Implementasi Skema Pembagian Kunci Rahasia untuk Keamanan Akun Bank Bersama Menggunakan Shamir's Secret Sharing

Ikmal Alfaozi - 13520125
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): ikmalalfaozi@gmail.com

***Abstrak*—Keamanan dalam pengelolaan akun bank bersama merupakan tantangan penting yang memerlukan solusi yang dapat melindungi akses terhadap dana dengan efektif. Dalam makalah ini dipaparkan hasil implementasi dan analisis skema pembagian kunci rahasia menggunakan algoritma Shamir's Secret Sharing untuk meningkatkan keamanan akun bank bersama. Algoritma ini memungkinkan pembagian rahasia (misalnya, kunci enkripsi) menjadi beberapa bagian yang didistribusikan kepada pemegang akun, sehingga hanya sejumlah bagian tertentu yang diperlukan untuk mengakses akun tersebut. Implementasi ini mencakup pembangkitan, distribusi, dan rekonstruksi bagian rahasia. Hasil eksperimen menunjukkan bahwa skema ini tidak hanya meningkatkan keamanan akses, tetapi juga memastikan tidak ada satu individu pun yang memiliki kontrol penuh atas akun tersebut.**

Kata Kunci—Keamanan; Kriptografi; Akun Bank Bersama; Shamir's Secret Sharing; Pembagian Kunci Rahasia

I. PENDAHULUAN

Dalam era digital saat ini, keamanan informasi menjadi salah satu isu yang paling krusial, terutama dalam konteks pengelolaan akun bank bersama. Akun bank bersama sering digunakan oleh organisasi, keluarga, atau kelompok tertentu untuk mengelola dana kolektif. Namun, pengelolaan akun ini rentan terhadap berbagai ancaman keamanan seperti pencurian identitas, akses tidak sah, dan penyalahgunaan dana. Oleh karena itu, diperlukan solusi yang dapat menjamin bahwa akses ke dana dalam akun bank bersama hanya dapat dilakukan oleh pihak-pihak yang berwenang dan dengan cara yang aman.

Shamir's Secret Sharing (SSS) adalah salah satu algoritma kriptografi yang dirancang untuk memecahkan masalah ini. Algoritma ini memungkinkan pembagian sebuah rahasia, seperti kunci enkripsi, menjadi beberapa bagian yang terpisah. Hanya kombinasi sejumlah bagian tertentu (*threshold*) yang diperlukan untuk merekonstruksi rahasia tersebut. Dengan menggunakan SSS, akses ke akun bank bersama dapat diamankan dengan memastikan bahwa tidak ada satu individu pun yang memiliki kontrol penuh atas akun tersebut, melainkan memerlukan kolaborasi antara beberapa pemegang bagian rahasia.

Makalah ini bertujuan untuk memaparkan hasil implementasi dan analisis skema pembagian kunci rahasia menggunakan algoritma Shamir's Secret Sharing dalam konteks keamanan akun bank bersama. Dalam makalah ini akan dijelaskan langkah0langkah pembangkitan, distribusi, dan rekonstruksi bagian rahasia serta mengevaluasi keefektifan skema ini dalam meningkatkan keamanan dan integritas akses terhadap akun bank bersama.

II. TINJAUN PUSTAKA

Dalam bidang kriptografi, pembagian kunci rahasia adalah metode yang sangat penting untuk meningkatkan keamanan informasi, terutama dalam konteks akses kolektif terhadap sumber daya yang sensitif, seperti akun bank bersama. Shamir's Secret Sharing (SSS) adalah salah satu skema yang paling dikenal dan digunakan secara luas untuk tujuan ini.

A. Shamir's Secret Sharing

Shamir's Secret Sharing adalah algoritma pembagian kunci rahasia yang diperkenalkan oleh Adi Shamir pada tahun 1979. Algoritma ini didasarkan pada prinsip matematika interpolasi polinomial. Rahasia, seperti kunci enkripsi, dibagi menjadi beberapa bagian yang disebut "*shares*". Hanya sejumlah tertentu dari *shares* (*threshold*) yang diperlukan untuk merekonstruksi rahasia asli [1]. Proses pembagian dan rekonstruksi rahasia dijelaskan sebagai berikut:

- Pembagian Rahasia

  Sebuah polinomial acak dengan derajat $k-1$ dibangkitkan, dengan koefisien konstan polinomial tersebut adalah rahasia yang dibagikan. Nilai-nilai polinomial pada n titik yang berbeda kemudian didistribusikan sebagai *shares*.

