Analisis Protokol Suara Mayoritas Terdesentralisasi Berbasis Pembagian Rahasia Shamir

Asif Hummam Rais 13517099
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
hashshura@gmail.com, 13517099@std.stei.itb.ac.id

***Abstract*—Pemungutan suara adalah pertarungan sengit antara beberapa kubu yang membuka peluang untuk melakukan kecurangan. Untuk mengantisipasi kecurangan dari sisi otoritas, dapat digunakan teori berlandasan kriptografi. Dalam kriptografi modern, pembagian rahasia homomorfik adalah tipe algoritme pembagian rahasia yang mengenkripsi suatu pesan rahasia melalui sistem enkripsi yang memiliki karakteristik homomorfisme, yaitu transformasi antara satu struktur aljabar ke bentuk yang lain dalam jenis yang sama dengan tetap mempertahankan strukturnya. Hal ini berarti bahwa setiap manipulasi pada data original akan memiliki perubahan yang bersesuaian dengan data yang ditransformasi. Salah satu pembagian rahasia homomorfik adalah pembagian rahasia Shamir. Makalah ini mencoba membangun protokol suara mayoritas terdesentralisasi sederhana berbasis sistem pembagian rahasia Shamir dan memberikan analisis mengenai kompleksitas maupun kualitas dari protokol yang dibangun.**

***Keywords*—suara mayoritas, homomorfisme, pembagian rahasia Shamir, desentralisasi.**

I. Pendahuluan

Dalam beberapa tahun terakhir, beberapa negara telah melakukan pemilihan umum yang menjadi agenda per empat atau lima tahun. Pemilihan umum ini ditujukan untuk menjadi proses demokrasi dalam memilih presiden dan wakil presiden suatu republik tertentu untuk periode selanjutnya. Di Indonesia sendiri, menurut Pasal 7 Undang-Undang Dasar Negara Republik Indonesia Tahun 1945, disebutkan di dalamnya, "Presiden dan Wakil Presiden memegang jabatan selama lima tahun, dan sesudahnya dapat dipilih kembali dalam jabatan yang sama, hanya untuk satu kali masa jabatan." Dalam artian, sudah sewajarnya bagi Negara Indonesia dan rakyatnya untuk berpartisipasi pada pemilihan umum yang terjadwal ini.

Namun, sebagai polemik dan isu yang tidak hanya berlaku di Indonesia, masalah keyakinan akan terjadinya kecurangan sering menjadi momok yang dapat memakan waktu dan biaya berlebih untuk dapat menyelesaikannya. Kemunculan dari keyakinan masyarakat akan hal tersebut dapat disebabkan oleh beberapa faktor. Salah satunya adalah penggunaan pena-dan-kertas sebagai medium dalam menentukan dan memberikan suara pilihannya.

Kriptografi, dengan empat aspek keamanan informasinya, memiliki potensi akan penggunaan yang lebih luas lagi. Dengan sifat *non-repudiation* yang perlu diperhatikan, kriptografi dapat digunakan sebagai metode yang melandaskan pembangunan protokol pemilihan suara untuk mengurangi kemungkinan terjadinya kecurangan maupun kelalaian dari sisi otoritas.

II. Dasar Teori

A. Kriptografi

Kriptografi, atau kriptologi (dari bahasa Yunani kuno: κρυπτός, dibaca *kryptós*, yang berarti "tersembunyi" dan γράφειν *graphein* yang berarti "menulis" atau -λογία *logia*, "belajar"), adalah praktik dan pengetahuan mengenai teknik untuk berkomunikasi secara aman dengan adanya pihak ketiga atau *adversary* (Rivest, 1990). Secara umum, kriptografi berbicara tentang mengkonstruksi dan menganalisis protokol yang dapat mencegah pihak ketiga atau pihak publik untuk dapat membaca pesan privat (Bellare et al., 2005).

Kriptografi modern membahas peran dan penggunaan kriptografi pada sistem informasi. Pengetahuan pada kriptografi modern berpotongan dengan disiplin ilmu matematika, ilmu komputer, teknik elektro, ilmu komunikasi, dan fisika. Pada kriptografi modern, terdapat empat aspek yang berkaitan dengan keamanan informasi yaitu *confidentiality* (kerahasiaan), *data integrity* (kebenaran), *authentication* (otentikasi atau identifikasi aktor), dan *non-repudiation* (tidak dapat disangkal) (Menezes et al., 1997).

