Analisis Penggunaan Hybrid Post-Quantum pada TLS 1.3 Untuk Pencegahan Serangan Harvest-Now-Decrypt-Later

Muhamad Rifki Virziadeili Harisman - 13522120
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13522120@stei.itb.ac.id [2]rifkivirziadeili@gmail.com

***Abstract*— Perkembangan komputer kuantum memunculkan kekhawatiran terhadap dunia kriptografi. Kekuatan dari algoritma kriptografi pada saat ini didasarkan pada kesulitan dalam perhitungan matematisnya. Hal ini membuat proses pemecahannya akan memakan waktu dan resource yang besar. Akan tetapi, komputer kuantum membawa kekuatan komputasi yang cukup besar untuk memecahkan banyak persoalan matematis, salah satunya adalah kriptografi. Salah satu penerapannya Adalah pada TLS. Versi terbaru dari TLS, 1.3 telah membekali keamanan yang cukup mumpuni. Namun, hal ini memiliki risiko ancaman "Harvest-Now-Decrypt-Later". Ancaman ini akan mengambil informasi sebanyak banyaknya pada masa kini, kemudian melakukan dekripsi di kemudian hari. Makalah ini akan mengevaluasi penggunaan skema pertukaran kunci Hybrid Post-Quantum yang mengombinasikan Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) dan Key Encapsulation Mechanism (KEM) Kyber pada TLS 1.3. Implementasi dilakukan dengan memodifikasi pustaka OpenSSL dan memanfaatkan liboqs. Evaluasi eksperimental menunjukkan bahwa skema hybrid bisa digunakan dengan baik pada komputer saat ini. Hal ini menunjukan, jika TLS pada masa sekarang menggunakan algoritma Hybrid Post Quantum, dapat mencegah terjadinya serangan Harvest-Now-Decrypt-Later.**

***Keywords*—TLS 1.3, Post-Quantum Cryptography, Hybrid Key Exchange, Kyber, Harvest-Now-Decrypt-Later**

I. INTRODUCTION

Keamanan komunikasi digital modern sangat bergantung pada protokol Transport Layer Security (TLS), yang digunakan secara luas untuk melindungi lalu lintas data pada layanan web, perbankan daring, sistem pemerintahan, layanan kesehatan, serta berbagai infrastruktur kritis lainnya. Sejak diperkenalkan pertama kali sebagai SSL, protokol ini terus mengalami evolusi untuk merespons berbagai insiden keamanan nyata, seperti serangan Man-in-the-Middle, kompromi otoritas sertifikat, serta kebocoran kunci privat server. Beberapa insiden besar, seperti kompromi sertifikat DigiNotar pada tahun 2011 dan berbagai serangan terhadap implementasi TLS lama, mendorong pengembangan TLS 1.3 sebagai versi yang lebih aman, ringkas, dan efisien.

TLS 1.3 dirancang dengan fokus utama pada peningkatan keamanan dan kinerja. Salah satu perubahan fundamental pada arsitektur TLS 1.3 adalah penghapusan algoritma kriptografi yang dianggap lemah serta penyederhanaan proses *handshake*. Pada TLS 1.3, pertukaran kunci sesi dilakukan secara eksklusif menggunakan mekanisme Elliptic Curve Diffie-Hellman Ephemeral (ECDHE), yang memberikan properti *forward secrecy* terhadap penyerang dengan kemampuan komputasi klasik. Selain itu, TLS 1.3 memperkenalkan *session resumption* berbasis Pre-Shared Key (PSK) untuk menurunkan latensi komunikasi pada koneksi berulang.

Meskipun demikian, seluruh jaminan keamanan TLS 1.3 secara implisit bergantung pada asumsi bahwa algoritma kriptografi kunci-publik yang digunakan, khususnya Elliptic Curve Cryptography (ECC), tidak dapat dipecahkan dalam waktu yang realistis. Asumsi ini mulai dipertanyakan seiring berkembangnya riset komputer kuantum. Algoritma Shor telah terbukti secara teoretis mampu menyelesaikan permasalahan logaritma diskret dan faktorisasi bilangan bulat dalam waktu polinomial pada komputer kuantum berskala besar. Dengan kata lain, algoritma kriptografi kunci-publik yang menjadi fondasi utama TLS 1.3 berpotensi menjadi tidak aman di masa depan.

Permasalahan ini menjadi semakin serius karena TLS tidak hanya melindungi komunikasi sesaat, tetapi juga digunakan untuk mengamankan data yang memiliki nilai kerahasiaan jangka panjang, seperti rekam medis, dokumen hukum, data finansial, dan arsip pemerintahan. Dalam konteks ini, muncul ancaman yang dikenal sebagai Harvest-Now-Decrypt-Later, yaitu skenario di mana penyerang merekam lalu lintas TLS saat ini tanpa harus memecahkannya secara langsung, kemudian menyimpan data tersebut untuk didekripsi di masa depan ketika komputer kuantum telah tersedia. Ancaman ini bersifat pasif, sulit dideteksi, dan tidak dapat dimitigasi hanya dengan *forward secrecy* berbasis kriptografi klasik.