- Rekonstruksi Rahasia

  Untuk merekonstruksi rahasia, minimal k *shares* diperlukan. Rahasia dapat diperoleh kembali dengan menggunakan interpolasi Lagrange pada titik-titik yang diberikan oleh *shares* tersebut.

Shamir's Secret Sharing dikenal karena keamanannya yang tinggi dan ketahanannya terhadap kehilangan sebagian dari *shares*. Bahkan jika beberapa *shares* hilang atau dicuri, rahasia tidak dapat direkonstruksi tanpa mencapai ambang batas minimal *shares* yang diperlukan.

B. Penerapan dalam Pengelolaan Akun Bank Bersama

Dalam konteks akun bank bersama, SSS dapat digunakan untuk memastikan bahwa tidak ada satu individu pun yang memiliki kontrol penuh terhadap akses akun. Beberapa studi dan aplikasi telah menunjukkan efektivitas SSS dalam berbagai situasi, termasuk sistem pemungutan suara elektronik dan pengelolaan kunci dalam komputasi awan.

- Keamanan Akses

  Dengan membagi kunci enkripsi untuk akses akun menjadi beberapa *shares*, akses ke akun hanya dapat diperoleh melalui kerja sama antara beberapa pemegang bagian rahasia. Hal ini mengurangi risiko penyalahgunaan dan pencurian identitas karena seorang individu tunggal tidak dapat mengakses akun tanpa kontribusi dari yang lain.

- Redundansi dan Ketahanan

  SSS juga menawarkan keunggulan dalam hal redundansi. Jika salah satu pemegang bagian kehilangan bagiannya, rahasia tetap dapat direkonstruksi selama ambang batas *shares* masih terpenuhi. Ini sangat penting untuk menjaga kontinuitas akses dalam situasi darurat atau kehilangan data.

C. Implementasi Kriptografi dalam Sistem Keuangan

Penggunaan kriptografi dalam sistem keuangan telah menjadi topik penelitian yang luas. Beberapa penelitian telah menunjukkan bahwa implementasi skema pembagian kunci rahasia tidak hanya meningkatkan keamanan, tetapi juga meningkatkan transparansi dan kepercayaan.

- Sistem Pemungutan Suara Elektronik

  Salah satu aplikasi yang serupa dengan pengelolaan akun bank bersama adalah sistem pemungutan suara elektronik. Dalam sistem ini, SSS digunakan untuk menjaga integritas dan kerahasiaan suara, memastikan bahwa hanya suara sah yang dapat dihitung [2].

- Manajemen Kunci dalam *Cloud Computing*

  Pengelolaan kunci dalam lingkungan komputasi awan juga menggunakan skema serupa untuk memastikan bahwa data sensitif terlindungi bahkan jika terjadi kebocoran data atau serangan terhadap server [3].

Tinjauan pustaka ini menunjukkan bahwa Shamir's Secret Sharing adalah solusi yang efektif dan telah terbukti dalam berbagai aplikasi untuk meningkatkan keamanan dan integritas akses terhadap informasi sensitif. Implementasi SSS dalam konteks akun bank bersama menawarkan berbagai keuntungan, termasuk peningkatan keamanan akses, redundansi, dan ketahanan terhadap kehilangan data, menjadikannya pilihan yang ideal untuk pengelolaan dana kolektif yang aman.

III. METODOLOGI

Makalah ini berfokus pada implementasi dan analisis skema pembagian kunci rahasia menggunakan algoritma Shamir's Secret Sharing (SSS) dalam konteks keamanan akun bersama. Metodologi penelitian ini mencakup pembangkitan kunci rahasia, pembagian kunci menggunakan SSS, distribusi kunci kepada pemegang akun, dan rekonstruksi kunci untuk akses akun. Langkah-langkah ini akan diuraikan secara rinci di bawah ini.

