Mekanisme Persetujuan Transaksi Aman dalam Jaringan Terdistribusi melalui Multi-Tanda Tangan dan Skema Pembagian Rahasia (DMSSAS)

Nicholas Liem - 13521135
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): nicholasliem01@gmail.com

***Abstract*—Jaringan terdistribusi sangat penting dalam aspek kehidupan kita. Sistem atau jaringan ini dipakai hampir diseluruh aspek seperti sistem keuangan, sistem kesehatan, sistem pertahanan, dan sebagainya. Persetujuan transaksi harus ditangani dengan tingkat keamanan yang tinggi diberbagai industri, termasuk pemerintahan dan perusahaan swasta, untuk mencegah penipuan, akses ilegal, dan kebocoran data. Sistem yang kuat, andal, dan aman dibutuhkan untuk menjaga integritas dan konfidentialitas data sensitif. Oleh sebab itu, diperlukan suatu mekanisme persetujuan transaksi yang aman untuk memastikan kepercayaan dan keandalan dalam aplikasi-aplikasi kritis. Penilitian ini berfokus pada peningkatan keamanan mekanisme persetujuan transaksi dalam jaringan terdistribusi menggunakan teknik multi-tanda tangan dan skema pembagian rahasia. Metode ini memberikan tingkat keamanan yang lebih tinggi dengan memastikan bahwa tidak ada titik kegagalan tunggal yang dapat mengkompromikan seluruh sistem. Tujuan dari penilitian ini adalah untuk mengembangkan sistem persetujuan transaksi yang aman dan efisien serta cocok untuk penggunaan di sektor swasta dan pemerintahan.**

Keywords—keamanan jaringan; tanda tangan digital; multi-tanda tangan; skema pembagian rahasia; persetujuan transaksi; jaringan terdistribusi; otentikasi node; integritas data

I. INTRODUKSI

Saat ini, jaringan terdistribusi sangat penting dalam berbagai aspek kehidupan kita. Aplikasi dari jaringan ini mencakup manajemen *supply chain*, jaringan komunikasi, layanan keuangan, sistem kesehatan, dan lain sebagainya. Persetujuan transaksi harus ditangani dengan tingkat keamanan yang tinggi diberbagai industri misalnya seperti pemerintahan dan perusahaan privat. Kebutuhan ini mencakup pencegahan penipuan, akses ilegal, kebocoran data, dan lain sebagainya. Jika tidak ditangani dengan benar maka setiap hal tersebut akan menjadi ancaman terhadap keamanan nasional bahkan akan menyebabkan kerugian yang sangat besar.

Untuk menjaga integritas dan konfidentialitas dari data yang sensitif, perusahaan maupun pemerintah harus memiliki sistem yang kuat, andal, dan aman dalam melaksanakan transaksinya. Prosedur-prosedur seperti pemrosesan dokumen, *e-voting*, dan transaksi-transaksi yang melibatkan kebutuhan publik adalah contoh-contoh hal yang perlu dijaga oleh setiap entitas atau penyedia layanan.

Oleh sebab itu, mekanisme persetujuan transaksi yang aman sangat penting untuk memastikan dan mempertahankan kepercayaan dan keandalan dalam aplikasi-aplikasi kritis ini. Lalu, selain menjamin keamanan dari aplikasi, kebutuhan dari sistem yang bersifat publik atau kritikal biasanya perlu dilakukan delegasi kekuasaan atau desentralisasi kekuasaan sehingga suatu entitas tidak bisa memegang kuasa penuh atas suatu sumber daya tetapi kekuasaan dipegang oleh beberapa partai atau entitas.

Dalam mengambangkan sistem yang membutuhkan tingkat keamanan yang tinggi, tentunya banyak sekali tantangan dan limitasi yang berasal bukan hanya dari segi teknikal tetapi juga segi sosial. Kita tidak bisa mengabaikan hukum yang perlu dipertimbangkan, serta kebutuhan teknis yang terlibat. Beberapa tantangan teknis yang dimaksud adalah seperti memastikan enkripsi yang kuat, mempertahankan skalabilitas sistem, serta perlindungan yang kuat terhadap serangan siber. Selain itu, beberapa undang-undang contohnya di Indonesia, memliki regulasi perlindungan data dan undang-undang transaksi elektronik yang mungkin perlu diperhatikan untuk memastikan kepatuhan dan integritas hukum terhadap aplikasi yang akan dikembangkan. Menyeimbangkan persyaratan teknis dan hukum ini sangat penting untuk keberhasilan implementasi sistem yang aman.