Post-Quantum Cryptography dikembangkan untuk mengatasi ancaman tersebut dengan merancang algoritma kriptografi yang tahan terhadap serangan komputer kuantum. Namun, integrasi kriptografi pasca-kuantum ke dalam protokol TLS bukanlah permasalahan trivial. Algoritma pasca-kuantum umumnya memiliki ukuran kunci dan ciphertext yang lebih besar serta karakteristik kinerja yang berbeda dibandingkan algoritma klasik. Oleh karena itu, penerapan kriptografi pasca-kuantum secara langsung tanpa perubahan arsitektur TLS berpotensi menimbulkan masalah kompatibilitas, *overhead* kinerja, dan kompleksitas implementasi.

Salah satu pendekatan yang saat ini banyak dikaji sebagai solusi transisi adalah skema *hybrid post-quantum*, yaitu penggabungan algoritma kriptografi klasik dan pasca-kuantum dalam satu mekanisme pertukaran kunci. Pendekatan ini bertujuan untuk mempertahankan kompatibilitas dan keamanan terhadap penyerang klasik, sekaligus memberikan ketahanan terhadap ancaman kuantum di masa depan. Dalam konteks TLS 1.3, skema hybrid dapat diterapkan dengan mengombinasikan ECDHE dan Key Encapsulation Mechanism (KEM) pasca-kuantum, seperti Kyber.

Berdasarkan latar belakang tersebut, makalah ini bertujuan untuk mengevaluasi sejauh mana penerapan skema *hybrid post-quantum* pada TLS 1.3 mampu meningkatkan ketahanan kerahasiaan jangka panjang terhadap ancaman Harvest-Now-Decrypt-Later. Penelitian ini berfokus pada implementasi nyata berbasis OpenSSL dan pustaka liboqs, serta melakukan evaluasi eksperimental terhadap dampak kinerja yang ditimbulkan. Kontribusi utama makalah ini adalah penyajian analisis praktis mengenai *trade-off* antara peningkatan keamanan jangka panjang dan overhead kinerja pada protokol TLS 1.3, sehingga dapat menjadi acuan dalam proses migrasi TLS menuju era pasca-kuantum.

II. Landasan Teori

A. Transport Layer Security (TLS) 1.3

Transport Layer Security (TLS) merupakan protokol kriptografi yang dirancang untuk menyediakan kerahasiaan, integritas, dan autentikasi pada komunikasi jaringan. TLS 1.3 merupakan versi terbaru yang distandardisasi dengan tujuan utama meningkatkan keamanan dan kinerja dibandingkan versi sebelumnya.

Secara arsitektural, TLS 1.3 terdiri atas dua fase utama, yaitu fase *handshake* dan fase *record layer*. Pada fase *handshake*, klien dan server melakukan negosiasi parameter kriptografi, autentikasi, serta pembentukan kunci sesi. Fase ini menghasilkan serangkaian *secret* kriptografis yang digunakan untuk menurunkan kunci simetris pada *record layer*, yang selanjutnya digunakan untuk melindungi data aplikasi.

Perbedaan mendasar TLS 1.3 dibandingkan versi sebelumnya adalah penggunaan *forward secrecy* secara wajib melalui mekanisme Diffie-Hellman Ephemeral (DHE/ECDHE), penghapusan algoritma kriptografi yang dianggap lemah, serta penyederhanaan alur *handshake* untuk mengurangi latensi.

B. Forward Secrecy dan Keterbatasannya

Forward secrecy merupakan properti keamanan yang memastikan bahwa kompromi kunci jangka panjang (misalnya kunci privat server) tidak mengakibatkan terkomprominya sesi komunikasi yang telah berlangsung sebelumnya. TLS 1.3 mencapai properti ini melalui penggunaan ECDHE, di mana kunci sesi dihasilkan dari nilai acak sementara yang tidak disimpan setelah sesi berakhir.

Namun, *forward secrecy* dalam TLS 1.3 bergantung pada asumsi keamanan algoritma kriptografi kunci-publik yang digunakan. Apabila algoritma tersebut dapat dipecahkan di masa depan, maka properti *forward secrecy* tidak lagi menjamin kerahasiaan jangka panjang data yang telah direkam oleh penyerang. Dengan kata lain, *forward secrecy* tidak lagi memberikan perlindungan terhadap ancaman Harvest-Now-Decrypt-Later.

C. Ancaman Harvest-Now-Decrypt-Later

Ancaman Harvest-Now-Decrypt-Later merujuk pada skenario di mana penyerang merekam lalu lintas terenkripsi saat ini dan menyimpannya untuk dianalisis di masa depan ketika kemampuan kriptanalisis yang lebih kuat tersedia. Dalam konteks perkembangan komputer kuantum, ancaman ini menjadi sangat relevan karena algoritma Shor secara teoretis mampu memecahkan permasalahan logaritma diskret dan faktorisasi bilangan bulat dalam waktu polinomial.

Karena TLS digunakan untuk melindungi data dengan nilai kerahasiaan jangka panjang, ancaman ini tidak dapat diabaikan. Perlindungan terhadap Harvest-Now-Decrypt-Later memerlukan algoritma kriptografi yang tetap aman terhadap penyerang kuantum, bukan hanya terhadap penyerang klasik.