Pada konteks kriptografi, terdapat tiga istilah mengenai objek yang dipertukarkan pada jalur komunikasi:

1. Pesan

Pesan adalah bentuk dasar dari objek informasi yang dapat dipahami oleh tujuan dan target dari pengiriman informasi. Pesan dapat berbentuk apapun (tulisan, gambar, video, dll.) dan merupakan bentuk asli yang ditujukan untuk dibaca atau dipahami oleh pihak penerima.

2. Plainteks

Plainteks adalah bentuk semula dari sebuah pesan sebelum dilakukan enkripsi, yaitu perubahan bentuk plainteks ke cipherteks melalui sebuah fungsi cipher. Plainteks juga merupakan bentuk hasil dekripsi, yaitu fungsi untuk mengembalikan hasil enkripsi ke bentuk semula, dari sebuah cipherteks.

3. Cipherteks

Cipherteks adalah bentuk informasi yang tidak dapat dipahami oleh tujuan dan target dari pengiriman informasi (misalnya manusia). Cipherteks merupakan hasil enkripsi dari sebuah plainteks yang melewati fungsi cipher. Cipherteks

digunakan sebagai pesan yang akan disalurkan melalui jalur ketiga, dengan tujuan kerahasiaan pesan tetap terjaga dengan pihak ketiga tidak dapat memahami isi dari informasi yang disalurkan.

B. Kriptosistem

Satu atau lebih primitif kriptografis sering digunakan untuk membangun algoritme yang lebih kompleks. Sistem kompleks ini disebut dengan sistem kriptografis atau *kriptosistem*. Kriptosistem didesain untuk menyediakan fungsionalitas yang spesifik, misalnya enkripsi kunci publik, dengan tetap menjamin beberapa properti keamanan tertentu, misalnya *chosen-plaintext attack* (CPA).

Kriptosistem menggunakan properti dari primitif-primitif kriptografis untuk mendukung kapasitas keamanan dari sistem itu sendiri. Karena perbedaan pada istilah primitif dan kriptosistem cenderung tidak terdefinisi dengan baik, kriptosistem yang "canggih" dapat terbentuk atas kombinasi dari beberapa kriptosistem yang cakupannya lebih sempit. Dalam banyak kasus, struktur kriptosistem berkaitan dengan komunikasi bolak-balik antara dua atau lebih pihak di sebuah ruang atau melewati waktu. Kriptosistem yang demikian biasa disebut dengan *protokol kriptografis*.

C. Polinomial

Pada bidang matematika, sebuah polinomial didefinisikan sebagai ekspresi yang terdiri atas variabel (atau *bentuk tak tentu*) dan koefisien, yang melibatkan hanya operasi penjumlahan, pengurangan, perkalian, dan perpangkatan bilangan bulat non-negatif dari variabel.

Fungsi polinomial adalah sebuah fungsi yang dapat didefinisikan dengan mengevaluasi sebuah polinomial. Persisnya, sebuah fungsi f terdiri atas satu argumen dari domain yang diberikan adalah sebuah fungsi polinomial jika terdapat sebuah polinomial

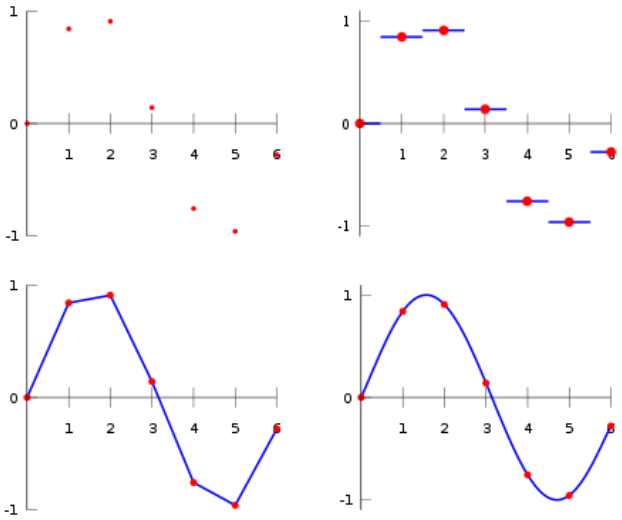
$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

yang dievaluasi menjadi $f(x)$ untuk seluruh x pada domain f (di sini, n adalah bilangan non-negatif dan a_0, a_1, a_2, ..., a_n adalah koefisien konstanta). Derajat polinomial mengacu kepada nilai pangkat tertinggi dari setiap *monomial* (istilah individual) dengan koefisien tidak nol. Dalam kasus fungsi polinomial di atas, derajat polinomialnya adalah senilai n.