Dalam konteks ini, penelitian penulisan berfokus pada peningkatan keamanan mekanisme persetujuan transaksi dalam jaringan terdistribusi dengan menggunakan teknik multi-tanda tangan dan skema pembagian rahasia. Metode-metode ini memberikan tingkat keamanan yang lebih tinggi dengan memastikan bahwa tidak ada titik kegagalan tunggal yang dapat mengkompromikan seluruh sistem. Penulis bertujuan untuk mengembangkan mekanisme persetujuan transaksi yang aman dan efisien supaya cocok untuk penggunaan di sektor swasta dan pemerintah.

II. Pernyataan Masalah

A. Keamanan Transaksi dalam Jaringan Terdistribusi

Saat menggunakan jaringan terdistribusi, tantangan utama dalam menjalankan transaksi adalah menjaga keamanan dan integritas transaksi. Oleh karena itu, node dalam cluster harus diotentikasi agar otorisasi transaksi dapat dilakukan.

B. Otentikasi Node

Otentikasi node diperlukan untuk memastikan bahwa setiap node yang berpartisipasi dalam menyetujui transaksi adalah sah dan untuk menghindari situasi di mana node yang tidak sah dapat mengotorisasi transaksi.

C. Pengelolaan Kunci Rahasia

Kunci rahasia harus dikelola sedemikian rupa sehingga tidak ada node yang memiliki kendali penuh atas kunci tersebut. Skema pembagian rahasia dapat digunakan untuk mendistribusikan kunci rahasia ke beberapa node sehingga kunci tersebut hanya dapat direkonstruksi jika jumlah node yang diautentikasi mencukupi.

D. Pencegahan Serangan dan Kompromi Sistem

Sistem harus mampu mencegah serangan yang dapat mempengaruhi keamanan transaksi dan integritas data dengan menciptakan mekanisme transaksi yang melindungi sistem sistem dari node yang terkompromi atau serangan eksternal.

E. Validitas dan Keamanan Transaksi

Setiap transaksi yang disetujui harus diautentikasi secara akurat dan aman, artinya sistem harus memastikan bahwa persetujuan transaksi dilakukan melalui mekanisme yang terpercaya dan dapat diandalkan.

F. Kebutuhan akan Mekanisme Gabungan

Mekanisme kombinasi multi-tanda tangan diperlukan untuk mengautentikasi node dan skema pembagian rahasia untuk melindungi kunci rahasia. Tujuan dari kedua elemen ini adalah untuk memastikan persetujuan transaksi yang aman dan andal dalam jaringan terdistribusi.

Pada intinya, dalam lingkungan jaringan terdistribusi, keamanan dan integritas transaksi merupakan tantangan utama. Proses persetujuan transaksi harus memastikan bahwa hanya node yang diautentikasi yang dapat mengotorisasi transaksi dan tidak ada node yang memiliki kendali penuh atas kunci privat. Hal ini penting untuk mencegah serangan yang dapat membahayakan sistem dan memastikan bahwa setiap transaksi yang disetujui adalah valid dan aman. Oleh karena itu, diperlukan suatu mekanisme yang dapat menggabungkan multi-signature untuk otentikasi node dan berbagi rahasia untuk melindungi kunci rahasia, untuk memastikan persetujuan transaksi dalam jaringan terdistribusi yang aman dan dapat dipercaya.

III. Landasan Teori

A. Sistem Terdistribusi dan Asumsi-Asumsi Pengembangan Aplikasi Terdistribusi

Sistem terdistribusi adalah sekumpulan unit komputer yang terhubung melalui suatu koneksi membentuk suatu jaringan. Masing-masing dari unit komputer ini berkomunikasi dan bekerjasama dalam menyelesaikan suatu masalah komputasi. Karakteristik utama dari aplikasi terdistribusi adalah skalabilitas, konkurensi, paralelisasi, transparansi, keterbukaan, dan toleransi terhadap kesalahan. Setiap sistem terdistribusi dapat disebut sebagai sistem yang terdistribusi jika setiap komponen komputer pada sistem memiliki suatu hal yang dapat dibagi bersama. Dalam mengembangkan aplikasi yang terdistribusi kita harus memastikan beberapa asumsi yang kita pegang terlebih dahulu. Beberapa asumsi yang harus dipegang adalah sebagai berikut.