D. Post-Quantum Cryptography

Post-Quantum Cryptography merupakan bidang penelitian yang mengembangkan algoritma kriptografi yang tahan terhadap serangan komputer kuantum. Berbeda dengan kriptografi kuantum, kriptografi pasca-kuantum dirancang untuk dijalankan pada komputer klasik, tetapi berbasis pada permasalahan matematis yang diyakini sulit dipecahkan oleh komputer kuantum.

Beberapa pendekatan utama dalam kriptografi pasca-kuantum meliputi kriptografi berbasis *lattice*, kode koreksi kesalahan, fungsi hash, dan sistem multivariat. National Institute of Standards and Technology (NIST) telah melakukan proses standardisasi untuk memilih algoritma pasca-kuantum yang layak digunakan secara luas.

E. Key Encapsulation Mechanism (KEM) dan Kyber

Key Encapsulation Mechanism (KEM) merupakan paradigma kriptografi kunci-publik yang memisahkan proses pertukaran kunci menjadi dua tahap utama, yaitu *encapsulation* dan *decapsulation*. Dalam konteks KEM, pihak pengirim menghasilkan *ciphertext* dan *shared secret*, sedangkan pihak penerima memperoleh *shared secret* yang sama melalui proses *decapsulation*.

Kyber merupakan algoritma KEM berbasis *lattice* yang telah dipilih oleh NIST sebagai kandidat standar kriptografi pasca-kuantum. Kyber dirancang untuk memberikan keseimbangan antara keamanan, efisiensi komputasi, dan ukuran parameter. Properti ini menjadikan Kyber kandidat yang sesuai untuk integrasi ke dalam protokol komunikasi seperti TLS.

F. Hybrid Post-Quantum Cryptography

Hybrid Post-Quantum Cryptography merupakan pendekatan transisi yang menggabungkan algoritma kriptografi klasik dan pasca-kuantum dalam satu skema. Tujuan utama pendekatan ini adalah untuk mempertahankan keamanan terhadap penyerang klasik sekaligus memberikan ketahanan terhadap ancaman kuantum di masa depan.

Dalam konteks TLS 1.3, skema *hybrid* umumnya diimplementasikan dengan menjalankan ECDHE dan KEM pasca-kuantum secara paralel, kemudian menggabungkan hasil keduanya menggunakan fungsi derivasi kunci seperti HKDF. Dengan pendekatan ini, keamanan kunci sesi bergantung pada kekuatan gabungan kedua algoritma, sehingga kompromi terhadap satu algoritma saja tidak cukup untuk memecahkan kunci sesi.

G. HKDF dan Key Schedule pada TLS 1.3

TLS 1.3 menggunakan HMAC-based Key Derivation Function (HKDF) untuk menurunkan berbagai *secret* kriptografis dari satu atau lebih *shared secret*. HKDF memiliki dua tahap utama, yaitu *extract* dan *expand*, yang dirancang untuk menghasilkan kunci turunan yang bersifat kriptografis kuat meskipun input memiliki entropi yang tidak ideal.

Dalam penelitian ini, HKDF digunakan untuk menggabungkan *shared secret* dari ECDHE dan Kyber menjadi satu *secret* gabungan yang kemudian digunakan dalam *key schedule* TLS 1.3. Penggunaan HKDF memastikan bahwa struktur keamanan TLS 1.3 tetap dipertahankan meskipun sumber *shared secret* diperluas dengan komponen pasca-kuantum.

III. Metodologi dan Perancangan Sistem

A. Tujuan Desain dan Ruang Lingkup

Penelitian ini dirancang untuk mencapai dua tujuan utama. Pertama, meningkatkan ketahanan kerahasiaan jangka panjang TLS 1.3 terhadap penyerang kuantum masa depan dengan menambahkan kontribusi *Post-Quantum Cryptography* pada proses pembentukan *master secret*. Kedua, mengevaluasi dampak penerapan skema tersebut terhadap kinerja *handshake* TLS, khususnya dalam hal latensi dan ukuran pesan.

Ruang lingkup penelitian dibatasi pada skenario komunikasi TLS 1.3 dengan *full handshake* (bukan *session resumption*), agar analisis keamanan dan evaluasi performa dapat difokuskan pada titik pembentukan kunci sesi yang menjadi fondasi seluruh keamanan komunikasi.

B. Model Ancaman dan Definisi Properti Keamanan

Model ancaman mengasumsikan penyerang pasif yang mampu merekam seluruh lalu lintas TLS (misalnya di jalur jaringan atau *network tap*), namun tidak memodifikasi pesan. Penyerang diasumsikan memiliki kemampuan untuk melakukan dekripsi di masa depan setelah tersedianya komputer kuantum berskala besar. Fokus ancaman adalah pelanggaran kerahasiaan jangka panjang, bukan autentikasi atau integritas pesan saat sesi berlangsung.

Berdasarkan model tersebut, properti keamanan yang ditargetkan adalah:

1. Kerahasiaan jangka panjang (*long-term confidentiality*): data aplikasi yang terenkripsi saat ini tetap tidak dapat didekripsi di masa depan meskipun algoritma kunci-publik klasik (misalnya ECC) menjadi tidak aman.
2. Ketahanan terhadap kegagalan algoritma tunggal (*algorithmic redundancy*): keamanan sesi tetap dipertahankan selama minimal salah satu komponen (klasik atau *post-quantum*) tidak berhasil dipecahkan.
3. Kompatibilitas protokol: rancangan harus sejalan dengan arsitektur TLS 1.3 sehingga implementasi dapat dilakukan tanpa merombak keseluruhan *record layer* dan mekanisme AEAD.