A. Pembangkitan Kunci Rahasia

Langkah pertama adalah pembangkitan kunci rahasia yang akan dibagikan. Kunci ini berupa string acak yang digunakan untuk mengakses akun bank bersama. Dalam implementasi ini, pustaka kriptografi digunakan untuk membangkitkan kunci rahasia yang kuat dan aman.

```
import secrets

def generate_secret_key():
    secret_key = secrets.token_hex(16)
    return secret_key
```

Fungsi di atas akan mengembalikan kunci rahasia yang terdiri dari 16 digit heksadesimal.

B. Pembagian Kunci Menggunakan Shamir's Secret Sharing

Langkah kedua adalah membagi kunci rahasia menjadi beberapa bagian menggunakan algoritma Shamir's Secret Sharing. Berikut adalah implementasi fungsi untuk membagi rahasia menggunakan SSS.

```
import random
from sympy import symbols, Integer

def split_secret(secret, n, k):
    coefficients = [secret] + [random.randrange(1,
                   256) for _ in range(k - 1)]
    x = symbols('x')
    polynomial = sum(coeff * x**i for i, coeff in
                 enumerate(coefficients))
    shares = [(i, polynomial.subs(x, i)) for i in
             range(1, n + 1)]
    return shares
```

Fungsi di atas menerima masukan berupa kunci rahasia (*secret*), jumlah bagian (*n*) yang ingin dibuat, dan ambang batas (*k*) yang diperlukan untuk rekonstruksi. Selanjutnya, fungsi tersebut akan membangkitkan *n* bagian kunci. Contoh salah satu bagian kunci yang dibangkitkan adalah (1, 448733009329002055712965803603310429468553318351763386670947117807702818421 91).

C. Distribusi Kunci keapada Pemegang Akun

Bagian-bagian kunci yang telah dihasilkan akan didistribusikan kepada pemegang akun. Setiap pemegang akun menerima satu bagian kunci. Distribusi ini dapat dilakukan melalui saluran komunikasi yang aman untuk memastikan

bahwa bagian kunci tidak jatuh ke tangan yang salah. Berikut adalah contoh simulasi distribusi bagian kunci.

```
# Simulasi distribusi bagian kunci
pemegang_akun = ["Alice", "Bob", "Charlie", "Dave",
                 "Eve"]
distribusi = dict(zip(pemegang_akun, shares))

print("Distribusi bagian kunci kepada pemegang
      akun:")
for pemegang, bagian in distribusi.items():
    print(f"{pemegang}: {bagian}")
```

D. Rekonstruksi Kunci untuk Akses Akun

Ketika akses ke akun bank bersama diperlukan, minimal sejumlah *k* bagian kunci harus dikumpulkan untuk merekonstruksi kunci rahasia. Proses ini menggunakan interpolasi polinomial untuk menggabungkan bagian-bagian kunci menjadi kunci yang sebenarnya.

```
from sympy import symbols, Integer

# Fungsi untuk merekonstruksi rahasia
def recover_secret(shares, threshold):
    x_vals, y_vals = zip(*shares[:threshold])
    x = symbols('x')
    polynomial = lagrange_interpolation(x_vals,
                 y_vals)
    secret = polynomial.subs(x, 0)
    return int(secret)

# Fungsi untuk melakukan interpolasi Lagrange
def lagrange_interpolation(x_vals, y_vals):
    x = symbols('x')
    polynomial = 0

    for j in range(len(x_vals)):
        numerator = Integer(1)
        denominator = Integer(1)
        for i in range(len(x_vals)):
            if i != j:
                numerator *= (x - x_vals[i])
                denominator *= (x_vals[j] -
                                x_vals[i])
        term = y_vals[j] * numerator / denominator
        polynomial += term

    return polynomial.expand()
```

E. Eksperimen dan Analisis

Langkah terakhir adalah melakukan eksperimen untuk mengevaluasi keefektifan skema ini dalam meningkatkan keamanan akun bank bersama. Eksperimen ini melibatkan simulasi skenario akses akun, kehilangan bagian kunci, dan upaya rekonstruksi kunci dengan berbagai kombinasi bagian.