1. Jaringan tidak selalu andal
2. Jaringan tidak selalu aman
3. Jaringan tidak homogen
4. Topologi dapat berubah
5. Latensi tidak nol
6. Bandwidth tidak tak terbatas
7. Biaya transportasi data tidak nol
8. Terdapat banyak administrator

Dari semua asumsi tersebut, salah satu hal yang perlu diperhatikan adalah keamanan jaringan karena pada kenyataannya jaringan pada sistem terdistribusi cukup mudah untuk terkompromi, terutama jika aplikasi yang dibuat tidak dibangun dengan metrik keamanan yang baik.

B. Keamanan Jaringan pada Sistem Terdistribusi

Keamanan jaringan adalah aspek penting dalam sistem terdistribusi. Sistem terdistribusi digunakan secara luas dalam beberapa layanan produk untuk memastikan ketersediaan, keandalan, dan keamanan transaksi. Salah satu penerapan teknologi terdistribusi untuk bidang finansial adalah *Distributed Ledger Technology* (DLT). Teknologi ini menawarkan berbagai manfaat seperti transparansi, desentralisasi, dan integritas data, tetapi juga menghadapi tantangan keamanan yang cukup signifikan khususnya karena sistem ini terdesentralisasi jadi banyak peretas yang berusaha untuk melakukan peretasan terhadap teknologi ini. Untuk itu, kebutuhan akan kerangka keamanan dalam penjaminan transaksi secara terdistribusi supaya transaksi dalam dijalankan secara aman.

Ancaman keamanan yang paling umum pada sistem terdistribusi adalah sebagai berikut.

1. Kompromi Node

   Kompromisasi dari node adalah bentuk serangan di mana penyerang berhasil mendapatkan akses ke node dalam sistem dan menggunaka node tersebut untuk

mengacaukan transaksi atau menyebarkan informasi palsu.

2. Man-in-the-Middle Attack

   Penyerang menyusup ke dalam komunikasi antara dua node dan dapat mengubah ataui mencuri informasi yang dikirimkan. Biasanya mitigasi untuk penyelesaian masalah ini adalah menggunakan protokol TLS atau enkripsi *end-to-end*.

3. Distributed Denial of Service (DDOS)

   DDOS adalah cara penyerang untuk membanjiri sistem dengan lalu lintas palsu untuk membuat layanan tidak tersedia bagi pengguna yang sah. Salah satu caranya mitigasinya adalah menggunakan firewall atau dengan menggunakan arsitektur yang cukup *scalable* untuk menangani lonjakan lalu lintas.

4. Sybil Attack

   Penyerang membuat banyak identitas palsu untuk mendapatkan pengaruh yang tidak semestinya. Salah satu mitigasinyua adalah untuk menggunakan validasi identitas dan memastikan bahwa node baru diverifikasi sebelum diberi hak akses.

Dari ancaman-ancaman tersebut maka dapat disimpulkan beberapa titik-titik keamanan yang perlu dijaga.

1. Autentikasi dan Otorisasi

   Sistem perlu memastikan hanya pengguna dan perangkat yang sah yang dapat mengakses sistem melalui penggunaan mekanisme autentikasi yang kuat dan otorisasi berbasis peran (*role based authorization*).

2. Enkripsi

   Pengiriman dan penyimpanan data harus menggunakan enkripsi yang kuat supaya menjaga kerahasiaan dan integritas data. Enkripsi digunakan misalnya pada saat proses pertukaran kunci atau pemberian kunci antar node.

3. Firewall dan IDS

   Menggunakan firewall untuk mengontrol lalu lintas jaringan masuk dan keluar, serta sistem deteksi intrusi (IDS) digunakan untuk mengidentifikasi dan merespons ancaman potensial.

C. Tanda Tangan Digital

Tanda tangan digital adalah suatu bukti otentikasi terkait suatu identitas. Tanda tandan mempunyai karakteristik sebagai berikut.

1. Tanda tangan adalah bukti yang otentik
2. Tanda tangan tidak dapat dilupakan
3. Tanda tangan tidak dapat dipindah untuk digunakan ulang
4. Dokumen yang telah ditandatangani tidak dapat diubah
5. Tanda tangan tidak dapat disangkal

Beberapa persyaratan pada tanda tangan digital adalah sebagai berikut.