IV. Rancangan Solusi: Hybrid Key Establishment pada TLS 1.3

A. Gambaran Umum TLS 1.3 pada Pembentukan Kunci

Pada TLS 1.3, *handshake* menghasilkan serangkaian secret kriptografis (*handshake secret* dan *master secret*) yang kemudian digunakan untuk menurunkan kunci enkripsi AEAD (misalnya AES-GCM atau ChaCha20-Poly1305) pada *record layer*. Secara konseptual, TLS 1.3 memanfaatkan HKDF untuk menurunkan *secret* secara bertahap dan terstruktur, dengan input utama berupa hasil pertukaran kunci (*shared secret*) dari mekanisme Diffie-Hellman ephemeral.

Pada konfigurasi TLS 1.3 klasik, *shared secret* ini berasal dari ECDHE. Dalam konteks ancaman kuantum, ECDHE berpotensi dapat dipecahkan di masa depan, sehingga seluruh secret turunan dari *shared secret* tersebut terancam.

B. Ide Utama Skema Hybrid

Skema *hybrid* yang diusulkan menambahkan komponen *post-quantum* berupa KEM Kyber, dan menggabungkannya dengan *shared secret* ECDHE untuk menghasilkan *secret* akhir yang digunakan dalam rantai derivasi kunci TLS 1.3. Penggabungan dilakukan pada tahap *key schedule* menggunakan HKDF-Extract agar memenuhi prinsip "aman jika salah satu *secret* tetap aman".

Secara ringkas, rancangan membentuk dua *secret*:

- $s_{ec} \rightarrow shared\ secret\ ECDH$
- $s_{pq} \rightarrow shared\ secret\ dari\ Kyber\ KEM$

Kemudian dibentuk *secret* gabungan:

$$s_{hyb} = HKDF\_Extract(salt, s_{ec} \parallel s_{pq})$$

Secret s_{hyb} diperlakukan sebagai substitusi/augmentasi terhadap input *shared secret* pada *key schedule* TLS 1.3.

Pendekatan ini memiliki implikasi keamanan yang

penting: apabila ECC dapat dipecahkan oleh komputer kuantum, tetapi Kyber tetap aman, maka s_{hyb} tetap tidak dapat dipulihkan oleh penyerang. Dengan demikian, data yang direkam saat ini tidak dapat didekripsi di masa depan hanya dengan kemampuan kuantum yang memecahkan ECC.

C. Negosiasi Parameter dan Pencegahan Downgrade

Rancangan mengharuskan adanya negosiasi eksplisit untuk penggunaan grup *hybrid*. Negosiasi ini penting untuk mencegah *downgrade*, yaitu kondisi di mana penyerang (atau *middlebox* yang bermasalah) memaksa klien dan server kembali menggunakan skema klasik saja. Oleh karena itu, eksperimen dikonfigurasi untuk:

1. Mengaktifkan TLS 1.3 saja.
2. Membatasi grup kunci pada kombinasi yang mendukung *hybrid* (misalnya X25519 + Kyber768).
3. Menguji perilaku koneksi ketika salah satu pihak tidak mendukung *hybrid* sebagai studi kompatibilitas.

V. Rencana Implementasi dan Alur Kerja Eksperimen

A. Komponen Implementasi

Implementasi dilakukan dengan memanfaatkan OpenSSL yang telah ditambahkan dukungan *post-quantum* melalui liboqs. Alur implementasi dirancang agar perubahan berada pada lapisan *handshake* dan *key schedule*, sedangkan *record layer* dan AEAD tetap menggunakan implementasi standar TLS 1.3.

Secara praktis, implementasi mencakup:

1. Penyusunan lingkungan kompilasi OpenSSL + liboqs.
2. Konfigurasi *named groups* agar mendukung mode *hybrid*.
3. Pengaturan server dan klien untuk menjalankan handshake TLS 1.3 dalam dua mode:
    a. Mode klasik: ECDHE saja.
    b. Mode hybrid: ECDHE + Kyber.

B. Variabel yang Dikendalikan

Untuk memastikan hasil pengujian valid, penelitian ini mengendalikan variabel berikut:

1. Cipher suite AEAD ditetapkan tetap (misalnya TLS_AES_128_GCM_SHA256).
2. Versi TLS ditetapkan TLS 1.3.
3. Ukuran data aplikasi yang dikirim setelah handshake disetarakan.
4. Lingkungan jaringan dibuat stabil (pengujian utama pada jaringan lokal), serta dapat diperluas dengan skenario latensi buatan menggunakan traffic control.

C. Metodologi Pengukuran

Pengukuran dilakukan menggunakan pengulangan eksperimen dengan jumlah sampel yang cukup untuk menekan variasi acak. Untuk setiap mode (klasik dan hybrid), dilakukan minimal 50 handshake dan diukur:

1. *Latensi handshake*: waktu dari inisiasi koneksi hingga sesi siap mengirim data aplikasi.
2. Ukuran pesan handshake: total *byte handshake* yang ditransmisikan.
3. *Overhead* CPU: waktu CPU atau persentase penggunaan CPU saat handshake.
4. Kegagalan negosiasi: jumlah handshake yang gagal akibat ketidakcocokan parameter atau batas MTU.