- Keamanan Akses
  Uji coba dilakukan untuk memastikan bahwa tanpa ambang batas minimal bagian kunci, kunci rahasia yang asli tidak dapat direkonstruksi.
- Redundansi
  Uji coba dilakukan dengan kehilangan beberapa bagian kunci dan memastikan bahwa rahasia masih dapat direkonstruksi selama ambang batas tercapai.
- Efisiensi
  Waktu dan sumber daya yang diperlukan untuk pembagian dan rekonstruksi kunci diukur untuk memastikan bahwa skema ini dapat diimplementasikan secara praktis dalam lingkungan akun bank bersama.

IV. IMPLEMENTASI

Pada bagian ini, akan dijelaskan implementasi sistem transfer dana bank menggunakan Shamir's Secret Sharing (SSS) untuk meningkatkan keamanan dan kontrol dalam pengelolaan akun bank bersama. Implementasi ini mencakup beberapa tahap, yaitu, pembangkitan kunci rahasia, pembagian kunci menggunakan SSS, distribusi kunci kepada pemegang akun, dan rekonstruksi kunci untuk akses akun. Seluruh implementasi dilakukan menggunakan bahasa pemrograman Python dan antarmuka pengguna berbasis web dengan Streamlit.

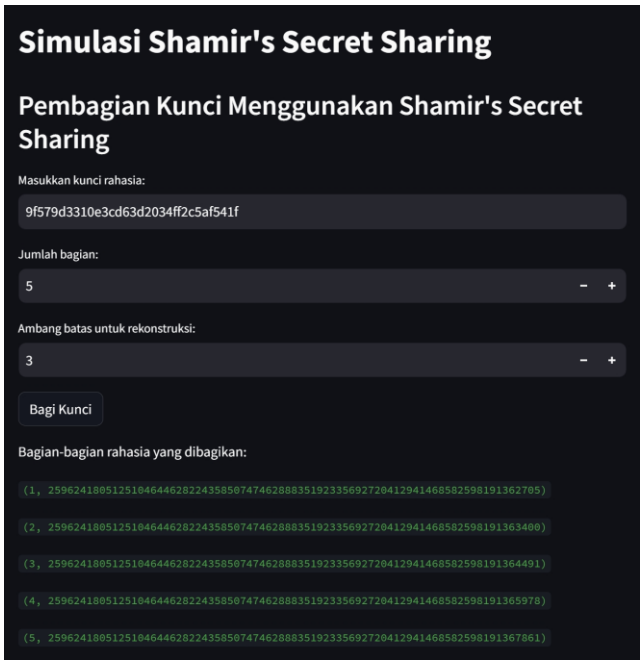
A. Pembangkitan Kunci

Tahap pertama dalam implementasi adalah pembangkitan kunci rahasia yang akan digunakan untuk mengamankan akses akun bank. Kunci rahasia ini kemudian akan dibagi menjadi beberapa bagian (*shares*) menggunakan algoritma Shamir's Secret Sharing.

Gambar 1. Simulasi pembangkitan kunci rahasia

B. Pembagian Kunci Menggunakan Shamir's Secret Sharing

Kunci rahasia yang telah dibangkitkan dibagi menjadi beberapa bagian menggunakan algoritma Shamir's Secret Sharing. Algoritma ini membagi rahasia menjadi *n shares*, dengan minimal *k shares* diperlukan untuk merekonstruksi kunci rahasia asli.



Gambar 2. Simulasi pembagian kunci menggunakan SSS

C. *Distribusi Kunci Menggunakan Shmair's Secret Sharing*

Shares yang dihasilkan kemudian didistribusikan kepada para pemegang akun. Pada implementasi ini, distribusi dilakukan secara manual dengan mencatat shares untuk setiap pengguna.

