(20) **RI Permohonan Paten**

(19) **ID** **(11)** **No Pengumuman : 2022/S/02281** **(13) A**

(51) **I.P.C : Int.Cl./**

(21) **No. Permohonan Paten :** S00202209537

(22) **Tanggal Penerimaan Permohonan Paten :**
05 September 2022

(30) **Data Prioritas :**
(31) Nomor (32) Tanggal (33) Negara

(43) **Tanggal Pengumuman Paten :**
13 September 2022

(71) **Nama dan Alamat yang Mengajukan Permohonan Paten :**
Universitas Siliwangi
Jl. Siliwangi No.24, Kahuripan, Kec. Tawang, Kab. Tasikmalaya, Jawa Barat 46115 Indonesia

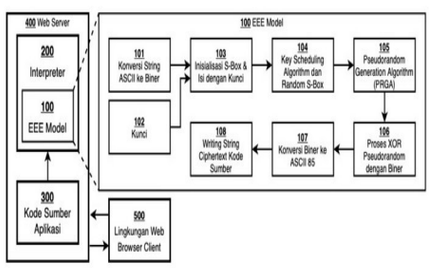
(72) **Nama Inventor :**
Alam Rahmatulloh,ID
Rohmat Gunawan,ID

(74) **Nama dan Alamat Konsultan Paten :**

(54) **Judul Invensi :** MODEL ENCRYPTION EXECUTION ENGINE (EEE) UNTUK MENINGKATKAN KEAMANAN PERLINDUNGAN KODE SUMBER

(57) **Abstrak :**

Suatu model proses mesin eksekusi enkripsi (Model Execution Engine (EEE)) pada web server untuk perlindungan keamanan kode sumber aplikasi berbasis web dari reverse engineering. Urgensi invensi ini yaitu perlindungan aplikasi berbasis web yang rentan terhadap modifikasi, karena kode sumber pada aplikasi web biasanya kode sumber aslinya langsung, bukan hasil compiler/executable. Metode penyimpanan lebih aman karena dibuat dalam bentuk library web server, kemudian proses dekripsi dilakukan bersamaan dengan proses interpreter pada web server. Tahapan model dimulai konversi string kode sumber asli atau terenkripsi dalam ASCII menjadi bentuk biner, penetapan kunci, inisialisasi s-box dan mengisi dengan kunci, proses key scheduling algorithm dan random s-box, pseudorandom generation algorithm (PRGA), XOR pseudorandom dengan biner, konversi biner ke ASCII 85, penulisan kode sumber terenkripsi atau hasil dekripsi dilanjutkan interpreter menjadi bentuk skrip sisi klien (client side scripting). Hasil penerapan invensi tahap kesatu yaitu kode sumber asli berubah menjadi kode acak (terenkripsi/ obfuscation) dan tidak bisa dimodifikasi. Jika terjadi modifikasi maka aplikasi web tidak dapat dijalankan (error). Tahap kedua kode sumber terenkripsi (obfuscation) didekripsi bersamaan dengan interpreter dan ditampilkan pada browser klien. Dengan model invensi EEE tidak mudah dimodifikasi (reverse engineering) dan terpisah dengan kode sumber aplikasi web.



Gambar 2

Deskripsi

MODEL ENCRYPTION EXECUTION ENGINE (EEE) UNTUK MENINGKATKAN KEAMANAN PERLINDUNGAN KODE SUMBER

Bidang Teknik Invensi

Invensi ini berhubungan dengan suatu model proses mesin eksekusi dalam web server menerapkan enkripsi untuk perlindungan keamanan kode sumber aplikasi berbasis web dari *reverse engineering*.

