* kali dengan setiap iterasi nilai $x = x+1$. Nilai *share* yang dihasilkan adalah sebagai berikut.

```
# Hasil share
>> [1, 2677990910660367]
>> [2, 5892001148328599]
>> [3, 779061851287806]
>> [4, 7557442937670005]
>> [5, 6008874489343189]
```

Tahapan selanjutnya adalah memasukkan tiap share ke dalam sebuah folder di dalam *cloud*. Setiap *share* dapat dimasukkan ke dalam sebuah *file* berformat *PDF*.

Gambar 3. *File share* disimpan ke dalam sebuah folder di dalam *cloud*.

B. *Block Divide*

Dengan menggunakan *string* yang sama, yaitu "`Hello!`", kita ubah *string* tersebut ke dalam bentuk ASCII. Namun, hal yang membuat berbeda dengan *simple divide* adalah *string* tersebut akan dibagi kedalam beberapa blok dengan sejumlah *b* buah. Hasil yang didapatkan menjadi seperti berikut.

```
# Menggunakan library NumPy
import numpy as np

# Mengubah karakter ke dalam bentuk ASCII
dan membaginya kedalam b jumlah blok
ascii_values = []
for character in extracted_string:
    ascii_values.append(ord(character))

splits = np.array_split(ascii_values, 3)
for array in splits:
    print(list(array))

>> [72, 101]
>> [108, 108]
>> [111, 33]
```

Setelah membaginya ke dalam *b* blok, dalam kasus ini menjadi 3 buah blok, dilakukan tahapan-tahapan seperti pada kasus pembagian *simple divide*.

Sebagai contoh, kita ambil `[72, 101]` dan mengubahnya mejadi sebuah integer dengan nilai $S = 72101$. Lalu, digenerasikanlah sebuah *P* dengan nilai $P = 128767$ dengan kode yang telah dibuat sebelumnya. Setelah itu, sama dengan kasus sebelumnya, pada contoh ini diinginkan jumlah share yang akan dibuat sebanyak $n = 5$ dengan nilai *threshold* $k = 3$. Artinya, juga dibutuhkan sebanyak $k-1$ atau dua buah bilangan acak untuk membentuk sebuah polinomial. Misalkan, dengan meggunakan kode yang telah dibuat, hasil generasi nilai acak adalah `35648` dan `8907`. Maka, dapat dibentuklah sebuah *f(x)* berdasarkan nilai *S*, *P*, dan dua koefisien yang telah diketahui. Untuk kasus ini,

$$f(x) \equiv 72101 + 35648x + 8907x^2 \pmod{128767}$$

Selanjutnya, dibuatlah sejumlah *share* sebanyak *n* buah. Sama seperti sebelumnya, nilai untuk masing-masing *share*, nilai *x* diiterasi sebanyak *n* kali dengan setiap iterasi nilai $x = x+1$. Nilai *share* yang dihasilkan adalah sebagai berikut.

```
# Hasil share
>> [1, 116656]
>> [2, 50258]
>> [3, 1674]
>> [4, 99671]
>> [5, 86715]
```

Tahapan selanjutnya adalah memasukkan tiap share ke dalam sebuah folder di dalam *cloud*. Setiap *share* dapat dimasukkan ke dalam sebuah *file* berformat *PDF*.

Keuntungan yang dapat diamati dengan menggunakan *block divide* adalah sebagai berikut.

1. Lebih banyak *share*

Dalam membuat *share*, jika pesan rahasia dibagi terlebih dahulu kedalam beberapa blok, jumlah *share* yang akan dibuat pun meningkat. Dalam kasus ini, dengan membagi pesan kedalam 3 blok yang lalu dibuat 5 buah *share* untuk masing-masing blok, maka jumlah *share* keseluruhan adalah 15 buah *share*, sehingga jika seseorang akan mengambil semua *share* dari satu bagian, informasi yang didapat tidak mencukup karena diperlukan untuk mengumpulkan seluruh *share* di setiap bloknya.

2. Komputasi yang lebih sedikit dalam pembuatan *share*

Setiap pembuatan *share*, waktu yang digunakan lebih cepat. Hal ini disebabkan *string* panjang yang seharusnya dikerjakan hanya sekali, pada kasus ini dibagi kedalam tiga tahap.

V. KESIMPULAN DAN SARAN

Proof of concept dari mengirimkan sebuah file dengan menggunakan *Shamir's secret sharing scheme* dapat dilakukan dengan baik. Namun, muncul beberapa masalah dalam penanganannya. Apabila teks dalam sebuah *file* banyak, pembuatan *share* akan menjadi sulit walaupun pembagian *block divide* sedikit mengatasi masalah tersebut. Masalah lainnya adalah, jika di dalam *file* tersebut termuat gambar, gambar tersebut harus diubah terlebih dahulu dengan menggunakan Base64. Selain itu, *file share* yang dikirimkan untuk pihak lain melalui *cloud* seharusnya dapat dienkripsi lagi sehingga *file share* tidak hanya disimpan secara polosan.

UCAPAN TERIMA KASIH

Pertama, penulis ingin mengucapkan syukur kepada Tuhan Yang Maha Esa karena atas berkat-Nya penulis dapat mengikuti mata kuliah IF4020 Kriptografi ini dari awal hingga pada akhirnya selesai dibuat makalah ini. Penulis juga hendak berterima kasih kepada Bapak Rinaldi Munir selaku dosen pengampu mata kuliah terkait atas bimbingan dan pengajaran yang telah diberikan beliau. Terakhir, penulis juga berterima kasih kepada keluarga dan teman-teman penulis yang telah memberi dukungan selama pengerjaan makalah ini dan selama proses belajar mata kuliah IF4020 Kriptografi.

REFERENCES

[1] M. Bellare and P. Rogaway, "Introduction to Modern Cryptography," Introduction to Modern Cryptography, 2005, pp. 16. [Online]. Available: https://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf. [Accessed: 19-Dec-2021].

[2] A. J. Menezes, S. A. Vanstone, and V. O. P. C., *Handbook of Applied Cryptology*, 1st ed. CRC, 1997.

[3] J. F. Steffensen, *Interpolation*. Mineola, NY: Dover Publications, 2006.

[4] Munir, Rinaldi. "Skema Pembagian Data Rahasia," Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2017-2018/Skema-Pembagian-Data-Rahasia-(2018).pdf. [Accessed: 19-Dec-2021].

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

Muhammad Cisco Zulfikar