PENERAPAN COVER GENERATION STEGANOGRAFI DAN KRIPTOGRAFI RSA PADA ENKRIPSI KUNCI SIMETRIS AES RIJNDAEL UNTUK KEAMANAN DATA DALAM JARINGAN LAN PT. HERO SUPERMARKET TBK.

Ali Mulyanto[1)], Abimanyu[2)]
[1)]Program Studi: Teknik Informatika, STMIK Cikarang
email: ali.stmikcikarang@gmail.com
[2)]Program Studi: Manajemen Informatika, STMIK Cikarang
email : aby.edukasi@gmail.com

ABSTRAK

Komunikasi informasi dalam infrastruktur jaringan Lokal Area Network (LAN) PT. Hero Supermarket Tbk. berpotensi munculnya celah keamanan data. Pihak yang tidak sah dapat mengeksplorasi dan mengungkap isi informasi secara ilegal, sehingga menimbulkan resiko terjadinya tindakan interupsi, intersepsi, modifikasi dan fabrikasi. Menghindari hal tersebut diperlukan tindakan proteksi menggunakan metode kriptografi dan steganografi. Kombinasi kunci publik dan kunci privat kriptografi RSA menghasilkan kunci enkripsi dan dekripsi yang berbeda antara pengirim dan penerima. Kriptografi kunci rahasia AES Rijndael menghasilkan berkas yang terenkripsi. Data yang terenkripsi disisipkan ke media lain dengan metode Cover Generation Steganografi, sehingga keberadaan data asli tidak dapat diketahui oleh pihak lain. Verifikasi komparatif fungsi hash SHA-256 menjamin otentikasi data benar-benar valid dari pengaruh perubahan data. Pengujian primalitas Miller-Rabin diuji untuk menghasilkan nilai eksponen bilangan prima acak kunci RSA yang besar. Ini berarti kemampuan kunci RSA dapat disimpulkan sebagai kunci utama yang berpengaruh dari rangkaian proses enkripsi dan dekripsi data.

Kata kunci: keamanan data, kriptografi RSA, enkripsi, AES Rijndael, Cover Generation Steganografi

1. Pendahuluan

Komunikasi informasi dalam data terangkai dalam infrastruktur *Local Area Network* (LAN) di lingkungan kerja dengan ditandai pengguna perangkat komputer yang berinteraksi antar pengguna perangkat komputer lainnya, baik akses secara individu maupun antar divisi, sehingga menimbulkan kerentanan keamanan data terhadap informasi didalamnya. Hal ini tentunya menuntut adanya pengamanan tingkat lanjut sebagai sistem keamanan informasi dan penyimpanan data.

Berdiri pada tanggal 23 Agustus 1971 sebagai perusahaan yang bergerak di bidang ritel modern pertama di Indonesia. PT. Hero Supermarket Tbk. menjalankan komunikasi informasi dari sejumlah komputer client yang terhubung oleh *Local Area Network* yang terintegrasi dalam domain *dairy-farm.com.id*, hal ini dapat dilihat bahwa status komputer yang aktif yaitu adanya lalu lintas kegiatan komunikasi informasi dalam data yang dilakukan antar pengguna (*user*) perangkat komputer terhadap jaringan domain, sehingga memicu timbulnya resiko celah komponen keamanan (*security hole*), yaitu resiko (*risk*), ancaman (*threats*) dan kelemahan (*vulnerabilities*) terhadap data serta informasi yang tersimpan dalam partisi media penyimpanan jaringan, seperti pertukaran informasi yang bersifat privasi dan konfidensial dalam berbagi berkas (*peer to peer*) maupun melalui e-mail.

Umumnya PT. HERO Supermarket Tbk. melakukan tindakan keamanan data melalui mekanisme otentikasi serta *access control* dengan mengimplementasikan otoritas keamanan Windows model *Netlogon Service*. Model ini membatasi otoritas pengguna yang berhak mengakses sumber daya pada jaringan domain. Sedangkan dari sisi lain apabila secara dominan mayoritas pengguna diberikan otoritas izin akses, memicu timbulnya permasalahan data yang tersimpan terhadap tindakan interupsi, intersepsi, modifikasi dan fabrikasi oleh pengguna lain untuk unsur tertentu.

Salah satu dari cara pencegahan hal tersebut diperlukan adanya teknik proteksi menggunakan metode kriptografi dan steganografi. Kriptografi ini merupakan proses enkripsi *plaintext* dengan merotasikan atau mengkombinasi karakter dengan aturan tertentu, agar proses tersebut bisa berjalan dengan sukses, baik pengirim dan penerima harus memiliki akses kunci yang komplementer, sedangkan steganografi merupakan teknik menyembunyikan informasi di dalam media lain yang disebut sebagai sampul (*cover*), sehingga hanya pengirim dan penerima yang mengetahui isi informasi didalamnya.