Gambar 3. Simulasi distribusi kunci kepada pemegang akun

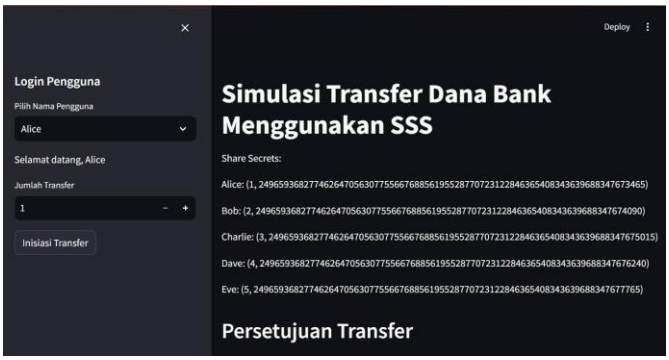
D. *Rekonstruksi Kunci untuk Akses Akun*

Untuk mengakses akun bank, minimal *k* pengguna harus menyetujui transfer dan memberikan *shares* mereka untuk merekonstruksi kunci rahasia menggunakan interpolasi Lagrange.

Gambar 4. Simulasi rekonstruksi kunci untuk akses akun

E. *Antarmuka untuk Simulasi Transfer Dana Bank*

Streamlit digunakan untuk membuat antarmuka pengguna berbasis web yang memungkinkan pengguna untuk login, menginisiasi transfer, menyetujui atau menolak transfer, dan merekonstruksi kunci rahasia.



Gambar 4. Antarmuka untuk Simulasi Transfer Dana Bank

V. HASIL DAN PEMBAHASAN

Pada bagian ini, dilakukan pengujian terhadap implementasi beserta analisis terhadap hasil pengujian yang didapatkan.

A. *Pengujian Skema Shamir's Secret Sharing*

Untuk memastikan keefektifan dan efisiensi skema Shamir's Secret Sharing, beberapa pengujian dilakukan:

1. Keamanan Akses dan Redundansi
   Uji coba keamanan akses dilakukan untuk memastikan bahwa tanpa ambang batas minimal bagian kunci, kunci rahasia yang asli tidak dapat direkonstruksi. Sementara itu, uji coba redundansi dilakukan dengan kehilangan beberapa bagian kunci dan memastikan bahwa rahasia masih dapat direkonstruksi selama ambang batas tercapai.

Tabel 1. Hasil Rekonstruksi Kunci Rahasia

Kunci Rahasia	***Threshold***	***Shares* (*k*)**	**Hasil Rekontruksi**
7280d9a8fb5bb84ffbd 9fd07147bb0b4	3	5	7280d9a8fb5bb84ffbd 9fd07147bb0b4
7280d9a8fb5bb84ffbd 9fd07147bb0b4	3	4	7280d9a8fb5bb84ffbd 9fd07147bb0b4
7280d9a8fb5bb84ffbd 9fd07147bb0b4	3	3	7280d9a8fb5bb84ffbd 9fd07147bb0b4
7280d9a8fb5bb84ffbd 9fd07147bb0b4	3	2	7280d9a8fb5bb84ffbd 9fd07147bb0a
7280d9a8fb5bb84ffbd 9fd07147bb0b4	3	1	7280d9a8fb5bb84ffbd 9fd07147bb0cy

Berdasarkan eksperimen di atas, dapat disimpulkan beberapa poin terkait keamanan akses dan redundansi:

- Hasil uji coba di atas menunjukkan bahwa rekonstruksi kunci gagal jika jumlah *shares* yang tersedia kurang dari ambang batas yang ditentukan. Hal ini membuktikan keamanan skema ini dalam mencegah akses tanpa persetujuan yang cukup.
- Hasil uji coba di atas menunjukkan bahwa meskipun beberapa bagian kunci hilang, kunci rahasia masih dapat direkonstruksi selama jumlah *shares* yang tersisa memenuhi ambang batas. Hal ini menunjukkan ketahanan skema terhadap kehilangan sebagian shares.

2. Efisiensi
   Waktu yang diperlukan untuk pembagian dan rekonstruksi kunci diukur untuk memastikan bahwa skema ini dapat diimplementasikan secara praktis dalam lingkungan akun

bank bersama. Kunci rahasia yang digunakan adalah 7280d9a8fb5bb84ffbd9fd07147bb0b4.