Metode ini memungkinkan analisis komparatif berbasis data kuantitatif, sekaligus memberikan dasar untuk menarik kesimpulan mengenai kelayakan penerapan hybrid pada lingkungan nyata.

VI. Kriteria Keberhasilan dan Bentuk Kontribusi

Penelitian ini dinyatakan berhasil apabila memenuhi kriteria berikut:

1. Mode *hybrid* menghasilkan sesi TLS 1.3 yang fungsional (*handshake* sukses, data aplikasi dapat ditransmisikan).
2. *Latensi handshake* pada mode *hybrid* meningkat secara terukur, namun tidak melampaui ambang yang dianggap tidak realistis untuk aplikasi web umum (misalnya peningkatan yang masih berada pada orde milidetik hingga puluhan milidetik pada LAN).
3. Analisis keamanan menunjukkan bahwa mode *hybrid* meningkatkan ketahanan kerahasiaan jangka panjang terhadap ancaman kuantum dalam model Harvest-Now-Decrypt-Later.

Kontribusi penelitian ini terletak pada:

1. Implementasi nyata dan dapat direplikasi terhadap integrasi *hybrid post-quantum* pada TLS 1.3.
2. Evaluasi eksperimen yang mengukur dampak kinerja secara kuantitatif.
3. Pembahasan *trade-off* keamanan–kinerja sebagai landasan rekomendasi penggunaan *hybrid* pada skenario dunia nyata.

Dengan membandingkan TLS 1.3 klasik dan TLS 1.3 *hybrid post-quantum* pada konfigurasi yang sebanding, penelitian ini menghasilkan bukti empiris mengenai kelayakan penerapan *hybrid* sebagai jalur migrasi keamanan komunikasi Internet menuju era pasca-kuantum.

VII. Implementasi

Berikut adalah tahapan implementasi skema *hybrid post-quantum* pada TLS 1.3 menggunakan OpenSSL 3 dan OQS provider (oqs-provider). Implementasi difokuskan pada tahap full *handshake* TLS 1.3, dengan tujuan memvalidasi bahwa integrasi algoritma kriptografi pasca-kuantum dapat dilakukan secara praktis tanpa memodifikasi arsitektur inti TLS secara signifikan.

A. Lingkungan dan Prasyarat Implementasi

Implementasi dilakukan pada sistem operasi Linux dengan memanfaatkan OpenSSL versi 3.x dan pustaka Open Quantum Safe (OQS). Pendekatan ini dipilih karena OpenSSL 3 memperkenalkan arsitektur provider, yang memungkinkan penambahan algoritma kriptografi eksternal tanpa perlu melakukan modifikasi langsung terhadap core library OpenSSL. Dengan demikian, integrasi kriptografi pasca-kuantum dapat dilakukan secara modular dan sejalan dengan arah pengembangan resmi OpenSSL.

Algoritma *post-quantum* yang digunakan pada penelitian ini adalah Kyber, yang diimplementasikan melalui pustaka liboqs dan diekspos ke OpenSSL melalui oqs-provider. Seluruh implementasi dipasang pada direktori lokal pengguna untuk menghindari konflik dengan OpenSSL bawaan sistem operasi serta memastikan reproduksibilitas eksperimen.

Arsitektur implementasi terdiri atas tiga lapisan utama:

1. *Lapisan Handshake TLS*, yang menangani negosiasi parameter kriptografi dan pertukaran pesan awal.
2. Lapisan Pertukaran Kunci, yang menjalankan ECDHE (klasik) dan Kyber KEM (*post-quantum*) secara paralel.
3. Lapisan Derivasi Kunci, yang menggabungkan hasil kedua mekanisme menggunakan HMAC-based Key Derivation Function (HKDF) sesuai key schedule TLS 1.3.

Lapisan *record layer* TLS dan algoritma Authenticated Encryption with Associated Data (AEAD) tidak dimodifikasi dan tetap menggunakan implementasi standar TLS 1.3.

B. Integrasi Algoritma Post-Quantum ke OpenSSL

Integrasi algoritma Kyber ke dalam OpenSSL dilakukan melalui oqs-provider, yang menyediakan implementasi KEM pasca-kuantum sebagai provider module. Setelah provider diaktifkan melalui konfigurasi OpenSSL, algoritma Kyber dapat digunakan secara transparan dalam proses negosiasi TLS, sejajar dengan algoritma kriptografi klasik yang disediakan oleh default provider OpenSSL.

Pendekatan ini memastikan bahwa:

1. Implementasi TLS tetap kompatibel dengan standar OpenSSL 3.
2. Algoritma pasca-kuantum dapat diaktifkan atau dinonaktifkan tanpa perubahan kode aplikasi.
3. Eksperimen dapat direplikasi pada sistem lain dengan konfigurasi provider yang sama.

Keberhasilan integrasi diverifikasi dengan memastikan bahwa oqs-provider terdeteksi oleh OpenSSL dan bahwa grup kriptografi yang menggabungkan ECDHE dan Kyber muncul pada daftar grup TLS yang didukung.