2. Landasan Teori

2.1. Keamanan Sistem Informasi

Kontribusi keamanan data memprioritaskan antisipasi internal suatu informasi oleh tindakan anomali (*malicious user*) yang memicu ancaman timbulnya potensi celah komponen keamanan. Keamanan sistem informasi adalah proteksi atas informasi dan sistem informasi terhadap penggunaan akses yang tidak sah, non-repudiasi, gangguan, modifikasi, atau kerusakan, dengan memelihara kerahasiaan, integritas dan ketersediaan (*Committee on National Security Systems* (CNSS) *Instruction* No. 4009, 2010:37).

Mengatasi kendala penyalahgunaan informasi, keamanan data pada jaringan dalam ketentuannya disertai dengan enam konsep komponen keamanan, yaitu:

1. Otentikasi (*authentication*), merupakan proses verifikasi identitas atau klaim atribut lainnya oleh asumsi dari entitas (pengguna, proses, atau perangkat), dan/atau untuk memverifikasi sumber dan integritas data (Yulia Cherdantseva dan Jeremy Hilton, 2014:20).
2. Otoritas (*authorization*), otoritas terdiri dari pelengkap layanan keamanan yang digunakan setelah proses identifikasi dan otentikasi. Menetapkan hak istimewa dari masing-masing pengguna yang berbeda (Mário Marques da Silva, 2012:419).
3. Non-repudiasi (*non-repudiation*), merupakan jaminan bahwa pengirim informasi disediakan dengan bukti identitas pengirim, sehingga tidak dapat menolak setelah informasi diproses (*Committee on National Security Systems* (CNSS) *Instruction* No. 4009, 2010:50).
4. Kerahasiaan (*confidentiality*), merupakan komponen penting dari privasi dan mengacu pada kemampuan kita untuk melindungi data dari mereka yang tidak diizinkan untuk melihatnya.
5. Integritas (*integrity*), integritas mengacu pada kemampuan untuk mencegah data kita agar tidak berubah dalam ketidaksahan atau cara yang tidak diinginkan.
6. Ketersediaan (*availability*), ketersediaan mengacu pada kemampuan untuk mengakses data ketika kita membutuhkannya. Hilangnya ketersediaan dapat merujuk pada beragam kerusakan selama didalam rangkaian yang memberikan kita akses terhadap data (Jason Andress, 2011:4-6).

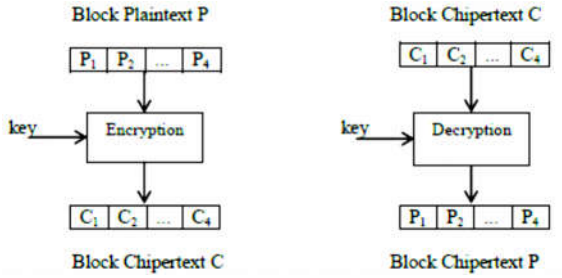
2.2. Model Serangan Sistem Keamanan

Serangan sistem komputer atau jaringan dapat diklasifikasikan sebagai berikut:

1. Gangguan (*interruption*), merupakan serangan terhadap ketersediaan. Aset/informasi mengalami rusaknya sistem atau menjadi tidak tersedia atau tidak dapat digunakan.
2. Intersepsi/pencegatan (*interception*), merupakan serangan terhadap kerahasiaan. Pihak yang tidak sah, seperti seseorang, program, atau komputer, memperoleh akses terhadap aset/informasi.
3. Modifikasi (*modification*), merupakan serangan terhadap integritas. Pihak yang tidak sah, tidak hanya dapat mengakses tetapi juga melakukan perubahan terhadap aset/informasi (Khaled M. Khan, 2013:11).
4. Fabrikasi (*fabrication*), merupakan serangan terhadap otentikasi. Musuh menyuntikan data palsu dan penyingkapan kepercayaan terhadap informasi yang disampaikannya (Bhavna Arora, 2013:6).

2.3. Konsep Enkripsi

Prinsip metode enkripsi mengimplementasikan proses perlindungan komunikasi terhadap informasi. Enkripsi dapat didefinisikan sebagai seni mengubah data dalam bentuk tersandi (*encoded*) dan dapat diterjemahkan (*decoded*) oleh penerima yang memiliki pengetahuan tentang dekripsi data *cipher*. Enkripsi dapat diaplikasikan pada teks, gambar, video untuk perlindungan data (V. V. Divya et al., 2012:286).



Gambar 2.1 Enkripsi-Dekripsi Blok *Cipher*

2.4. Konsep dan kriteria Kriptografi

Kriptografi adalah ilmu menulis rahasia, mengenkripsi pesan *plaintext* sebelum dikirim, dengan tujuan untuk menjaga kerahasiaan konten didalamnya jika dicegat oleh pihak yang salah. Pesan terenkripsi disebut sebagai *ciphertext*. *Ciphertext* akan didekripsi kembali menjadi plaintext ketika diterima, dalam rangka untuk memperoleh informasi aslinya (G. Michael Schneider dan Judith L. Gersting, 2010:342).