1. Tanda tangan harus berupa rangkaian bit yang bergantung pada pesan yang ditandatangani
2. Tanda tangan harus menggunakan informasi yang unik dari pengirim untuk mencegah pemalsuan dan penyangkalan
3. Membangkitkan tanda tangan digital harus relatif mudah dilakukan
4. Mengelani dan memverifikasi tanda tangan digital harus relatif mudah dilakukan
5. Secara komputasi hampir tidak mungkin memalsukan tanda tangan digital, baik dengan merekonstruksi pesan baru untuk tanda tangan digital yang sudah ada, atau merekonstruksi tanda tangan curang untuk pesan yang diberikan
6. Menyimpan salinan tanda tangan digital ke dalam storage harus mudah dilakukan secara praktek

Proses pada tanda tangan digital dibagi menjadi dua bagian, yakni proses *signing* dan proses *verification*.

Pada proses penandatanganan, ada dua cara yang dapat dilakukan, yakni dengan mengenkripsi pesan dan yang kedua adalah menggunakan kombinasi fungsi hash dan kriptografi kunci publik untuk pesan yang tidak perlu rahasia. Ada dua skema terkenal yang biasa digunakan untuk melakukan skema tanda tangan digital.

Alur utama dari skema tanda tangan digital adalah sebagai berikut. Pertama pesan akan dihash menggunakan fungsi hash kemudian akan dihasilkan *Message Digest* (MD) yang kemudian akan dienkripsi menggunakan kunci privat pengirim sehingga menghasilkan suatu signature. Signature ini akan ditambahkan pada pesan yang ingin dikirim ke penerima.

Penerima akan memecah pesan menjadi pesan dan signature-nya kemudian akan melakukan pengecekan dengan melakukan dekripsi signature menggunakan kunci publik dari pengirim. Jika hasil dari *Message Digest* dekripsi sama dengan *Message Digest* yang diterima penerima oleh pesan tersebut maka dapat dikatakan bahwa tanda tangan valid. Beberapa algoritma yang digunakan secara luas adalah RSA dan ElGamal Signature, terdapat beberapa algoritma standar juga seperti DSA (*Digital Signature Algorithm*).

Proses penandatanganan menggunakan DSA adalah sebagai berikut.

1. Proses penandatanganan (*signing*)

   Proses pembangkitan pasangan kunci:

   1. Pilih bilangan prima *p* dan *q* yang memenuhi persamaan berikut ini.

      $$(p-1) \bmod q = 0$$

2. Hitung $g = h^{(p-1)/q} \bmod p$ yang dalam hal ini juga $1 < h < p-1$ dan $h^{(p-1)/q} \bmod p > 1$

3. Tentukan kunci privat *x*, di mana $0 < x < q$

4. Hitung kunci publik $y = g^x \bmod p$

Pembangkitan tanda tangan adalah sebagai berikut

1. Hitung *message digest* pesan *m* dengan fungsi hash *SHA-1*, *H(m)*.
2. Tentukan bilangan acak *k*, $0 < k < q$.
3. Tanda tangan dari pesan *m* adalah bilangan r dan s, hitung *r* dan *s* sebagai berikut
   - $r = (g^k \bmod p) \bmod q$
   - $s = (k^{-1}(H(m) + x \cdot r)) \bmod q$
4. Kirim pesan *m* beserta tanda tangan (*r*, *s*)

2. Proses verifikasi (*verification*)

1. Hitung *message digest* pesan *m* dengan fungsi hash *SHA-1*, *H(m)*.
2. *V*erifikasi tanda tangan *r* dan s sebagai berikut:
   - $w = s^{-1} \bmod q$
   - $u_1 = (H(m) \cdot w) \bmod q$
   - $u_2 = (r \cdot w) \bmod q$
   - $v = ((g^{(u_1)} \cdot y^{(u_2)}) \bmod p) \bmod q$
3. Jika *v* = *r* maka tanda tangan sah

D. Skema Pembagian Rahasia Shamir

1. Skema Ambang

   Pada skema ambang, misalkan ada dua buah bilangan positif *t* dan *n* sedemikian rupa $t \leq n$. Skema ini membagi secret *S* kepada *n* partisipan sehingga sembarang himpunan bagian yang terdiri dari *t* partisipan dapat merekonstruksi *S*, tetapi jika kurang dari nilai *t* maka *S* tidak dapat direkonstruksi.

2. Skema Shamir

   Skema Shamir diawali dengan ide persoalan interpolasi misalkan untuk persamaan linear dibutuhkan dua buah titik untuk membentuk persamaannya, butuh tiga titik untuk membentuk persamaan kuadratik, dan seterusnya untuk membentuk polinomial derajat *n* maka dibutuhkan sebanyak *n + 1* titik.