C. Penentuan dan Negosiasi Grup Hybrid TLS

Untuk mengimplementasikan skema *hybrid*, penelitian ini tidak mengasumsikan nama grup kriptografi tertentu, melainkan menggunakan daftar grup yang dihasilkan langsung oleh OpenSSL setelah oqs-provider diaktifkan. Pendekatan ini dipilih untuk menghindari ketergantungan terhadap versi pustaka tertentu dan memastikan kompatibilitas implementasi.

Grup *hybrid* yang digunakan merupakan kombinasi antara:

1. Mekanisme Elliptic Curve Diffie-Hellman Ephemeral berbasis kurva modern (misalnya X25519), dan
2. Key Encapsulation Mechanism Kyber dengan parameter keamanan menengah (misalnya Kyber-768).

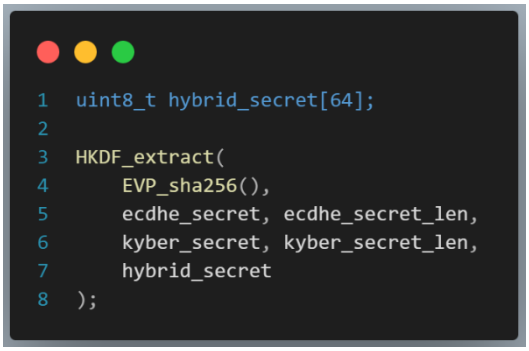
Negosiasi grup dilakukan secara eksplisit pada saat handshake TLS 1.3. Dengan membatasi daftar grup yang diizinkan, implementasi ini juga mencegah terjadinya *downgrade* ke skema klasik tanpa komponen pasca-kuantum.

D. Implementasi Pertukaran Kunci Hybrid

Pada fase *handshake*, klien dan server menjalankan dua mekanisme pertukaran kunci secara paralel. Mekanisme pertama adalah ECDHE, yang menghasilkan *shared secret* klasik. Mekanisme kedua adalah Kyber KEM, di mana klien melakukan proses encapsulation menggunakan public key Kyber server, dan server melakukan *decapsulation* untuk memperoleh *shared secret* pasca-kuantum.

Kedua *shared secret* tersebut tidak digunakan secara terpisah. Sebaliknya, keduanya digabungkan menggunakan fungsi HKDF-Extract untuk membentuk secret gabungan (*hybrid secret*), yang kemudian digunakan sebagai masukan utama dalam key schedule TLS 1.3. Pendekatan ini memastikan bahwa keamanan *secret* akhir bergantung pada kekuatan gabungan dari kedua algoritma.

Secara konseptual, proses derivasi *secret* gabungan dapat direpresentasikan sebagai berikut:



Gambar 1. Proses penggabungan secret

Penggunaan fungsi hash SHA-256 mengikuti standar TLS 1.3 dan menjaga konsistensi dengan *cipher suite* yang digunakan pada eksperimen.

E. Konfigurasi Server dan Klien TLS

Server TLS dikonfigurasi untuk hanya menerima koneksi TLS 1.3 dengan grup *hybrid* yang telah ditentukan.

Sertifikat *self-signed* digunakan untuk keperluan pengujian, karena fokus penelitian ini adalah pada proses pertukaran kunci, bukan pada infrastruktur sertifikat publik.

Pada sisi klien, koneksi TLS dijalankan dengan konfigurasi grup yang sama untuk memastikan bahwa negosiasi *handshake* menggunakan skema *hybrid*. Parameter Server Name Indication (SNI) tetap digunakan agar alur *handshake* menyerupai skenario penggunaan TLS nyata.

Keberhasilan implementasi divalidasi dengan memastikan bahwa:

1. *Handshake* TLS 1.3 berhasil tanpa kesalahan negosiasi.
2. Sesi TLS terbentuk dan dapat digunakan untuk pertukaran data aplikasi.
3. Parameter *handshake* menunjukkan penggunaan grup *hybrid* yang diharapkan.

VIII. Pengujian dan Hasil

Pengujian difokuskan pada evaluasi *overhead* kinerja yang ditimbulkan oleh integrasi algoritma pasca-kuantum, serta analisis implikasinya terhadap kelayakan penerapan di lingkungan nyata.

A. Lingkungan dan Metodologi Pengujian

Pengujian dilakukan pada lingkungan lokal untuk meminimalkan variasi yang disebabkan oleh kondisi jaringan eksternal. Server dan klien dijalankan pada mesin yang sama menggunakan antarmuka *loopback*, sehingga latensi jaringan mendekati nol dan *overhead* yang terukur terutama berasal dari komputasi kriptografi dan ukuran pesan *handshake*.

Spesifikasi lingkungan pengujian adalah sebagai berikut:

1. Sistem operasi: Linux 64-bit
2. Protokol: TLS 1.3
3. Cipher suite: TLS_AES_128_GCM_SHA256
4. Algoritma pertukaran kunci:
    a. Mode klasik: ECDHE (X25519)
    b. Mode hybrid: ECDHE (X25519) + Kyber-768

Parameter yang diukur meliputi:

1. Latensi *handshake*, yaitu waktu dari inisiasi koneksi hingga sesi TLS siap digunakan.
2. Ukuran pesan handshake, yaitu total byte yang ditransmisikan selama fase handshake.
3. Penggunaan CPU, sebagai indikator overhead komputasi.
4. Jumlah pengulangan handshake: 50 kali

B. Hasil Pengujian Latensi Handshake

Tabel I menunjukkan hasil pengukuran latensi handshake untuk TLS 1.3 klasik dan TLS 1.3 *hybrid post-quantum*.