Tabel 2. Hasil pengujian efisiensi pembagian dan rekonstruksi kunci

Total *Shares* (*n*)	***Threshold* (*k*)**	**Waktu Pembagian kunci (*second*)**	**Waktu Rekonstruksi (*second*)**
100	25	1.126752	2.350524
100	50	2.194237	20.727431
100	75	3.292854	66.552331
200	25	2.178888	2.335517
200	50	4.259512	20.852085
200	75	6.679863	65.948645
200	100	8.699462	151.604938
200	150	13.068565	489.524322
300	25	3.411583	2.336865
300	50	6.47388	20.572922
300	75	10.090981	68.552407
300	100	13.222865	424.840017
300	150	40.995712	732.801353

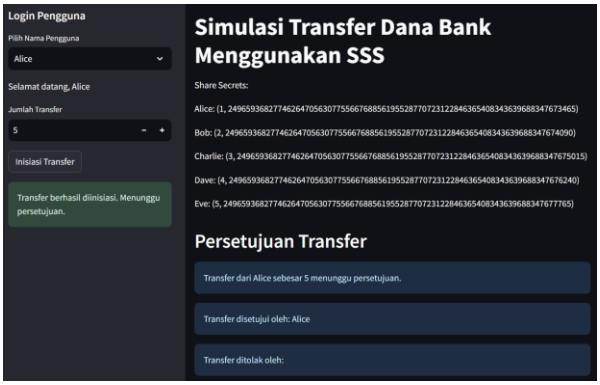
Berdasarkan hasil eksperimen yang dilakukan untuk mengukur waktu pembagian kunci dan rekonstruksi kunci dengan berbagai kombinasi jumlah total bagian kunci (*n*) dan ambang batas rekonstruksi (*k*), dapat disimpulkan beberapa poin penting terkait efisiensi skema Shamir's Secret Sharing:

- Peningkatan waktu pembagian kunci dengan bertambahnya bagian kunci (*n*) dan ambang batas (*k*).
- Peningkatan waktu rekonstruksi dengan bertambahnya ambang batas (*k*)
- Waktu pembagian kunci relatif lebih rendah dibandingkan dengan waktu rekonstruksi kunci, terutama ketika nilai ambang batas (*k*) mendekati jumlah total kunci (*n*).
- Peningkatan ambang batas rekonstruksi (*k*) menyebabkan peningkatan eksponensial dalam waktu rekonstruksi kunci, yang bisa menjadi kurang praktis untuk implementasi dalam skenario nyata dengan kebutuhan rekonstruksi cepat.
- Untuk implementasi dalam lingkungan akun bank bersama, memilih ambang batas rekonstruksi (*k*) yang lebih rendah namun tetap aman (seperti *k*=25 atau *k*=50) akan memberikan keseimbangan yang baik antara keamanan dan efisiensi waktu.

B. Simulasi Transfer Dana Menggunakan Shamir's Secret Sharing (SSS)

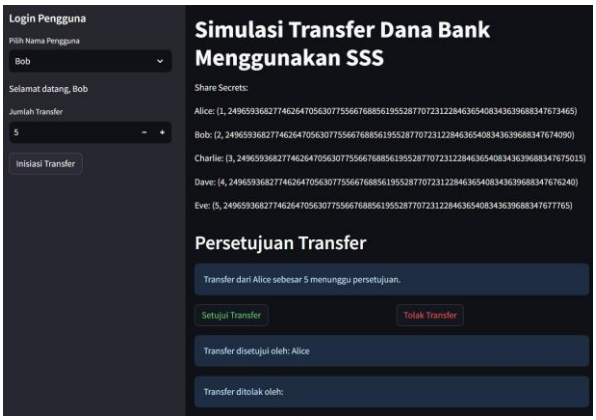
Pada simulasi transfer dana menggunakan SSS yang telah dibuat, terdapat beberapa tahapan utama yang berhasil diimplementasikan dan diuji. Berikut adalah hasil dari setiap tahapannya:

1. Login Pengguna
   - Pengguna dapat login menggunakan nama mereka yang telah terdaftar.
   - Sistem menampilkan sambutan sesuai dengan nama pengguna yang login.
2. Inisiasi Transfer
   - Pengguna yang login dapat menginisiasi transfer dengan memasukkan jumlah dana yang ingin ditransfer.
   - Sistem mencatat detail transfer dan menunggu persetujuan dari pengguna lain.