Latar Belakang Invensi

Permasalahan utama yang ingin diselesaikan oleh invensi ini adalah mengenai perlindungan *reverse engineering* pada development berbasis web, karena instalasi dan konfigurasi aplikasi berbasis web tidak menggunakan *excutable file*, melainkan kode sumber langsung yang terbuka. Sehingga kode sumber tersebut sangat rentan terhadap keamanan modifikasi data kode sumber. Berdasarkan hasil penelusuran beberapa paten yang mendekati invensi ini yaitu nomor paten US20080141336A1 yang berjudul Secure execution environments for process models. Invensi ini membahas keamanan lingkungan eksekusi, dan mesin eksekusi proses yang dikonfigurasi untuk mengeksekusi model proses dalam lingkungan eksekusi. Invensi tersebut membahas keamanan mesin eksekusi secara umum, sementara invensi yang diusulkan ini fokus terhadap model penerapan enkripsi pada proses mesin eksekusi dalam web server. Begitu juga proses dekripsi dilakukan mesin eksekusi bersamaan dalam interpreter bahasa pemrograman kode sumber menjadi *skrip sisi klien (client side scripting)*. Invensi lain yaitu nomor IDP000065856 berjudul Metode Untuk Meningkatkan Keamanan Sistem Informasi Menggunakan Enkripsi Data Melalui *Block*

Cipher Dengan Isi Algoritma Yang Dinamis. Invensi ini juga menggunakan enkripsi dalam melindungi kode sumbernya, tidak seperti perlindungan enkripsi lainnya, model enkripsi ini disimpan pada layer web server sehingga terpisah dengan kode sumber aplikasi. Selanjutnya nomor paten IDP000018247 dengan judul Metode Untuk Menentukan Algoritma Enkripsi Pada Komunikasi Keamanan Berdasarkan Pada Kode Negara Bergerak, membahas mengenai penerapan algoritma enkripsi yang dinamis berdasarkan jumlah kode negara stasiun bergerak *(Mobile Country Code/MCC)*. Namun dalam invensi ini komunikasi keamanan dilakukan bersamaan dengan proses interpretasi kode sumber menjadi kode HTML/Css/Js yang bisa dibaca oleh klien. Kode sumber yang terenkripsi diproses dekripsi bersamaan dengan proses interpreter bahasa kode sumber menjadi *skrip sisi klien (client side scripting)*.

Berdasarkan uraian dari beberapa invensi sebelumnya, maka invensi ini merupakan langkah inovatif untuk mengatasi permasalahan perlindungan kode sumber aplikasi berbasis web terhadap *reverse engineering*. Sehingga kode sumber aman karena telah terenkripsi (kode sumber teracak), namun aplikasi web tersebut tetap dapat terakses oleh klien. Selain itu invensi model proses eksekusi enkripsi ini terletak pada web server, sehingga proses enkripsi dan dekripsi dilakukan bersamaan dengan interpreter bahasa pemrograman kode sumber.

Uraian Singkat Invensi

Bertitik-tolak dari hal-hal tersebut di atas, dan untuk memberikan hasil yang lebih baik dan lebih sempurna, maka tujuan dari invensi ini adalah untuk meningkatkan perlindungan terhadap kode sumber aplikasi berbasis web. Dengan penggunaan aloritma enkripsi pada proses mesin eksekusi web server yang menjadikan kode sumber terobfukasi maka kode sumber menjadi aman tidak dapat dimodifikasi.

Masih menjadi tujuan lain dari invensi ini, dekripsi kode sumber yang sudah terenkripsi (obfukasi) juga dilakukan pada proses mesin eksekusi *web server* sehingga *decryptor* terpisah dari aplikasi berbasis webnya. Hal tersebut mempersulit proses pembobolan, karena *decryptor* dalam bentuk *library web server*.

Uraian Singkat Gambar

Gambar 1 adalah Proses Client Server Aplikasi Berbasis Web;

Gambar 2 adalah Uraian Invensi Proses;

Gambar 3 adalah Uraian Detail Invensi Proses.

Proses dan cara kerja **500** aplikasi berbasis *web klien-server* dimulai dari permintaan klien kemudian pada **400** *web server* akan melakukan proses *compiler* dari bahasa *server side scripting* seperti *PHP* dimanipulasi data atau pengambilan data dari atau ke database, kemudian di-*interpreter* menjadi bahasa *skrip sisi klien (client side scripting)* dalam hal ini adalah HTML/CSS/Javascript. Sehingga tampilan yang muncul pada web browser di *klien* merupakan bahasa *skrip sisi klien (client side scripting)*. Permasalahan yang ada yaitu kode sumber aplikasi yang disimpan pada *web server* secara langsung terbuka sehingga kode sumber tersebut mudah untuk dimodifikasi. Selain itu walaupun sudah dilakukan perlindungan menggunakan *encryptor/decryptor*, namun penyimpanannya biasanya bersatu dengan kode sumber aplikasinya, sehingga tentunya sama juga menjadi celah *encryptor/descryptor*-nya dengan mudah dimodifikasi atau bahkan dihilangkan.