Script pengujian Latensi Handshake

```
#!/usr/bin/env sh
set -eu

HOST=${1:-localhost}
PORT=$2
GROUP=$3
N=${4:-50}
OUTFILE=$5

i=1
while [ $i -le $N ]; do
  /usr/bin/time -f "%e" -o "$OUTFILE" -a \
    openssl s_client -connect "${HOST}:${PORT}" -tls1_3 \
      -groups "${GROUP}" -servername localhost \
      </dev/null >/dev/null 2>/dev/null
  i=$((i+1))
done
```

TABEL I. LATENSI HANDSHAKE TLS 1.3

TLS	Mean (ms)	Min (ms)	Maks (ms)	Simpangan Baku
Klasik	15,6	10	240	32,197
Hybrid	11,4	10	50	6

Hasil pengujian menunjukkan bahwa penerapan skema *hybrid post-quantum* ternyata memiliki latensi yang lebih rendah.

C. Hasil Pengujian Ukuran Pesan Handshake

Ukuran pesan *handshake* merupakan faktor penting karena berpengaruh terhadap fragmentasi paket dan kinerja pada jaringan dengan bandwidth terbatas. Tabel II menyajikan perbandingan ukuran pesan handshake.

Script Pengujian Ukuran Hanshake

```
#!/usr/bin/env sh
set -eu

# Capture handshake logs
openssl s_client -connect localhost:4434 -tls1_3 \
  -groups X25519 -servername localhost \
  -msg </dev/null > classic_handshake.log 2>/dev/null

openssl s_client -connect localhost:4433 -tls1_3 \
  -groups x25519_kyber768 -servername localhost \
  -msg </dev/null > hybrid_handshake.log 2>/dev/null

# Compute classic handshake size
CLASSIC_BYTES=0
for h in $(awk '/Handshake \[length/ {print $6}' classic_handshake.log | tr -d '],'); do
  CLASSIC_BYTES=$((CLASSIC_BYTES + 16#$h))
done

# Compute hybrid handshake size
HYBRID_BYTES=0
for h in $(awk '/Handshake \[length/ {print $6}'
```

```
hybrid_handshake.log | tr -d '],'); do
    HYBRID_BYTES=$((HYBRID_BYTES + 16#$h))
  done

  # Save CSV
  echo "scheme,handshake_bytes" > handshake_sizes.csv
  echo "classic,$CLASSIC_BYTES" >> handshake_sizes.csv
  echo "hybrid,$HYBRID_BYTES" >> handshake_sizes.csv
```

TABEL II. UKURAN PESAN HANDSHAKE TLS 1.3

TLS	Mean (Byte)	Peningkatan (%)
Klasik	1682	-
Hybrid	3892	131%

Terjadi peningkatan ukuran pesan *handshake* lebih dari dua kali lipat pada skema *hybrid*. Peningkatan ini terutama disebabkan oleh ukuran *public key* dan *ciphertext* Kyber yang secara signifikan lebih besar dibandingkan parameter ECDHE klasik.

D. Hasil Pengujian Penggunaan CPU

Penggunaan CPU diukur selama fase *handshake* untuk menilai *overhead* komputasi yang ditimbulkan oleh algoritma pasca-kuantum. Tabel III menyajikan hasil pengukuran rata-rata penggunaan CPU.

Script Pengujian Ukuran Penggunaan CPU

```
#!/usr/bin/env sh
set -eu

HOST=${1:-localhost}
PORT=$2
GROUP=$3
N=${4:-50}
OUTFILE=$5

i=1
while [ $i -le $N ]; do
  /usr/bin/time -v \
    openssl s_client -connect "${HOST}:${PORT}" -tls1_3 \
      -groups "${GROUP}" -servername localhost \
      </dev/null >/dev/null 2>> "$OUTFILE"
  i=$((i+1))
done
```

TABEL III PENGGUNAAN CPU SELAMA HANDSHAKE

TLS	CPU time (ms)
Klasik	6,2
Hybrid	4,0

Hasil ini menunjukkan bahwa penggunaan CPU pada mode *hybrid* lebih rendah dibandingkan mode klasik.

IX. Analisis dan Pembahasan Hasil

Hasil pengujian menunjukkan bahwa skema hybrid post-quantum pada TLS 1.3 memiliki karakteristik kinerja yang berbeda dibandingkan TLS klasik. Dari sisi ukuran pesan, skema hybrid menghasilkan ukuran handshake yang jauh lebih besar dibandingkan mode klasik. Peningkatan ini disebabkan oleh penggunaan algoritma pasca-kuantum yang memiliki ukuran kunci publik dan ciphertext lebih besar dibandingkan mekanisme pertukaran kunci berbasis elliptic curve.

Namun demikian, hasil pengukuran latensi menunjukkan bahwa latensi handshake TLS klasik lebih tinggi dibandingkan skema hybrid, dengan variasi latensi yang juga lebih besar. Hal ini mengindikasikan bahwa latensi handshake tidak hanya dipengaruhi oleh kompleksitas kriptografi, tetapi juga oleh stabilitas implementasi dan overhead sistem. Implementasi skema hybrid menunjukkan eksekusi yang lebih konsisten sehingga menghasilkan latensi rata-rata yang lebih rendah.