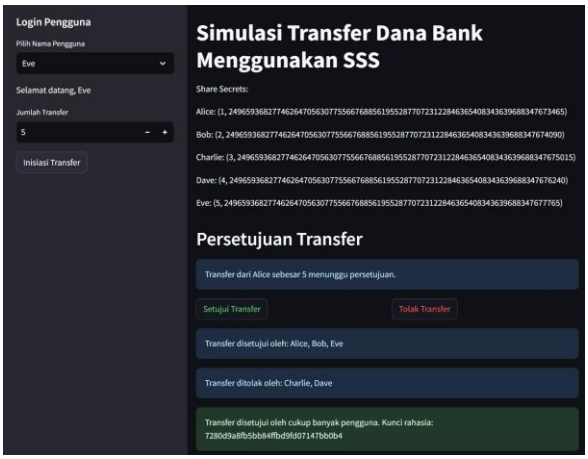
Gambar 5. Hasil simulasi inisiasi transfer oleh pengguna

3. Notifikasi dan Persetujuan Transfer
   - Pengguna lain yang login setelah transfer diinisiasi dapat melihat detail transfer yang menunggu persetujuan.
   - Pengguna dapat memilih untuk menyetujui atau menolak transfer dengan memverifikasi menggunakan share secret mereka.



Gambar 6. Hasil simulasi notifikasi dan persetujuan transfer

4. Rekonstruksi Kunci Rahasia
   - Jika jumlah persetujuan mencapai ambang batas (*threshold*), sistem akan merekonstruksi kunci rahasia menggunakan interpolasi Lagrange.
   - Transfer dinyatakan berhasil jika kunci rahasia berhasil direkonstruksi, sebaliknya transfer dibatalkan jika tidak ada cukup persetujuan.



Gambar 7. Hasil rekonstruksi kunci rahasia untuk transfer dana

C. Kelebihan dan Kelemahan Shamir's Secret Sharing

Shamir's Secret Sharing (SSS) terbukti sebagai metode yang efektif untuk memastikan keamanan dalam pengelolaan akun bank bersama. Dalam implementasi ini, SSS digunakan untuk membagi kunci rahasia menjadi beberapa bagian (*shares*) yang didistribusikan kepada para pemegang akun. Beberapa poin terkait keamanan dan implementasi adalah:

1. Keamanan Pembagian Kunci
   - Kunci rahasia yang dibagi menjadi beberapa *shares* tidak dapat direkonstruksi oleh satu pengguna saja, melainkan memerlukan kombinasi dari beberapa *shares* untuk mendapatkan kunci asli.
   - Hal ini meningkatkan keamanan karena tidak ada satu individu yang memiliki kontrol penuh atas dana di akun tersebut.
2. Proses Verifikasi
   - Pengguna yang berpartisipasi dalam persetujuan transfer harus memverifikasi menggunakan *share secret* mereka.
   - Verifikasi ini memastikan bahwa hanya pengguna yang sah yang dapat memberikan persetujuan untuk transfer dana.
3. Ambang Batas (*Threshold*)
   - *Threshold* yang ditetapkan untuk rekonstruksi kunci rahasia memberikan fleksibilitas dalam menentukan berapa banyak persetujuan yang diperlukan untuk menyelesaikan transfer.
   - Ambang batas yang lebih tinggi meningkatkan keamanan tetapi juga dapat meningkatkan kompleksitas dalam memperoleh persetujuan yang cukup.