Gambar 1. Grafik dari fungsi polinomial dengan derajat 3

D. Interpolasi

Pada bidang matematika dengan subbidang analisis numerik, interpolasi didefinisikan sebagai sebuah tipe estimasi, yaitu sebuah metode untuk mengkonstruksi titik data baru yang masuk ke dalam jarak set diskrit berisi titik data yang diketahui (Sheppard, 1911). Salah satu metode yang paling sederhana dari interpolasi adalah interpolasi *nearest-neighbor* (tetangga terdekat) yang menggunakan nilai $f(x)$ dengan nilai x terdekat yang diketahui. Metode sederhana lainnya adalah interpolasi linear yang menghubungkan seluruh titik data diketahui dengan sebuah garis menggunakan persamaan garis. Generalisasi dari interpolasi linear adalah interpolasi polinomial, di mana akan dicari satu (dan hanya ada satu) fungsi polinomial dengan derajat n - 1 jika terdapat n titik data yang semula diketahui.



Gambar 2. Tujuh titik yang semula diketahui (kiri atas), interpolasi *nearest-neighbor* (kanan atas), interpolasi linear (kiri bawah), dan interpolasi polinomial (kanan bawah).

E. Polinomial Lagrange

Dalam menyelesaikan permasalahan interpolasi, salah satu algoritme yang umum digunakan adalah interpolasi menggunakan polinomial Lagrange. Untuk sebuah set titik yang diberikan (x_j, y_j) dengan tidak ada nilai x_j yang sama, polinomial Lagrange adalah polinomial dengan derajat paling rendah yang akan mengasumsikan setiap nilai x_j dengan x_j, sehingga fungsi tersebut akan berimpit dengan setiap titik.

Polinomial Lagrange didefinisikan sebagai berikut. Diberikan sebuah set yang terdiri atas $k + 1$ titik data (x_0, y_0), ..., (x_j, y_j), ..., (x_k, y_k) dengan tidak ada dua x_j yang sama, polinomial interpolasi dalam bentuk Lagrange adalah kombinasi linear

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

dari basis polinomial Lagrange

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

dengan $0 \le j \le k$.

F. Pembagian Rahasia Shamir

Pembagian rahasia Shamir adalah algoritme dalam kriptografi yang diciptakan oleh Adi Shamir. Algoritme ini adalah salah satu dari bentuk pembagian rahasia, di mana sebuah rahasia dibagi menjadi beberapa bagian, dan setiap partisipan mendapatkan sebuah bagian unik masing-masing. Untuk merekonstruksi pesan rahasia, sejumlah bagian minimum dibutuhkan. Dalam skema *threshold*, angka ini kurang dari jumlah total bagian yang ada. Jika tidak, seluruh partisipan dibutuhkan untuk merekonstruksi rahasia semula.

Ide dasar dari skema *threshold* Shamir berlandaskan pada bahwa 2 titik sudah cukup untuk mendefinisikan garis, 3 titik untuk parabola, 4 titik untuk kurva kubik, dan seterusnya. Dalam artian, dibutuhkan k titik untuk mendefinisikan polinomial dengan derajat k - 1. Jika kita menggunakan skema *threshold* (k, n) untuk membagi rahasia S, tanpa adanya generalitas diasumsikan sebagai elemen pada bidang terbatas F berukuran P dengan $0 < k \leq n < P$; $S < P$, dan P bilangan prima. Pilih k - 1 bilangan positif acak $a_1, ..., a_{k-1}$ dengan $a_i < P$ dan $a_0 = S$. Bangun 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, misalnya pada set $i = 1, ..., n$ untuk mengambil $(i, f(i))$, Setiap partisipan diberikan sebuah titik (bilangan bulat masukan tidak nol pada polinomial dan bilangan bulat keluaran yang bersesuaian) bersamaan dengan bilangan prima yang mendefinisikan bidang terbatas yang digunakan. Dengan subset berukuran k, dapat ditemukan koefisien polinomial menggunakan interpolasi. Rahasia yang disimpan terletak pada bilangan konstanta a_0.