Berdasarkan fungsi dan jenis kunci yang digunakan, algoritma kriptografi dapat dibedakan menjadi dua jenis yaitu:

1. Kriptografi kunci klasik (*symmetric algorithm*), merupakan kunci enkripsi simetris atau secret key encryption, dalam prosesnya hanya menggunakan satu kunci untuk mengenkripsi dan mendekripsi data. Kunci tersebut didistribusikan sebelum melakukan transmisi antar entitas. Kunci memainkan peran penting. Jika menggunakan kunci yang lemah (*weak key*) dalam algoritma, maka semua orang dengan mudah dapat mendekripsi data. Kekuatan kunci simetris ini tergantung pada ukuran kunci yang digunakan (Yogesh Kumar et al., 2011:60).
2. Kriptografi kunci publik (*asymmetric algorithm*), merupakan kunci enkripsi asimetris kriptografi yang memerlukan dua proses pembentukan kunci, yaitu kunci publik sebagai proses enkripsi dan kunci privat (*private key*) sebagai proses dekripsi (Shreekalpa Sarkar dan Srijeet Chatterjee, 2017:9).

2.5. Algoritma Kriptografi Simetris AES Rijndael

Menurut N. Pramstaller et al. (2004), *Advanced Encryption Standard* (AES) memiliki ukuran tetap pada blok *plaintext* 128-bit (16-byte). 16-byte ini diwakili dalam

4x4 matriks dan AES beroperasi pada matriks byte. Selain itu, fitur penting lainnya dalam AES adalah jumlah putaran angka. Jumlah putaran bergantung pada panjang kunci. Ukuran kunci menentukan putaran angka seperti AES yang menggunakan 10 putaran untuk kunci 128-bit (Ako Muhammad Abdullah, 2017).

Berikut adalah *pseudocode* untuk algoritma AES-128:

Algoritma Start (Nama berkas, Kunci Rahasia)
1. Cari berkas
2. Masukkan 16-byte kunci rahasia

Algoritma Enkripsi:
1. Konversi ke susunan keadaan (*state array*)
2. Tambahkan baris kunci()
3. Sub byte()
4. Geser baris()
5. Campur kolom()
6. Ekspansi kunci

Kemudian untuk algoritma dekripsi:
1. Konversi ke susunan keadaan (*state array*)
2. Tambahkan baris kunci()
3. *Invers* sub bytes()
4. *Invers* geser baris()
5. *Invers* campur kolom()
6. Ekspansi kunci

2.6. Algoritma Kriptografi Asimetris RSA

Algoritma pembentukan kunci publik kriptografi asimetris dengan skema *Rivest-Shamir-Adleman* **(**RSA) menghasilkan dua bentuk kunci yang berbeda (kunci publik dan kunci privat), dan ada tiga langkah dalam pembuatan kedua kunci tersebut, yaitu:
1. Membuat dua bilangan prima sangat besar secara acak. Angka-angka ini nantinya disebut sebagai p dan q, kemudian angka ini dikalikan untuk mendapatkan nilai yang disebut sebagai n.
2. Buat angka acak sebagai nilai e, yang mana relatif prima dengan (p-1) x (q-1).
3. Hitung modulus invers dari e, yang disebut sebagai d. Kunci publik akan menjadi dua nilai sebagai n dan e, sedangkan kunci privat akan menjadi dua nilai sebagai n dan d (Al Sweigart, 2013:383).

Proses enkripsi RSA menunjukkan sebuah mekanisme blok *cipher*. Memisahkan dengan memasukan teks biner ke dalam 8-bit terpisah, mengonversikan 8-bit teks pertama ke dalam bentuk bilangan bulat (*integer*), kemudian sebuah kunci publik dari hasil *generate* kunci melakukan operasi enkripsi kepada *integer* tersebut, dengan menghasilkan $C = P^e \bmod n$.

Proses dekripsi RSA memisahkan masukan teks biner ke dalam 16-bit terpisah dan 16-bit teks dikonversikan ke dalam bentuk *integer*, kemudian sebuah kunci privat (*private key*) dari hasil *generate* kunci melakukan operasi dekripsi kepada *integer* tersebut, dengan menghasilkan $P = C^d \bmod n$ (Yogesh Kumar et al., 2011:62).