   Dari titik sebanyak *n + 1* tersebut dan disulihkan ke dalam persamaan $y = p_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$ sehingga akan dibentuk sebanyak *n* buah sistem persamaan lanjar. Solusi persamaan ini dapat dicari menggunakan metode eliminasi Gaussian.

3. Skema (*t*, *n*)

   Pilih bilangan prima *p* yang lebih besari dari semua kemungkinan nilai rahasia *S* dan juga lebih besar dair jumlah *n* partisipan. Kemudian pilih *t – 1* buah bilangan bulat acak dalam modulus *p* dan nyatakan dalam polinomial seperti berikut

   $f(x) = S + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{(t-1)}$

   sehingga nilai $f(0) = S \pmod p$.

4. Untuk *n* partisipan, pilih sebanyak *n* buah bilangan berbeda dan setiap orang memperoleh pembagian nilai (x_i, y_i) di mana nilai y ditentukan dari $y_i = f(x_i) \bmod p$

IV. RANCANGAN PENYELESAIAN MASALAH

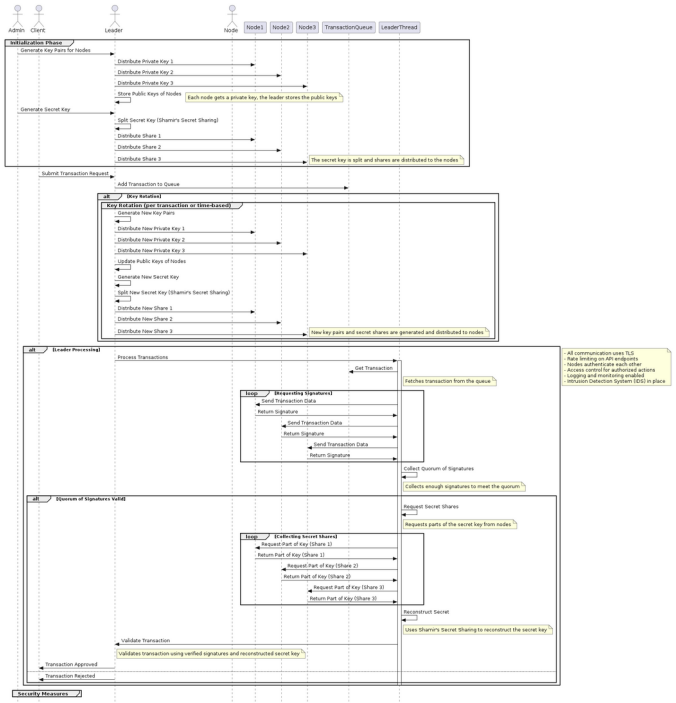

Fig. 1. Diagram sekuens untuk DMSSAS (Distributed Multi-Signature and Secret Sharing Authentication Scheme)

DMSSAS (Distributed Multi-Signature and Secret Sharing Authentication Scheme) adalah sebuah skema bertahap yang dirancang untuk memastikan dua hal, yakni penggunaan *multi-signature* untuk melakukan otentikasi dari setiap node yang tergabung dalam suatu *cluster* dan *secret sharing authentication* memastikan bahwa tidak satu pun node dalam

cluster yang memiliki kuasa penuh atas suatu request atau resource. Sebuah request atau resource harus memenuhi jumlah kuorum dari node-node yang terautentikasi. Dengan demikian, keamanan dijaga melalui dua lapisan keamanan.

A. Aktor

Aktor pada DMSSAS terdiri dari 4 buah entitas, yakni Admin, Client, Leader, dan Node. Masing-masing aktor memiliki peran yang berbeda pada skema ini.

Admin adalah aktor yang berfungsi untuk membangkitkan kunci pada tahap inisiasi. Kunci yang dibangkitkan adalah kunci pasangan untuk *digital signature* dan kunci rahasia untuk skema pembagian rahasia Shamir.

Client adalah aktor yang melakukan inisiasi *request* transaksi kepada Leader untuk diproses. Client juga nanti akan menjadi entitas yang menerima pesan atau konfirmasi apakah *request* yang dikirim ke suatu cluster berhasil dilakukan atau tidak.

Leader adalah aktor yang tugasnya adalah untuk membagi atau mendistribusikan hasil pembangkitan kunci dari admin untuk node-node yang ada pada suatu klaster (semuanya berbeda), serta menyimpan hasil dari pembagian kunci untuk signature. Selain itu, Leader juga melakukan skema pembagian kunci untuk skema pembagian rahasia Shamir dan membagikan setiap hasil pecahan kunci ke node-node yang ada pada clusternya. Selain itu, tugas Leader adalah menerima *request* transaksi dari klien dan kemudian memasukkan *request* tersebut dalam sebuah *queue* yang disebut TransactionQueue untuk kemudian diproses oleh thread dari Leader atau LeaderThread.