G. Prinsip Elektoral Indonesia

Pemilihan umum di Indonesia mengikuti enam prinsip yang biasa disingkat dengan istilah *luber-jurdil*. Empat prinsip pertama, "luber", diadopsi pada zaman orde baru saat pemilihan umum tahun 1971. Setelah reformasi tahun 1998, mengikuti liberalisasi politik, dua prinsip tambahan yaitu "jurdil" diadopsi untuk pertama kalinya pada pemilihan umum tahun 1999 (Nita, 2019). Adapun prinsip tersebut antara lain

1. **L**angsung: pemilih harus memberikan suaranya tanpa ada penengah;
2. **U**mum: seluruh Warga Negara Indonesia yang memenuhi kriteria pemilih dapat menggunakan hak pilihnya tanpa pembatasan;
3. **Be**bas: pemilih dapat memilih menggunakan kata hatinya sendiri tanpa adanya paksaan untuk memilih kandidat tertentu;
4. **R**ahasia: pemilihan anonim dijamin, sehingga pilihan pemilih tidak diketahui oleh siapapun kecuali dirinya sendiri;
5. Ju**jur**: pemilih, kandidat, dan institusi elektoral harus melakukan kewajibannya dengan sepenuh kejujuran;
6. A**dil**: pemilih dan kandidat memiliki perlakuan yang sama di mata hukum, dengan tidak ada pemilih atau kandidat tertentu menerima perlakuan preferensial maupun diskriminatif.

III. Skema Arsitektur

Asumsikan sebuah komunitas menginginkan untuk menjalankan sebuah pemilihan berbasis suara mayoritas dengan dua pilihan suara berbeda, namun mereka ingin meyakinkan bahwa penghitung suara tidak dapat berbohong mengenai hasilnya. Menggunakan pembagian rahasia Shamir, setiap anggota dari komunitas dapat menambahkan pilihannya dalam bentuk yang dibagi ke bagian-bagian, setiap bagian diserahkan kepada penghitung suara yang berbeda. Bagian-bagian ini didesain sedemikian sehingga penghitung suara tidak dapat memprediksi efek dari perubahan dari bagian yang ia peroleh kepada hasil akhirnya, sehingga hal ini dapat memastikan bahwa mereka tidak akan melakukan pemalsuan nilai suara. Ketika seluruh suara sudah diperoleh, penghitung suara menggabungkannya dan merekonstruksi hasil agregat pemilihan suara yang sudah dilakukan.

Lebih detailnya dijelaskan pada poin-poin berikut. Asumsikan bahwa pada pemilihan suara, terdapat:

- Dua kemungkinan hasil akhir, yaitu pilihan pertama atau pilihan kedua (misalnya *ya / tidak*). Pilihan itu direpresentasikan dengan nilai +1 dan -1.
- Sejumlah otoritas k yang akan menghitung hasil suara.
- Sejumlah pemilih n yang akan memberikan nilai suara.

Keberjalanan arsitektur adalah sebagai berikut.

1. Sebelum pemilihan dimulai, setiap otoritas penghitung suara membangkitkan kunci publik x_k menggunakan metode apapun.
2. Setiap pemilih akan mengenkode pilihannya ke bentuk polinomial p_n dengan aturan berikut: polinomial itu memiliki derajat k - 1, nilai konstantanya bernilai +1 atau -1 (sesuai pilihan yang disuarakan), serta nilai koefisiennya dibangkitkan secara acak.
3. Setiap pemilih menghitung nilai polinomial p_n untuk setiap kunci publik otoritas x_k, menghasilkan k titik untuk setiap otoritas.