Berikut adalah *pseudocode* untuk algoritma dalam menghasilkan sepasang kunci RSA:
1. Pilih p dan q, dimana p dan q merupakan bilangan prima, $p \neq q$
2. Hitung n, dimana $n = p.q$
3. Hitung $\varphi(n) = (p-1)(q-1)$
4. Pilih nilai acak e dari: gcd ($\varphi(n)$, e) = 1, dimana $1 < e < \varphi(n)$
5. Hitung d, dimana $d.e \equiv 1 \pmod{\varphi(n)}$
6. Kunci publik sebagai (n, e)
7. Kunci privat sebagai (n, d)

Enkripsi dimana pengirim melakukan hal berikut:
1. Dapatkan kunci publik (n, e) penerima
2. Pesan *plaintext* sebagai bilangan bulat positif M, dimana $1 < M < n$
3. Menghitung *ciphertext* $C = M^e \bmod n$
4. Kirim *ciphertext* C ke penerima

Dekripsi dimana penerima melakukan hal berikut:
1. Gunakan kunci privat (n, d) dengan menghitung $M = C^d \bmod n$
2. Ekstrak *plaintext* dari pesan yang mewakili M

2.7. Metode Pengujian Primalitas Miller-Rabin

Algoritma uji primalitas Miller-Rabin dikemukakan oleh Michael Rabin mengenai algoritma *randomized polinimial-time* tahun 1980 untuk menguji suatu bilangan prima yang terkait erat dengan algoritma deterministik yang dipelajari oleh Gary Miller tahun 1976. Miller-Rabin adalah ekstensi yang sederhana dari *Fermat's Little Theorem* untuk menguji primalitas dengan probabilitas yang jauh lebih tinggi (Rajesh Pavuluru, 2015:2).

Berikut adalah *pseudocode* untuk algoritma Miller-Rabin dalam menguji suatu bilangan prima, dimana:

Input: $n > 3$, bilangan ganjil untuk diuji primalitasnya sebagai parameter x

Ouput: n merupakan bilangan komposit atau kemungkinan bilangan prima

write n-1 = $2^s.m$, dimana m bilangan ganjil faktorisasi pangkat 2 dari n-1

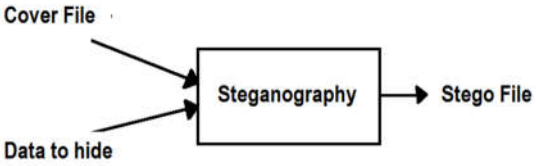
Loop:
1. **repeat** x dalam beberapa langkah:
2. Tentukan nilai acak a dari *range* [2, n-2]
3. $T_0 \leftarrow a^m \bmod n$
4. Jika $T_0 = 1$ atau $T_0 = n$-1 lanjutkan proses *looping* berikutnya
5. **for** r = 1, ... , s-1
6. $T_1 \leftarrow T_0^2 \bmod n$
7. jika $T_1 = 1$ **return** komposit, maka bilangan komposit
8. jika $T_1 = n$-1 **return** *loop* selanjutnya, maka bilangan prima
9. **return** komposit
10. **return** kemungkinan bilangan prima

2.8. Konsep Steganografi Penyisipan Data

Steganografi merupakan praktek pesan rahasia dengan tujuan informasi tersembunyi disisipkan (*embedding*) dalam sebuah objek sampul atau *cover*, sehingga tidak ada orang (kecuali pengirim dan penerima yang mengetahui tentang informasi tersembunyi) yang dapat melihat keberadaan pesan tersembunyi (Beant Singh dan Kul Bhusan Agnihotri, 2015:5).

Menurut Weiss (2009), ada tiga teknik umum yang digunakan untuk menyembunyikan informasi dalam objek sampul, yaitu:

1. *Injection* atau *insertion*, merupakan metode penyimpanan data yang ingin disembunyikan di area tertentu dalam sampul, sehingga diabaikan oleh usaha pemrosesan. Hal ini dilakukan guna mencegah modifikasi terhadap data-data yang relevan terhadap pengguna akhir dengan mengacuhkan keberadaan data tersembunyi dalam objek sampul yang digunakan.
2. *Substitution,* merupakan metode penukaran bagian dari deretan biner yang memiliki nilai paling terkecil atau tidak memiliki arti (*least significant bits*) dari informasi dan menetapkan konten berarti dari data asli dengan data baru pada sebuah cara yang menyebabkan sedikit distorsi.
3. *Generation*, tidak seperti dua metode sebelumnya, teknik ini tidak memerlukan keberadaan objek sampul. Ini menghasilkan sebuah objek sampul baru untuk menyembunyikan data. Kelemahan dari *insertion* dan *substitution* yaitu peretas dapat membandingkan berkas *stego* dengan salinan objek sampul yang sudah ada sebelumnya. Ketika menggunakan *Generation-based*, hasilnya adalah file asli, dan bebas terhadap percobaan komparasi (Ajmal K.A et al., 2012:36).



Gambar 2.2 Skenario Penerapan Steganografi

Memasuki pertengahan era 90-an hingga saat ini, steganografi berkembang menjadi tiga tipe yaitu:

1. *Pure Steganography*
2. *Secret key steganography*
3. *Public key steganography*

2.9. *Cover Generate Methods Steganography*

Cover Generate Methods steganografi memiliki persyaratan bahwa media yang dihasilkan secara prosedural pada umumnya belum ada sebelumnya atau reversibilitas, pembentukan objek baru tidak terbatas pada bidang penyembunyian informasi dan objek yang dihasilkan dapat didekonstruksi untuk memulihkan bit informasi (Philip Carson Ritchey, 2015:10).