Gambar 1 merupakan usulan invensi, dengan model *encryption executing engine (EEE)* ini maka perlindungan terhadap kode sumber menjadi lebih tinggi. Karena *encryptor/decryptor*-nya tidak mudah untuk dibongkar dan

modifikasi dan sudah berbentuk library pada *web server*. Begitu juga kode sumber aplikasi yang sudah terobfuskasi (di enkripsi), sehingga kode sumber berbentuk acak.

Uraian Lengkap Invensi

Mengacu pada gambar-gambar invensi, model ini terdiri dari dua tahap proses yaitu enkripsi (kode sumber asli) dan dekripsi (kode sumber terenkripsi/terobfukasi). Tahap pertama proses perlindungan kode sumber dimulai dari **301** kode sumber asli dilakukan proses enkripsi dengan **100** model *Encryption Executing Engine (EEE)*. Model *Encryption Executing Engine (EEE)* terdiri dari beberapa tahap, dimulai dari **101** konversi string pada kode sumber ASCII ke bentuk biner, kemudian dilakukan **103** inisialisasi *s-box* dan mengisinya dengan **102** kunci yang digunakan. Setelah itu dilakukan proses **104** *key scheduling algorithm* dan *random s-box*, hasilnya dilakukan tahap selanjutnya **105** *pseudorandom generation algorithm* (PRGA). Hasil dari proses tersebut selanjutnya dilakukan proses **106** XOR *pseudorandom* dengan biner. Biner tersebut dikonversi dengan **107** ASCII 85, kemudian dilanjutkan dengan **108** penulisan kembali pada kode sumber.

Tahap kedua, kode sumber yang **302** terenkripsi *(obfuscation code)* di proses oleh **400** *web server* dengan memanggil **100** model EEE *(encryptor/decryptor)* dan di **200** interpreter, kemudian **402** hasilnya dalam bentuk **403** HTML/Css/Js dikirim ke klien maka **501** aplikasi berbasis web muncul pada layar web browser.

Invensi berupa model ini dapat meningkatkan keamanan perlindungan terhadap kode sumber, karena kode sumber sudah terenkripsi *(obfuscation code)* dan sulit untuk dimodifikasi atau *reverse engineering.* Kode sumber terenkripsi dan sulit dibaca akan tetapi kode sumber tersebut tetap dapat diakses

oleh klien dan muncul sebagai aplikasi berbasis *web* pada layar *web browser*.

5

Klaim

1. Suatu model proses mesin eksekusi enkripsi *(encryption execution engine (EEE))* untuk meningkatkan keamanan perlindungan kode sumber dalam bentuk library pada web server, dimana proses yang dilakukan adalah sebagai berikut:
   - Konversi string kode sumber asli atau yang sudah terenkripsi dalam ASCII menjadi bentuk biner;
   - Penetapan kunci;
   - Melakukan inisialisasi s-box dan mengisi dengan kunci;
   - Proses *key scheduling algorithm* dan *random s-box;*
   - Proses *pseudorandom generation algorithm (PRGA);*
   - Proses XOR *pseudorandom* dengan biner;
   - Konversi biner ke ASCII 85;
   - Penulisan kembali kode sumber terenkripsi atau hasil dekripsi dilanjutkan interpreter menjadi bentuk *skrip sisi klien (client side scripting);*
   - Aplikasi berbasis web muncul pada web browser klien.

2. Suatu model proses mesin eksekusi enkripsi *(encryption execution engine (EEE))* untuk meningkatkan keamanan perlindungan kode sumber dalam bentuk library pada web server, sesuai dengan klaim 1, *yang dicirikan* dengan proses yang terjadi tersebut terdiri dari dua bagian yaitu:
   - Bagian pertama merupakan proses enkripsi atau perubahan kode sumber asli menjadi terenkripsi/obfukasi;
   - Bagian tahap kedua adalah proses dekripsi atau pembacaan kode sumber terenkripsi dan perubahan menjadi kode *skrip sisi klien (client side scripting)* agar aplikasi web dapat muncul dan terbaca pada *web browser klien*.