Dalam bahasa pemrograman Python, langkah 2 dan 3 dijelaskan pada potongan kode berikut.

```
import random

MAX_SIZE = 10**5

def compute_polynom(x, coeff):
    return sum([x**(len(coeff) - i - 1) *
coeff[i] for i in range(len(coeff))])

def generate_coeff(k, choice):
    coeff = [random.randrange(0, MAX_SIZE)
for _ in range(k - 1)]
    coeff.append(choice)
    return coeff
```

```
def encode_vote(public_keys, choice):
    coeff =
generate_coeff(len(public_keys), choice)
    shares = []
    for x in public_keys:
        shares.append([x,
compute_polynom(x, coeff)])
    return shares
```

4. Setiap pemilih mengirimkan nilai yang didapatkan dari enkode pada langkah 3 ke masing-masing otoritas yang memegang kunci publik bersesuaian.
5. Setiap otoritas menerima nilai yang diterima dari langkah 4. Karena setiap otoritas hanya mendapatkan satu nilai dari setiap pemilih, dia tidak dapat memprediksi nilai yang dipilih oleh pemilih tersebut. Selain itu, otoritas tidak dapat memprediksi apabila melakukan perubahan, efek apa yang akan diberikan pada hasil akhirnya.
6. Setelah seluruh pemilih telah menyuarakan pilihannya, setiap otoritas k menghitung nilai total dari nilai-nilai yang diperoleh pada langkah 5 dan mengumumkan nilai total yang ia miliki yaitu A_k.
7. Karena ada k nilai total (A_k), ketika nilai itu digabungkan menjadi satu, akan membentuk polinomial unik baru $P(x)$ yaitu nilai jumlah dari seluruh polinomial pemilih: $P(x) = p_1(x) + p_2(x) + ... + p_n(x)$. Nilai konstanta pada polinomial ini adalah jumlah dari keseluruhan nilai suara semula yang diberikan. Jika nilainya positif, maka suara mayoritas memihak pilihan +1, sedangkan jika bernilai negatif, maka suara mayoritas memihak pilihan -1. Penjelasan ini diilustrasikan lebih lanjut pada tabel berikut.

	***voter*$_1$**	***voter*$_2$**	**...**	***voter*$_n$**	
***otoritas*$_1$**	$p_1(x_1)$	$p_2(x_1)$	...	$p_n(x_1)$	$total(x_1)$
***otoritas*$_2$**	$p_1(x_2)$	$p_2(x_2)$	...	$p_n(x_2)$	$total(x_2)$
...	...	...	...	...	...
***otoritas*$_k$**	$p_1(x_k)$	$p_2(x_k)$	...	$p_n(x_k)$	$total(x_k)$
	$p_1(x)$	$p_2(x)$	...	$p_n(x)$	$total(x)$

Tabel 1. Ilustrasi protokol pemilihan. Perhatikan bahwa pada pembagian rahasia Shamir, berlaku sifat homomorfik sehingga menjumlahkan nilai *total* setiap otoritas akan memiliki hasil sama seperti menjumlahkan seluruh nilai polinomial per pemilih (yang nilai konstantanya adalah pilihan suara).

Dalam bahasa pemrograman Python, untuk menemukan nilai konstanta yaitu hasil akhir atau total hasil pemilihan suara menggunakan interpolasi polinomial Lagrange dijelaskan pada potongan kode berikut.

```
from decimal import Decimal

def reconstruct(shares):
    sums = 0
    for j in range(len(shares)):
        xj, yj = shares[j][0],
shares[j][1]
        prod = Decimal(1)
        for i in range(len(shares)):
            xi = shares[i][0]
            if i != j: prod *=
Decimal(Decimal(xi)/(xi-xj))
        prod *= yj
        sums += Decimal(prod)
    return int(round(Decimal(sums),0))
```

Ilustrasi kode utama atau *driver* yang mensimulasikan langkah 1 sampai 7 dijelaskan pada potongan kode berikut.

```
if __name__ == '__main__':

    # kunci publik otoritas
    keys = [1, 2, 3]

    # pilihan suara setiap pemilih
    votes = [1, 1, -1, -1, 1, 1]

    # nilai total Ax
    A_total = [[key, 0] for key in keys]

    for vote in votes:
        shares = encode_vote(keys, vote)
        for i in range(len(A_total)):
            A_total[i][1] += shares[i][1]

    print(A_total)
    print(reconstruct(A_total))
```

Sebagai contoh, bila kode di atas dijalankan, akan menghasilkan keluaran sebagai berikut. Perhatikan bahwa nilai A_x (sub-elemen kedua) pada setiap elemen dari `A_total` akan berubah setiap eksekusi program *driver* dikarenakan fungsi polinomial yang digunakan berkoefisien acak.

```
[[1, 595468], [2, 1748336], [3, 3458606]]
2
```

IV. Analisis Arsitektur

A. Analisis Kompleksitas

Menggunakan kakas *time* dengan bahasa pemrograman Python 3.7 pada lingkungan eksperimen komputer dengan prosesor Intel® Core™ i5-9600K, penulis mengujikan kasus penambahan jumlah pemilih dan jumlah otoritas dengan memanfaatkan kode *driver* yang disediakan pada Bab III. Diperoleh hasil sebagai berikut, beserta analisisnya.