2.10. Fungsi Hash Kriptografi SHA-256

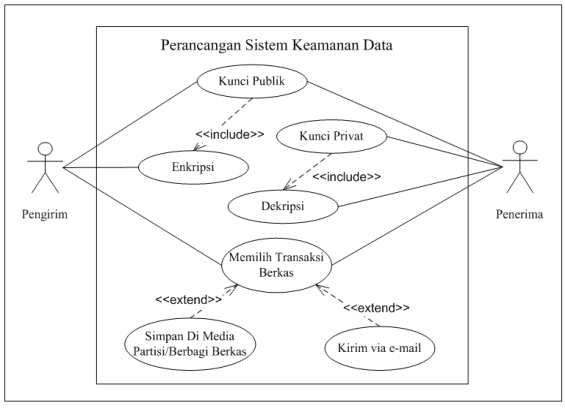
Fungsi hash digunakan untuk memberikan integritas data yang digunakan dalam kombinasi algoritma tanda tangan digital dan *Media Acces Controll* (MAC), serta menyediakan otentikasi. Penginterpretasian *hashing* menggunakan algoritma matematika terhadap data adalah untuk menghasilkan nilai numerik yang representatif dari data (Ignacio Algredo Badillo et al., 2011:544; *Committee on National Security Systems* (CNSS) *Instruction* No. 4009, 2010:33).

Secure Hashing Algorithm (SHA-256) merupakan salah satu turunan kriptografi dari fungsi hash SHA-2 yang memproses 512-bit *message block* dan menghasilkan nilai hash 256-bit.

Berikut ini adalah *pseudocode* untuk algoritma SHA-256:

1. **for** $i = 1$ pada N:
2. persiapan penjadwalan pesan W_t
3. inisialisasi delapan variabel kerja $A,\ldots, H$
4. **for** $t = 0$ sampai 63:
5. Hitung Temporal T_1
6. Hitung Temporal T_2
7. Hitung baru H, G, F
8. Hitung baru $E = D + T_1$
9. Hitung baru D, C, B
10. Hitung baru $A = T_1 + T_2$
11. Hitung i^{th} intermediasi nilai hash $H^{(i)}$
12. Hasilkan pesan *digest* dengan $H^{(N)}$

3. Rancangan Sistem Dan Aplikasi



Gambar 3.1 *Use Case Diagram* Rancangan Sistem Keamanan Data

Skenario dari masing-masing *Use Case* (Gambar 3.1) adalah analisis dua aktor (pengirim dan penerima) dalam perancangan sistem keamanan data. Dalam sistem ini penerima berkas menghasilkan sepasang kunci asimetris

yaitu kunci publik dan kunci privat, kunci publik diberikan kepada pengirim berkas, sedangkan kunci privat tetap disimpan kerahasiaannya oleh penerima. Proses enkripsi dilakukan pengirim setelah menerima kunci publik dari penerima. Berkas terenkripsi kemudian didistribusikan kepada pengirim dengan dua opsi transaksi pengiriman berkas yaitu melalui e-mail maupun disimpan dalam media penyimpanan atau berbagi berkas dalam jaringan.

Proses dekripsi dilakukan oleh penerima setelah adanya konfirmasi bahwa berkas terenkripsi tersebut sudah didistribusikan oleh pengirim. Penerima kemudian mengunduh berkas tersebut dari salah satu opsi pengiriman yang dipilih pengirim atau yang diminta penerima. Dekripsi berkas dilakukan penerima menggunakan kunci privatnya untuk memperoleh isi berkas aslinya.

3.1. Pengujian Miller-Rabin

Setiap bilangan prima dicocokkan nilainya melalui probabilitas pengujian Miller-Rabin dengan basis acak dan tes Lucas tunggal. Parameter Miller-Rabin yang dibutuhkan dalam pengujian ini dilakukan dengan langkah-langkah sebagai berikut:

1. Penerima membuat sepasang kunci publik/privat dengan menentukan dua bilangan prima p dan q. Misalkan diketahui, $p = 11$ dan $q = 7$.
   Hitung nilai n melalui persamaan, $n = p.q$
   Maka $n = 11.7 = 77$
   Pengujian Miller-Rabin:
   a. Temukan nilai dari $n\text{-}1 = 2^s.m$, dimana m harus bilangan ganjil.
      $77\text{-}1 = 2^s.m$
      $76 = 2^2.19$
      Maka nilai $s = 2$, dan nilai $m = 19$
   b. Pilih nilai a, dari $1 < a \leq n\text{-}1$
      $1 < a \leq 1072$, $a = 5$
   c. Hitung,
      $T_0 = a^m \bmod n$
      $T_0 = 5^{19} \bmod 77$
      $T_0 = 75$
      $T_1 = T_0^2 \bmod n$
      $T_1 = 75^2 \bmod 77$
      $T_1 = 4$

   Ketentuan lulusnya pengujian Miller-Rabin, jika $T_1\text{-}n = +1$ menghasilkan bilangan komposit dan jika $T_1\text{-}n = -1$ menghasilkan bilangan prima, maka 4-77 = -73. Kesimpulannya, n adalah bilangan komposit, walaupun 77 merupakan bilangan bulat ganjil, akan tetapi n bisa disebut sebagai bilangan semu prima.