Leader juga berfungsi untuk melakukan *key rotation* di mana untuk menjaga keamanan untuk setiap *request* yang datang, setiap kali sebuah *request* baru akan diproses atau untuk lebih efisien misalnya setiap beberapa waktu tertentu misalnya 24 jam, maka kunci rahasia untuk *signature* dan kunci rahasia untuk skema pembagian Shamir akan dihasilkan ulang. Beberapa pendekatan dapat dilakukan untuk mengeksekusi hal ini, salah satunya dengan menggunakan sebuah *cron-server*.

Leader juga berfungsi untuk memastikan apakah *signature* serta kunci rahasia Shamir yang dibentuk benar-benar valid berdasarkan *response* dari setiap node. Hal ini dilakukan karena Leader diasumsikan sebagai *source of truth*. Selain itu, Leader juga bertugas untuk memberitahu Client apakah transaksi itu valid dan dapat dilaksanakan berdasarkan hasil dari verifikasi *signature* dan kunci rahasia Shamir yang tepat.

Aktor terakhir adalah Node yang berfungsi untuk menjaga dua tipe kunci rahasia yang diberikan oleh Leader. Kunci pertama yang harus dijaga adalah kunci untuk *signature* di mana kunci ini akan menandakan autentitas dari node tersebut pada cluster yang diberikan. Kunci kedua adalah kunci dari skema pembagian rahasia Shamir yang akan digunakan sebagai bagian dari kebutuhan kuorum saat rekonstruksi kunci rahasia pada Leader.

B. Tahapan Skema

Berikut adalah *pseudocode* untuk implementasi program

```
PROGRAM

Initialization Phase
Admin generates secret key
Leader splits secret key using Shamir's Secret Sharing
Leader distributes shares:
    Send Share 1 to Node 1
    Send Share 2 to Node 2
    Send Share 3 to Node 3
    Send Share n to Node n

Transaction Processing
Client submits transaction request to Leader
Leader adds transaction to queue

Key Rotation (per transaction or time-based)
Leader generates new secret key
Leader splits new secret key using Shamir's Secret Sharing
Leader distributes new shares:
    Send New Share 1 to Node 1
    Send New Share 2 to Node 2
    Send New Share 3 to Node 3
    Send New Share n to Node n

Leader Processing
Leader starts thread to process transactions
Leader thread fetches transaction from queue
Requesting signatures:
    Request signature from Node 1
    Receive signature from Node 1
    Request signature from Node 2
    Receive signature from Node 2
    Request signature from Node 3
    Receive signature from Node 3
    Request signature from Node n
    Receive signature from Node n

Leader thread collects quorum of signatures
If quorum of signatures is valid:
    Leader thread reconstructs secret using Shamir's Secret
Sharing
    Leader validates transaction using reconstructed secret
    Leader approves transaction and notifies Client
Else:
    Leader rejects transaction and notifies Client
```

Tahapan dalam skema DMSSAS terdiri dari tiga tahapan utama. Tahapan pertama adalah tahap inisialisasi, tahap kedua adalah tahap rotasi kunci, dan tahapan ketiga adalah tahap pemrosesan *request*. Masing-masing tahap memiliki tujuannya masing-masing yang akan dijelaskan selanjutnya.

Tahap pertama adalah tahap inisalisasi atau *Initialization Phase*. Pada tahap ini, Admin membangkitkan sejumlah

pasangan kunci sebanyak *n* buah di mana *n* adalah jumlah node pada cluster (dihitung tidak termasuk Leader). Selanjutnya, pada tahap ini juga Leader akan mendistribusikan setiap kunci rahasia tersebut ke node-node dalam cluster tersebut. Setiap node memiliki kunci rahasianya sendiri yang unik. Selanjutnya, Admin juga akan membangkitkan kunci lagi untuk skema pembagian rahasia Shamir di mana Leader akan membagi kunci rahasia tersebut dan mendistribusikannya ke node-node lain yang ada di cluster tersebut.