Jumlah otoritas	Jumlah pemilih	*Runtime* (detik)
10	10	0.0009987354278
10	100	0.0059995651245
10	1.000	0.0510005950927
10	10.000	0.4830281734466
10	100.000	4.8389997482299
10	1.000.000	48.693008661270

Tabel 2. Perbandingan *runtime* dengan jumlah pemilih variabel

Perhatikan bahwa secara kasat mata, jumlah pemilih berbanding lurus dengan *runtime* yang menandakan kompleksitas algoritme terlihat O(n). Namun, hal ini sebenarnya dipengaruhi oleh *bottleneck* pada sistem *encode_vote* pada kode *driver* yang bergerak secara serial. Dalam kasus nyata, proses enkode dapat dijalankan oleh pemilih pada perangkatnya masing-masing secara paralel dan hanya perlu mengirimkan nilai polinomial untuk setiap otoritas saja. Selain itu, proses penjumlahan ke dalam *A_total* juga dijalankan secara serial, sehingga cukup memakan waktu.

Pada dasarnya, apabila diasumsikan seluruh pekerjaan tersebut (proses enkode dan penjumlahan) dilakukan secara paralel, maka *runtime* yang diterima tidak akan dipengaruhi oleh jumlah pemilih (O(1)). Dengan asumsi demikian, setelah uji coba hanya menggunakan nilai *A_total* yang sudah memiliki nilai, diperoleh *runtime* rata-rata 0.001 detik.

Jumlah otoritas	Jumlah pemilih	*Runtime* (detik)
10	10	0.0010452270507
100	10	0.0109996795654
1.000	10	7.0789990425109
10.000	10	lebih dari 5 menit, diberhentikan

Tabel 3. Perbandingan *runtime* dengan jumlah otoritas variabel

Pada kasus jumlah otoritas variabel, *runtime* yang dijalankan tampak kuadratik atau O(n^2), sesuai dengan algoritme yang digunakan yaitu interpolasi polinomial Lagrange. Hal ini menandakan bahwa pada jumlah otoritas 10.000, besar kemungkinan untuk *runtime* dapat mencapai 20 jam. Selain itu, sama seperti kasus sebelumnya, permasalahan pada kasus ini dapat diselesaikan dengan implementasi interpolasi Lagrange secara paralel. Namun, tidak seperti permasalahan sebelumnya, mengubah interpolasi Lagrange secara sederhananya hanya dapat memparalelkan bagian iterasi terluar setiap bagian *share* saja dikarenakan untuk iterasi di dalamnya memiliki *shared variable* yaitu *prod*. Sehingga, secara sederhana hanya dapat dimodelkan menjadi O(n^2/m) dengan asumsi terdapat m komputer yang akan menjalankannya secara paralel. Untuk modifikasi paralelisasi interpolasi Lagrange lebih lanjutnya tidak dibahas pada makalah ini.

Walaupun begitu, perlu dipertimbangkan bahwa pihak otoritas berjumlah ≤ 1.000 pada dasarnya sudah sangat cukup untuk menjadi penghitung suara dalam sebuah pengambilan suara mayoritas. Protokol ini tetap dapat bekerja sekecil-kecilnya apabila tidak seluruh k otoritas melakukan korupsi, yang akan dijelaskan pada subbab selanjutnya.