2. Langkah selanjutnya, menentukan nilai e yang merupakan relatif prima dengan $(p\text{-}1).(q\text{-}1)$ sebagai berikut:
   $\varphi(n) = (p\text{-}1).(q\text{-}1)$
   $\varphi(n) = (11\text{-}1).(7\text{-}1)$
   $\varphi(n) = 10.6 = 60$
   $60 = 2^2.3.5$, melalui fungsi Euler's totient:
   Fungsi penghitungan perkalian,
   $\varphi(x^a\, y^b) = \varphi(x^a).\varphi(y^b)$
   $\varphi(60) = \varphi(2^2).\ \varphi(3).\ \varphi(5)$
   Fungsi dimana $\varphi(x^a) = (x\text{-}1)\, x^{a\text{-}1}$
   $\varphi(2^2) = (2\text{-}1).2^{2\text{-}1} = 1.2 = 2$
   $\varphi(3) = (3\text{-}1).3^0 = 2.1 = 2$
   $\varphi(5) = (5\text{-}1).5^0 = 4.1 = 4$
   Maka $\varphi(60) = 2.2.4 = 16$

Dari hasil tersebut dijelaskan bahwa, 16 merupakan jumlah bilangan bulat (*integer*) prima dari 7 hingga 59 dan bilangan ganjil yang tidak bisa dibagi oleh 3 dan 5.

Nilai e terdiri atas { 1, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 49, 53, 59 }. Misalkan ditentukan $e = 13$ sebagai salah satu bilangan relatif prima.

Menentukan nilai d melalui persamaan:

$e.d \bmod \varphi(n) = 1$

$d \equiv e^{-1} \bmod \varphi(n)$, disederhanakan;

$d \equiv \frac{1}{e} \bmod \varphi(n)$, disederhanakan;

$$d \equiv \frac{1 + \bmod \varphi(n)}{e}, \text{ disederhanakan;}$$

$$d \equiv \frac{1 + k.\varphi(n)}{e}$$

Melalui persamaan tersebut, nilai $k = \{0, 1, 2, 3, ...\}$, temukan nilai d yang menghasilkan bilangan bulat (*integer*).

$k = 0$, $d_0 \equiv \frac{1 + 0.60}{13} = \frac{1}{13} = 0.08$,
=> d_0 bukan bilangan bulat

$k = 1$, $d_1 \equiv \frac{1 + 1.60}{13} = \frac{61}{13} = 4.7$,
=> d_1 bukan bilangan bulat

$k = 2$, $d_2 \equiv \frac{1 + 2.60}{13} = \frac{121}{13} = 9.3$,
=> d_2 bukan bilangan bulat

$k = 3$, $d_3 \equiv \frac{1 + 3.60}{13} = \frac{181}{13} = 13.9$,
=> d_3 bukan bilangan bulat

$k = 4$, $d_4 \equiv \frac{1 + 4.60}{13} = \frac{241}{13} = 18.5$,
=> d_4 bukan bilangan bulat

$k = 5$, $d_5 \equiv \frac{1 + 5.60}{13} = \frac{301}{13} = 23.2$,
=> d_5 bukan bilangan bulat

$k = 6$, $d_6 \equiv \frac{1 + 6.60}{13} = \frac{361}{13} = 27.8$,
=> d_6 bukan bilangan bulat

$k = 7$, $d_7 \equiv \frac{1 + 7.60}{13} = \frac{421}{13} = 32.4$,
=> d_7 bukan bilangan bulat

$k = 8$, $d_8 \equiv \frac{1 + 8.60}{13} = \frac{481}{13} = 37$,
=> d_8 bilangan bulat

$d_8 = 37$, maka nilai d adalah 37
$e.d \bmod \varphi(n) = 1$, itu berarti $13.37 \bmod 60 = 1$
Hasilnya, kunci publik $(n, e) = (77, 13)$, dan kunci privat $(n, d) = (77, 37)$.