Ketika ada suatu *request* transaksi yang masuk dari Client ke Leader, maka Leader akan memasukkan *request* tersebut dalam sebuah *Queue* supaya transaksi dalam dilakukan secara sekuensial. Tergantung pada implementasinya, Leader dalam melakukan rotasi kunci pada setiap *request* baru yang masuk atau dalam beberapa waktu tertentu misalnya setiap 30 menit atau jangka waktu yang ditentukan. Hal ini ditujukan supaya kunci baru akan dihasilkan terus sehingga akan sangat sulit untuk membobol kunci yang dipakai dan menghindari *re-use* key secara berlebihan. Tahap rotasi kunci (kedua) identikal dengan tahap inisialisasi tetapi bedanya dilakukan pada saat cluster sudah berjalan atau sedang berjalan (bukan inisialisasi).

Tahap ketiga adalah tahap pemrosesan *request*. Pada tahap ini, *request* yang ada di TransactionQueue akan diambil oleh LeaderThread kemudian diproses. Setelah transaksi didapatkan dari Queue, maka langkah selanjutnya adalah LeaderThread akan meminta *signature* dari masing-masing node dan memverifikasi apakah *signature* tersebut valid dan memenuhi jumlah kuorum yang ditetapkan. Jika kuorum dari *signature* sudah valid, maka selanjutnya Leader akan meminta kepada setiap node yang telah divalidasi sebelumnya untuk memberikan bagian dari kunci rahasianya supaya kunci rahasia bisa direkonstruksi ulang. Kemudian, setelah kedua tahap verifikasi tersebut berhasil dilakukan, maka transaksi bisa sepenuhnya diverifikasi atau ditindaklanjuti.

C. Catatan Tambahan

Beberapa hal yang perlu diperhatikan dan baik dijadikan asumsi adalam interaksi antara setiap Aktor adalah semua komunikasi dilakukan di bawah TLS, selain itu juga lakukan *rate-limitting* pada setiap API *endpoints*, lalu setiap node melakukan otentikasi terhadap sesamanya, juga harus ada akses kontrol pada *authorized actions* misalnya pemrosesan transaksi, dan perlu juga ada *logging* dan *monitoring*, dan yang terakhir untuk meningkatkan keamanan pada sistem diperlukan juga IDS atau Intrusion Detection System.

V. Hasil dan Analisis

A. Hasil Benchmarking

Perbandingan hasil atau *benchmarking* dilakukan menggunakan Laptop ASUS X415 dengan RAM 12 GB dan Processor Intel i3 11th Gen. *Benchmarking* dilakukan pada tiga tahap utama dalam program, yakni proses inisialisasi (pembagian kunci untuk *digital signature*, kemudian proses pembagian rahasia dengan skema Shamir, dan yang terakhir adalah proses pemrosesan transaksinya sendiri). Program dijalankan dengan menggunakan Python v3.10 dengan *framework* Flask. Server dijalankan menggunakan arsitektur satu buah Leader dan tiga buah Node (anggota).

TABLE I. Waktu Penyelesaian Pekerjaan (End-To-End & Transaction success)

Task	***Time Taken (avg)***
Pembagian Kunci (*Signature*)	7 ms
Pembagian Rahasia *(Secret Share*)	6 ms
Pemrosesan dan Pengecekan Transaksi	74 ms

B. Analisis

Dari segi efisiensi dan signifikasnsinya, pada *task* pembagian kunci dan pembagian rahasia, waktu yang dibutuhkan cukup rendah sehingga dapat dikatakan bahwa sistem efisien dalam melakukan *request* dan *receiving* kunci untuk tanda tangan digital. Hal ini sama dengan pembagian rahasia.

Hal yang menarik dari *benchmarking* tersebut adalah waktu yang digunakan untuk pemrosesan dan pengecekan transaksi. Task ini terdiri dari beberapa lojik yang kompleks dan bersifat sekuensial yakni Leader akan meminta *signature* dari seluruh Node terlebih dahulu kemudian jika memenuhi kuorum maka baru dilanjutkan dengan rekonstruksi rahasia dari Node yang masuk kuorum tersebut. Hal ini memakan waktu 10 kali lebih banyak dari pembagian kunci. Namun, jika dipertimbangkan dengan waktu aslinya, dapat dikatakan pemrosesan masih cukup cepat.

VI. Simpulan dan Saran

Penelitian ini menunjukkan bahwa penggunakan mekanisme multi-tanda tangan dan skema pembagian rahasia dalam jaringan terdistribusi dapat meningkatkan keamanan dan keandalan sistem dalam persetujuan transaksi. Berdasarkan hasil *benchmarking*, sistme ini cukup efisien dalam pembagian kunci dan pembagian rahasia, dengan waktu yang cukup rendah. Namun, pemrosesan dan pengecekan transaksi membutuhkan waktu yang lebih lama, disebabkan oleh kompleksitas logika dan langkah-langkah sekuensial yang dijalankan.