B. Analisis Kualitas

Dari kode *driver* yang ditampilkan pada Bab III, pada subbab ini akan diujikan apabila salah satu nilai *A_total* dilakukan perubahan yaitu korupsi oleh otoritas. Perubahan ditampilkan dengan latar belakang warna merah.

```
if __name__ == '__main__':

    # kunci publik otoritas
    keys = [1, 2, 3]

    # pilihan suara setiap pemilih
    votes = [1, 1, -1, -1, 1, 1]

    # nilai total Ax
    A_total = [[key, 0] for key in keys]

    for vote in votes:
        shares = encode_vote(keys, vote)
        for i in range(len(A_total)):
            A_total[i][1] += shares[i][1]

    A_total[0][1] -= 1

    print(A_total)
    print(reconstruct(A_total))
```

Apabila kode di atas dijalankan, dihasilkan keluaran berikut.

```
[[1, 477248], [2, 1488922], [3, 3035021]]
-1
```

Sedangkan apabila indeks yang diubah adalah indeks kedua, yaitu `A_total[1][1] -= 1`, dihasilkan keluaran berikut.

```
[[1, 547225], [2, 1684381], [3, 3411473]]
5
```

Dalam hal ini, ditemukan bahwa perubahan nilai A_x untuk setiap otoritas tidak memiliki perubahan yang dapat digeneralisasi untuk seluruh otoritas. Pada otoritas dengan indeks 0, pengurangan nilai A_x sebesar 1 menurunkan nilai akhir sebesar 3, namun pada otoritas dengan indeks 1 malah menaikkan nilai akhir. Hal ini membuktikan bahwa metode protokol suara mayoritas berbasis pembagian rahasia Shamir tidak dapat dipalsukan dikarenakan mengubah nilai menjadi nilai acak tak tentu tidak akan menguntungkan pihak manapun.

Dilihat dari aspek keamanan informasi pada konteks kriptografi, protokol ini dapat dinilai secara kualitas.

1. *Confidentiality* (kerahasiaan): dalam aspek kerahasiaan, pada dasarnya selama tidak semua otoritas melakukan tindakan korupsi atau curang, atau dengan kata lain selama masih ada satu otoritas yang tidak berlaku curang, protokol ini tetap menjaga kerahasiaan pemilihnya dikarenakan nilai pilihan suara dari pemilih hanya dapat didekode dengan menggunakan seluruh data A_x dari seluruh otoritas.
2. *Data integrity* (kebenaran): dilihat dari aspek kebenarannya, protokol ini dapat menjamin bahwa nilai total hasil dekode $\mathbf{\Sigma} A_x$ akan selalu sama dengan nilai total sesungguhnya. Hal ini ditunjukkan oleh sifat homomorfik dari pembagian rahasia Shamir yang dibuktikan lebih lanjut dengan sifat fungsi polinomial hanya terdapat satu yang berderajat k - 1 (dengan nilai konstanta adalah *sum*) jika terdapat k titik (otoritas). Namun, terdapat kelemahan fatal penggunaan protokol ini yaitu bahwa otoritas tidak dapat dengan yakin mengerti apakah pilihan (berbentuk p_n) yang diberikan pemilih legal, misalnya jika didekode hanya bernilai -1 sampai +1, atau tidak valid seperti +999. Mengenai hal ini dapat digunakan sebagai bahan pertimbangan dalam penggunaannya maupun untuk bahan penelitian selanjutnya.
3. *Authentication* (otentikasi atau identifikasi aktor): dilihat dari aspek identifikasi aktor, protokol ini dapat menyediakan verifikasi dengan setiap otoritas menyimpan identifikasi pemilih yang masing-masing dilabeli sesuai nilai yang diberikan pada langkah 4-5 Bab III, namun cara ini akan memunculkan isu baru yaitu bagaimana otoritas akan melakukan penyimpanan yang aman (agar tidak dapat diakses oleh pihak yang korupsi). Sistem penyimpanan aman itu tidak menjadi luaran dalam makalah ini.
4. *Non-repudiation* (tidak dapat disangkal): dilihat dari sifat tidak dapat disangkalnya, analisis kualitas pada aspek ini memiliki pembuktian yang sama dengan no (2). Dalam dunia yang sempurna, secara logika tidak akan ada otoritas yang akan mengubah nilai A_x miliknya karena tidak akan menguntungkan pihak manapun akibat tidak dimengertinya efek dari perubahan tersebut. Namun, tetap saja, ada kemungkinan bahwa seorang otoritas maupun pemilih dapat menyangkal dengan asumsinya sendiri bahwa terdapat otoritas yang berbuat korupsi, sehingga hasil dari pemilihan ini tidak valid. Protokol ini tidak memiliki cara khusus untuk menghadapi permasalahan tersebut, sehingga menjadi salah satu kekurangan.

Dilihat dari prinsip elektoral Indonesia, protokol ini juga dapat dinilai secara kualitas.