3. Pengirim melakukan proses enkripsi terhadap *plaintext* menggunakan kunci publik RSA melalui persamaan:

$$Ciphertext = M^e \bmod n$$

Misalkan *plaintext* (M) sebagai format ASCII dikonversikan ke dalam bentuk heksadesimal.
Plaintext = " aes "
$a = 01100001_2 = 97_{10} = 61_{16}$
$e = 01100101_2 = 101_{10} = 65_{16}$
$s = 01110011_2 = 115_{10} = 73_{16}$
Maka $M = 74\ 65\ 73$
$Ct = 61^{13} \bmod 77$
$= 161915287432152755657581 \bmod 77$
$= 40$
$Ce = 65^{13} \bmod 77$
$= 369720589101871337890625 \bmod 77$
$= 65$
$Ct = 73^{13} \bmod 1073$
$= 167184950739378888594103 3 \bmod 1073$
$= 24$
$C = 40\ 65\ 24$ atau dalam ASCII hasilnya,
Ciphertext = " @ e $ "

4. Proses dekripsi dilakukan penerima dengan memulihkan proses enkripsi menggunakan kunci privatnya dengan persamaan:

$$Plaintext = C^d \bmod n$$

Diketahui *ciphertext* (C) = 40 65 24
$P_1 = 40^{37} \bmod 77$
$= 61$
$P_2 = 65^{37} \bmod 77$
$= 65$
$P_1 = 24^{37} \bmod 77$
$= 73$
Hasil dekripsi *ciphertext* dikonversikan ke dalam format ASCII.
$61_{16} = (1.1) + (6.16) = 97_{10} = 01100001_2 = a$
$65_{16} = (5.1) + (6.16) = 101_{10} = 01100101_2 = e$
$73_{16} = (3.1) + (3.16) = 115_{10} = 01110011_2 = s$

4. Hasil Dan Pembahasan

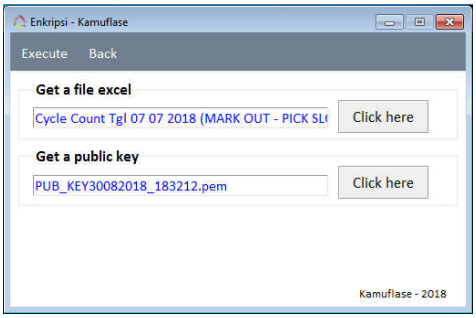
Proses dari analisis pengguna sistem enkripsi dan dekripsi terhadap data di jaringan LAN merupakan sistem keamanan pendukung dari mekanisme otentikasi *access control* serta otoritas keamanan Windows model *Netlogon Service* yang sudah ada di PT. Hero Supermarket Tbk.

Sebelum penerima A meminta berkas dari pengirim B, penerima A membuat dua buah kunci asimetris yaitu: kunci publik dan kunci privat.

Gambar 4.1 Tampilan Antarmuka Menu Pembentukan Kunci Asimetris

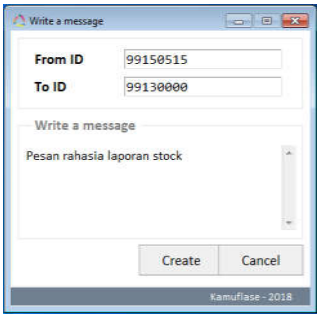
Kemudian, karena kunci privat bersifat rahasia dan kunci publik bersifat umum, maka penerima A menyimpan kunci privatnya, sedangkan kunci publik diberikan kepada pengirim B.

Pengirim B mengambil berkas konfidensial yang diminta oleh penerima A dan memasukkan kunci publik yang diterimanya dari penerima A untuk dienkripsi.

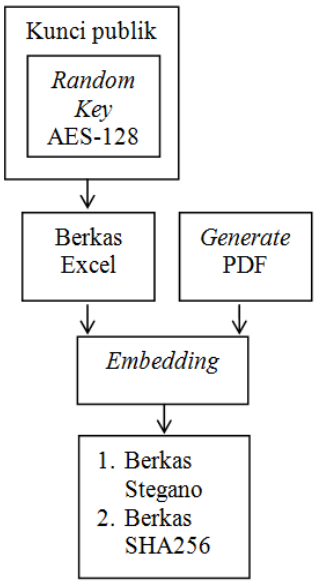


Gambar 4.2 Tampilan Antarmuka Menu Program Enkripsi

Pengirim B menyertakan ID pengirim dan ID penerima sebagai identitas, serta deskripsi berkas yang akan dikirim.



Gambar 4.3 *Pop-up* "*Write a message*" Menu Enkripsi



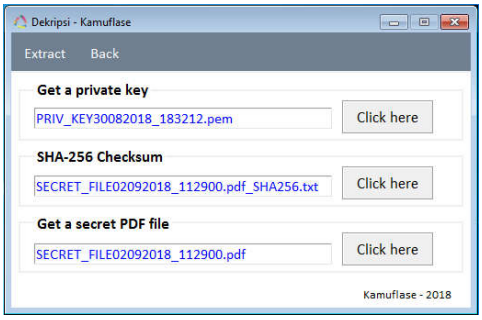
Gambar 4.4 Proses Kriptografi dan Steganografi

Rincian dari urutan proses enkripsi pada gambar 4.4 adalah sebagai berikut :

1. Kunci publik mengenkripsi kunci AES128 yang dihasilkan program secara acak.
2. Kunci AES-128 yang terenkripsi mengenkripsi berkas excel.
3. Program menghasilkan sebuah berkas berekstensi PDF.
4. Proses steganografi dimana berkas excel yang terenkripsi disisipkan kedalam berkas PDF tersebut, sehingga menghasilkan output berkas steganografi yang berekstensi PDF dan berkas teks SHA256 terenkripsi.