Penerapan skema DMSSAS (*Distributed Multi-Signature and Secret Sharing Authentication*) memastikan bahwa tidak ada satu node yang memiliki kuasa penuh atas kunci privat, yang penting untuk mencegah serangan yang ditujukan kepada sistem. Penggunaan skema ini dapat dilakukan dalam lingkungan yang membutuhkan keamanan tinggi, seperti sistem keuangan dan pemerintahan sehingga dapat meningkatkan integritas dan konfidentialitas data sensitif.

Beberapa saran untuk meningkatkan performa dari prototipe skema DMSSAS adalah sebagai berikut.

1. Memanfaatkan parallelisme

   Salah satu cara untuk mengurasi waktu pemrosesan dan pengecekan transaksi adalah dengan memanfaatkan parallelisme. Pemrosesan permintaan tanda tangan dari beberapa node bisa dilakukan secara

paralel sehinga waktu total yang dibutuhkan untuk mengumpulkan tanda tangan dapat dikurangi secara signifikan. Selain itu juga saat meminta tanda tangan, hal yang bisa dijalankan secara paralel juga pengumpulan *secret* yang nanti kemudian digabungkan lojiknya atau di*filter* berdasarkan permintaan tanda tangan sebelumnya.

2. Penyesuaian kuorum

   Evaluasi dan penyesuaian jumlah kuorum yang diperlukan untuk persetujuan transaksi dapat memberikan keseimbangan antara keamanan dan kinerja. Menentukan jumlah kuorum yang optimal bisa membantu mengurangi waktu pemrosesan dan tetap menjaga tingkat keamanan yang memadai. Salah satu caranya dengan menghitung jumlahnya dengan rumus n/2 + 1 di mana n adalah jumlah node pada suatu klaster.

3. Penggunaan *asynchronous I/O*

   Menggunakan *Async I/O* untuk menangani komunikasi jaringan dapat meningkatkan kinerja sistem dengan mengurangi waktu tunggu yang tidak perlu. Misalnya ketika pembagian kunci, penerimaan kunci, dan lain sebagainya.

PRANALA KODE PROGRAM IMPLEMENTASI

https://github.com/NicholasLiem/DMSSAS

UCAPAN TERIMA KASIH

Puji syukur kepada Tuhan Yang Maha Esa atas berkat dan penyertaan-Nya sehingga penulis bisa menyelesaikan makalah yang berjudul *Mekanisme Persetujuan Transaksi Aman dalam Jaringan Terdistribusi melalui Multi-Tanda Tangan dan Skema Pembagian Rahasia (DMSSAS)* yang diselesaikan dengan baik. Penulis ingin mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T selaku dosen mata kuliah IF4020 Kriptografi yang telah memberikan bimbingan, ilmu, dan pemahaman tentang berbagai materi yang digunakan dalam penulisan makalah ini. Penulis juga ingin mengucapkan terima kasih kepada para penulis sumber referensi yang digunakan pada makalah ini karena telah memberikan ilmunya juga.

DAFTAR PUSTAKA

[1] Munir, Rinaldi. “Tanda-tangan digital (digital signature)”. https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2023-2024/29-Tanda-tangan-digital-2024.pdf. Diakses pada tanggal 12 Juni 2024.

[2] Munir, Rinaldi. “Elgamal Signature Scheme dan Schnorr Signature Scheme”. https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2023-2024/30-Elgamal-Signature-dan-Schnorr-Signature-Scheme-2024.pdf . Diakses pada tanggal 12 Juni 2024.

[3] Munir, Rinaldi. “Skema pembagian data rahasia (Shamir secret sharing)”. https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2023-2024/35-Skema-Pembagian-Data-Rahasia-2024.pdf. Diakses pada tanggal 12 Juni 2024.

[4] Geeks For Geeks. “What is Multisignature-Wallets?”. https://www.geeksforgeeks.org/what-is-mulisignature-wallets/. Diakses tanggal 12 Juni 2024.

[5] Kistijantoro, Achmad Imam, et al. “Konsep Dasar Sistem Terdistribusi”. Diakses tanggal 12 Juni 2024.

[6] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612-613, Nov. 1979. [Online]. Available: https://web.mit.edu/6.857/OldStuff/Fall03/ref/Shamir-HowToShareASecret.pdf. Diakses tanggal 12 Juni 2024.

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

Nicholas Liem (13521135)