1. *Langsung*: pemilih tetap dapat memberikan suaranya sendiri tanpa ada penengah, dikarenakan pada dasarnya yang harus memberikan nilai p_n kepada masing-masing otoritas adalah pemilih itu sendiri.
2. *Umum*: tidak terpengaruh oleh penggunaan protokol ini, melainkan tergantung pada kebijakan pemerintah.
3. *Bebas*: tidak terpengaruh oleh penggunaan protokol ini, melainkan tergantung pada kondisi dari pemilih itu sendiri.
4. *Rahasia*: dengan asumsi tidak seluruh k otoritas adalah korupsi, maka pilihan pemilih akan selamanya rahasia.
5. *Jujur*: aspek ini ditekankan lebih dalam dengan menggunakan protokol pembagian rahasia Shamir, dikarenakan perubahan pada sisi otoritas memiliki hasil yang tidak dapat diduga sehingga mencegah terjadinya perubahan.
6. *Adil*: tidak terpengaruh oleh penggunaan protokol ini, melainkan tergantung pada kebijakan pemerintah.

V. Kesimpulan

Protokol suara mayoritas terdesentralisasi berbasis pembagian rahasia Shamir dapat menjadi salah satu alternatif solusi untuk menghilangkan kecurangan dari pihak otoritas penghitung suara dikarenakan sifat homomorfik yang dihasilkan dari karakteristik fungsi polinomial. Namun, protokol ini memiliki masalah utama yaitu bahwa kecurangan dapat muncul dari pihak pemilih apabila pemilih menghitung nilai polinomial dengan parameter x yang tidak valid, sehingga perlu dijadikan bahan pertimbangan untuk penggunaan maupun penelitian selanjutnya.

VI. Pengakuan

Penulis memberikan rasa terima kasih terbesar kepada Tuhan Yang Maha Esa karena dengan kehendaknya, penulis diberikan kesehatan dan kemampuan untuk dapat menyelesaikan makalah IF4020 Kriptografi secara tepat waktu. Penulis juga berterima kasih kepada orang tua penulis yang sudah mendidik dan mengajari ilmu-ilmu bermanfaat dari lahir sampai saat ini dan tidak ada hentinya.

Penulis juga ingin memberikan rasa terima kasih kepada Dr. Ir. Rinaldi Munir, M. T. sebagai dosen pengajar mata kuliah IF4020 Kriptografi karena telah memberikan ilmu-ilmu bermanfaat mengenai kriptografi tradisional dan modern. Penulis juga ingin memberi terima kasih kepada komunitas

Stack Overflow maupun kaum terpelajar daring karena dapat memberikan informasi penting mengenai hal-hal yang dipertimbangkan sebagai bahan analisis untuk menjadi isi dan isu dalam pembuatan makalah ini.

Terakhir, penulis juga berterima kasih kepada rekan-rekan Teknik Informatika 2017 yang selalu bersedia untuk saling membantu dan memberikan motivasi untuk tetap memiliki semangat berusaha, khususnya teman-teman dari K3 pada grup “OTW Seminar TA Gan”: Juniardi, Jofiandy, Irfan, Hamzah, Faiz, Winston, Hanif, dan Abda. Atas perhatian dan kepeduliannya selama ini, penulis mengucapkan terima kasih banyak.

Referensi

[1] Bellare, Mihir; Rogaway, Phillip (21 September 2005). *"Introduction". Introduction to Modern Cryptography*. p. 10.

[2] Menezes, A.J.; van Oorschot, P.C.; Vanstone, S.A. (1997). *Handbook of Applied Cryptography*. ISBN 978-0-8493-8523-0.

[3] Nita Hidayati (2019). "*Sejarah Singkat Perjalanan Pemilihan Umum di Indonesia*" (dalam bahasa Indonesia). Beritabaik.id.

[4] Rivest, Ronald L. (1990). "*Cryptography*". Pada J. Van Leeuwen (ed.). *Handbook of Theoretical Computer Science*. 1. Elsevier.

[5] Sheppard, William Fleetwood (1911). "*Interpolation*" . Pada Chisholm, Hugh (ed.). *Encyclopædia Britannica*. 14 (11th ed.). Cambridge University Press. pp. 706–710.

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, 18 Desember 2020

Asif Hummam Rais 13517099