Setelah melakukan proses enkripsi, kemudian pengirim B mengirimkan kedua berkas tersebut kepada penerima A dalam jaringan LAN.

Penerima A mengunduh kedua berkas dari pengirim B melalui media penyimpanan partisi jaringan maupun e-mail. Untuk memperoleh atau memulihkan berkas aslinya, penerima A menggunakan kunci privatnya untuk melakukan proses dekripsi.



Gambar 4.5 Tampilan Antarmuka Menu Program Dekripsi

5. Kesimpulan Dan Saran

5.1. Kesimpulan

Kunci utama dari rangkaian metode proses enkripsi ini adalah pembentukan kunci asimetris RSA, peran kunci publik yang mengenkripsi kunci simetris AES-128 menghasilkan kunci simetris yang kompleks untuk mengenkripsi data. Proses steganografi melalui metode *Cover Generation*, menciptakan model penyisipan berkas sampul baru, sehingga dapat mencegah eksplorasi konfidensial dalam kondisi jaringan yang sama, serta verifikasi komparatif fungsi hash kriptografi SHA-256 menjamin keaslian data dari tindakan modifikasi data.

5.2. Saran-Saran

Berdasarkan hasil penelitian atas keamanan data yang dilakukan penulis pada PT Hero Supermarket Tbk., penulis berharap penerapan enkripsi dan dekripsi ini menjadi pertimbangan dalam kepentingan menjaga keamanan informasi dan data diseluruh PT. Hero Supermarket Tbk. yang terhubung dalam jaringan LAN baik dalam ruang lingkup penulis maupun yang tersebar diseluruh Indonesia.

Daftar Pustaka

Andress, Jason. *The Basics of Information Security*, Elsevier Inc, Waltham, 2011

Arora, Bhavna. *A Threat Model Approach for Classification of Network Layer Attacks in WSN*, International Journal of Computer Applications (0975-8887) Volume 63– No.9, February 2013

Badillo, Ignacio Algredo. et al. *Novel Hardware Architecture for implementing the inner loop of the SHA-2 Algorithms*, 2011 14th Euromicro Conference on Digital System Design, IEEE Computer Society, 2011

Carson Ritchey, Philip. *Synthetic Steganography: Methods For Generating And Detecting Covert Channels In Generated Media*, Purdue University Graduate School, 2015

Cherdantseva, Yulia., and Hilton, Jeremy. *Understanding Information Assurance and Security*, Technological Dimensions of IS Administrator, IGI Global, September 2014

Divya V. V., Sudha, S. K., dan Resmy, V.R. *Simple and Secure Image Encryption*, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 3, November 2012

K. A., Ajmal. et al. *Security using Colors, Figures and Images*, International Conference on Emerging Technology Trends on Advanced Engineering Research (ICETT'12) Proceedings published by International Journal of Computer Applications® (IJCA), Desember 2012

Khan, Khaled M. *Developing and Evaluating Security-Aware Software Systems*, United States of America by Information Science Reference, IGI Global, ISBN 978-1-4666-2483-2, Hersey PA, 2013

Kumar , Yogesh., Munjal, Rajiv., dan Sharma, Harsh. *Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures*, IJCSMS International Journal of Computer Science and Management Studies, Vol. 11, Issue 03 ISSN (Online): 2231-5268, Oktober 2011

Lamberger, Mario., dan Mendel, Florian. *Higher-Order Differential Attack on Reduced SHA-256*, Institute for Applied Information Processing and Communications Graz University of Technology, Inffeldgasse 16a, A-8010 Graz, Austria, Januari 2011

Muhammad Abdullah, Ako. *Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data*, Computer Science Department of Applied Mathematics & Computer Science Eastern Mediterranean University, Cyprus, 2017

National Information Assurance (IA) Glossary, Committee on National Security Systems, CNSS Instruction No. 4009, April 2010

Pavuluru, Rajesh. *Miller-Rabin,* Desember 2015

Sarkar, Shreekalpa., dan Chatterjee, Srijeet. *A Review Paper on ASCII Coded Cryptography for Artificial Neural Networking*, IJSRD - International Journal for Scientific Research & Development, Vol. 5, Issue 05, 2017

Schneider, G. Michael., dan Gersting, Judith. *Invitation to Computer Science*, Course Technology, Cengage Learning, Boston, 2010

Singh, Beant Kul., dan Agnihotri, Bhusan. *A Method to Hide Secret Information: Steganography*, ISSN (Online): 2347 - 2812, Volume-3, Issue-10, 2015

Sweigart, Al. *Hacking Secret Ciphers with Python*, a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License, ISBN 978-1482614374, 1st Edition, 2013