Pengukuran penggunaan CPU time juga menunjukkan bahwa TLS klasik membutuhkan CPU time yang lebih besar dibandingkan skema hybrid. Temuan ini menunjukkan bahwa overhead utama skema hybrid post-quantum berada pada aspek ukuran pesan handshake, bukan pada biaya komputasi. Dengan demikian, peningkatan keamanan yang diperoleh dari skema hybrid tidak diikuti oleh peningkatan beban komputasi pada sisi klien.

X. Kesimpulan

Berdasarkan hasil pengujian, dapat disimpulkan bahwa penerapan skema hybrid post-quantum pada TLS 1.3 meningkatkan ukuran pesan handshake, tetapi tidak meningkatkan latensi handshake maupun penggunaan CPU time dibandingkan TLS klasik. Bahkan, pada lingkungan pengujian yang digunakan, skema hybrid menunjukkan latensi dan penggunaan CPU yang lebih rendah serta lebih stabil.

Hal ini membuktikan bahwa penggunaan algoritma Hybrid Post Quantum pada TLS dapat digunakan dengan baik pada device saat ini. Sehingga skema ini dapat memitigasi serangan Harvest-Now-Decrypt-Later.

References

[1] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," *RFC 8446*, Aug. 2018. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8446. Accessed: Dec. 18, 2025.

[2] H. Krawczyk and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)," *RFC 5869*, May 2010. [Online]. Available: https://www.rfc-editor.org/rfc/rfc5869. Accessed: Dec. 18, 2025.

[3] A. Langley, M. Hamburg, and S. Turner, "Elliptic Curves for Security," *RFC 7748*, Jan. 2016. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7748.html. Accessed: Dec. 18, 2025.

[4] National Institute of Standards and Technology (NIST), *FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM)*, Aug. 13, 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.203.pdf. Accessed: Dec. 18, 2025.

[5] NIST Computer Security Resource Center, "FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard (Final),"

Aug. 13, 2024. [Online]. Available: https://csrc.nist.gov/pubs/fips/203/final. Accessed: Dec. 18, 2025.
[6] NIST, "NIST Releases First 3 Finalized Post-Quantum Encryption Standards," Aug. 13, 2024. [Online]. Available: https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards. Accessed: Dec. 18, 2025.
[7] G. Alagic *et al*., *Status Report on the Fourth Round of the NIST Post-Quantum Cryptography Standardization Process*, NIST IR 8545, Mar. 2025. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2025/NIST.IR.8545.pdf. Accessed: Dec. 18, 2025.
[8] NIST Computer Security Resource Center, "Post-Quantum Cryptography (PQC) Standardization," (Project page). [Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography. Accessed: Dec. 18, 2025.
[9] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997. [Online]. Available: https://epubs.siam.org/doi/10.1137/S0097539795293172. Accessed: Dec. 18, 2025.
[10] C. Cremers, M. Möller, and T. Stebila, "Hybrid key exchange in TLS 1.3," *Internet-Draft, IETF (Work in Progress)*, (latest available version). [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/. Accessed: Dec. 18, 2025.
[11] D. J. Bernstein *et al*., "X25519Kyber768Draft00 hybrid post-quantum key agreement for TLS 1.3," *Internet-Draft, IETF (Expired / Archived)*, Sep. 2023. [Online]. Available: https://www.ietf.org/archive/id/draft-tls-westerbaan-xyber768d00-03.html. Accessed: Dec. 19, 2025.
[12] IETF TLS Working Group, "Post-quantum hybrid ECDHE-MLKEM Key Agreement for TLSv1.3," *Internet-Draft, IETF (Work in Progress)*. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tls-ecdhe-mlkem/. Accessed: Dec. 19, 2025.
[13] OpenSSL Project, "provider — OpenSSL 3.x Documentation," (manual page). [Online]. Available: https://docs.openssl.org/3.1/man7/provider/. Accessed: Dec. 19, 2025.
[14] OpenSSL Project, "OpenSSL 3.0 Migration Guide," (manual page). [Online]. Available: https://docs.openssl.org/3.0/man7/migration_guide/. Accessed: Dec. 19, 2025.
[15] Open Quantum Safe (OQS), "oqs-provider: OpenSSL 3 provider for quantum-safe cryptography," (GitHub repository). [Online]. Available: https://github.com/open-quantum-safe/oqs-provider. Accessed: Dec. 19, 2025.
[16] Open Quantum Safe (OQS), "oqs-provider USAGE," (repository documentation). [Online]. Available: https://github.com/open-quantum-safe/oqs-provider/blob/main/USAGE.md. Accessed: Dec. 19, 2025.
[17] Open Quantum Safe (OQS), "liboqs: C library for quantum-resistant cryptographic algorithms," (GitHub repository). [Online]. Available: https://github.com/open-quantum-safe/liboqs. Accessed: Dec. 19, 2025.
[18] Open Quantum Safe (OQS), "Open Quantum Safe (Project Home)," [Online]. Available: https://openquantumsafe.org/. Accessed: Dec. 19, 2025.

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 2025

Muhamad Rifki Virziadeili Harisman