Meskipun implementasi SSS memberikan banyak keuntungan dalam hal keamanan, terdapat beberapa kelemahan dan tantangan yang perlu dipertimbangkan:

1. Kompleksitas Implementasi
   - Implementasi algoritma interpolasi Lagrange dan distribusi *shares* memerlukan pemahaman yang mendalam tentang matematika dan kriptografi.
   - Kesalahan dalam implementasi dapat menyebabkan kegagalan dalam rekonstruksi kunci rahasia.
2. Pengelolaan Shares
   - *Shares* harus dikelola dengan hati-hati untuk memastikan tidak hilang atau dicuri.
   - Pengguna harus memahami pentingnya menjaga kerahasiaan *shares* mereka.
3. Responsivitas Pengguna
   - Sistem memerlukan respons dari beberapa pengguna untuk menyetujui transfer, yang dapat menyebabkan penundaan jika pengguna tidak segera memberikan persetujuan.

D. Penggunaan di Dunia Nyata

Penggunaan Shamir's Secret Sharing dalam pengelolaan akun bank bersama menunjukkan potensi besar dalam meningkatkan keamanan transaksi keuangan. Beberapa aplikasi nyata yang dapat memanfaatkan metode ini antara lain:

1. Akun Bank Perusahaan
   Perusahaan dapat menggunakan SSS untuk memastikan bahwa transaksi besar memerlukan persetujuan dari beberapa anggota tim keuangan atau manajemen.
2. Keuangan Keluarga
   Keluarga yang mengelola dana bersama dapat menggunakan SSS untuk memastikan bahwa setiap anggota keluarga memiliki kontrol yang setara dan tidak ada satu anggota yang dapat mengakses dana tanpa persetujuan yang lain.
3. Platform Investasi
   Platform investasi dapat menggunakan SSS untuk mengelola dana yang diinvestasikan oleh beberapa investor, memastikan bahwa keputusan investasi memerlukan persetujuan dari mayoritas investor.

VI. KESIMPULAN

Implementasi Shamir's Secret Sharing dalam simulasi transfer dana bank menunjukkan bahwa metode ini dapat meningkatkan keamanan dan kontrol dalam pengelolaan akun bersama, SSS memastikan bahwa tidak ada satu individu yang memiliki kontrol penuh atas dana. Namun, tantangan dalam implementasi dan pengelolaan s*hares* harus diatasi untuk memastikan keberhasilan penerapan metode ini dalam situasi dunia nyata. Uji coba yang dilakukan menunjukkan bahwa skema ini efektif dalam menjaga keamanan akses, memiliki redundansi yang baik, dan efisiensi untuk diimplementasikan.

UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur yang sebesar-besarnya kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga makalah berjudul "Implementasi Shamir's Secret Sharing untuk Keamanan Akses Akun Bank Bersama" dapat diselesaikan dengan baik. Penulis mengucapkan terima kasih kepada dosen pengajar IF4020 Kriptografi, Dr. Rinaldi Munir, S.T, M.T., yang telah memberikan bimbingan dan ilmu terkait materi Kriptografi ini, khususnya pada materi secret sharing. Penulis juga mengucapkan terima kasih banyak kepada para penulis sumber referensi yang digunakan pada makalah ini yang telah memberikan ilmu yang dibutuhkan penulis untuk menyelesaikan makalah ini.

PRANALA KODE PROGRAM IMPLEMETNASI

Kode program implementasi dapat diakses pada pranala berikut.

https://github.com/ikmalalfaozi/secure-shared-bank-account

REFERENCES

[1] Munir, Rinaldi, Skema Pembagian Data Rahasia Shamir Secret Sharing (Bahan Kuliah IF4020 Kriptografi). 2024. Dakses pada 8 Juni 2024. https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2023-2024/35-Skema-Pembagian-Data-Rahasia-2024.pdf

[2] Tejedor-Romero, Marino, et al. "Distributed remote e-voting system based on Shamir's secret sharing scheme." Electronics 10.24 (2021): 3075.

[3] Bentajer, Ahmed, et al. "Secure Cloud Key Management based on Robust Secret Sharing." CS and IT Conference Proceedings. Vol. 11. No. 9. 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, 12 Juni 2024

Ikmal Alfaozi
13520125