MEDIA TEKNOLOGI

Jurnal Ilmiah Teknologi

Vol. 7 - No. 02 : Periode Juli - Desember 2013 ISSN : 1693-8291

**Penentuan Fungsi Dan Status Jaringan Jalan Di Wilayah Kabupaten
Studi Kasus Kabupaten Cianjur**

Oleh : H. Ilham, Ir., MT.

**Studi Preferensi Moda Angkutan Laut Dalam Efisiensi Distribusi Barang Koridor
Jawa - Sumatera**

Oleh : Hendy Suryana, ST., MT.

**Kajian Pelaksanaan Perkerasan Kaku (Rigid Pavement) Menurut SNI PD T-14-
2003 Dan Spesifikasi Teknik Jalan**

**Studi Kasus : Proyek Peningkatan Jalan Sindangbarang - Agrabinta -
Tegalbuleud Di Kabupaten Cianjur**

Oleh : H. Yudi Sekaryadi, Ir., MT.

**Analisa Strategi Perusahaan Pakaian Jadi Menggunakan Metoda Whellen Hunger
Di Daerah Sukabumi - Bogor**

Oleh : Hendra Permana, Ir.

**Perwujudan Terminal Antar Kota Di Kabupaten Cianjur Melalui Pendekatan
Sistem Rekayasa**

Oleh : Hj. Elisabeth Yuniarti, Ir., MT.

**Enkripsi Dan Deskripsi Data Text Dengan Menggabungkan Metoda Enkripsi
(SHA-2, AES, Dan TWOFISH)**

Oleh : Tarmin Abdulghani, ST

**FAKULTAS TEKNIK
UNIVERSITAS SURYAKANCANA
CIANJUR**

Jl. Pasir Gede Raya Cianjur 43216
Telp. / Fax. (0263) 283578, 5019915
Website: <http://www.ft.unsur.ac.id>
Email: ftunsur@yahoo.com

FAKULTAS TEKNIK - UNSUR

PENGANTAR REDAKSI

TIM JURNAL

Penanggung jawab

Dekan

H. Ali Subhan ST., SH., MT., MKn.

Pimpinan redaksi

Pembantu Dekan I

Yogi Yogaswara, Ir., MT.

Sekretaris

Isma Masrofah, SST., SH.

Bendahara

Widy Setyawan, Ir.

Staff Ahli

Fuad Nasher, ST.

Agus Suheri, ST., MKom.

H. Ilham, Ir., MT.

H. Yudi Sekaryadi, Ir., MT.

Hendy Suryana, ST., MT.

Akhmad Sutoni, ST.

Redaksi Pelaksana

Hendra Permana, Ir.

H. Ilham, Ir., MT.

Hendy Suryana, ST., MT.

H. Yudi Sekaryadi, Ir., MT.

Tarmin Abdulghani, ST.

Hj. Elisabeth Yuniarti, Ir., MT.

Alamat redaksi

Fakultas Teknik Universitas Suryakencana Cianjur

Jl. Pasir Gede Raya (Belakang RSUD) Cianjur 43216

Telp. (0263) 5019915 e-mail : jurnal.ft.unsur@yahoo.com

DAFTAR ISI

Pengantar Redaksi	i
Sambutan Dekan	ii
Daftar Isi	iii
Penentuan Fungsi Dan Status Jaringan Jalan Di Wilayah Kabupaten Cianjur	1
Oleh : H. Ilham, Ir., MT.	
Studi Preferensi Moda Angkutan Laut Dalam Efisiensi Distribusi Barang Koridor Jawa - Sumatera	15
Oleh : Hendy Suryana, ST., MT.	
Kajian Pelaksanaan Perkerasan Kaku (<i>Rigid Pavement</i>) Menurut SNI PD T-14-2003 Dan Spesifikasi Teknik Jalan Studi Kasus Proyek Peningkatan Jalan Sindangbarang – Agrabinta – Tegalbuleud Di Kabupaten Cianjur	26
Oleh : H. Yudi Sekaryadi, Ir., MT.	
Analisa Strategi Perusahaan Pakaian Jadi Menggunakan Metoda <i>Whellen Hunger</i> Di Daerah Sukabumi – Bogor	34
Oleh : Hendra Permana, Ir.	
Perwujudan Terminal Antar Kota Di Kabupaten Cianjur Melalui Pendekatan Sistem Rekayasa	45
Oleh : Hj. Elisabeth Yuniarti, Ir., MT.	
Enkripsi Dan Deskripsi Data Text Dengan Menggabungkan Metoda Enkripsi (SHA-2, AES, Dan TWOFISH)	55
Oleh : Tarmin Abdulghani, ST.	

Enkripsi dan Deskripsi data text dengan menggabungkan metoda Enkripsi (SHA-2, AES, dan TWOFISH)

Oleh : **Tarmin Abdulghani, ST.**

Dosen Pengampu di Jurusan Teknik Informatika

Fakultas Teknik UNSUR

Abstrak

Secara umum enkripsi (encryption) dalam sebuah data agar menyulitkan membacaca Informasi yang sebenarnya. Dan enkripsi ini, semakin sulit di buka secara langsung di anggap semakin baik enkripsi. Dan dekripsi (decryption).

Dengan kunci yang sudah di tentukan atau sudah di sepakati diantara yang menyembunyikan data (cryptanalyst) dan yang mendekripsi data, akan menghasil data asal dengan data yang sama sebelum dilakukan data di enkripsi. Dalam implementasi penggabungan multi enkripsi ini menggunakan metoda enkripsi SHA-2, Enkripsi Advanced Encryption Standar (AES), Enkripsi TwoFish.

Kata Kunci : enkripsi, dekripsi, SHA-2, AES, TwoFish, CTI Text, encryptions, decryption

1. Pendahuluan

Adakalanya pengguna komputer ingin mengenkripsi tulisan atau teks penting agar tak semua orang dapat membaca atau mengerti arti tulisan tersebut. Bisa saja menggunakan password untuk melindungi agar tulisan atau teks penting tersebut tidak dapat dibuka oleh orang lain. Namun semakin majunya perkembangan software, sangat riskan untuk dijebol password yang telah digunakan untuk melindungi teks atau tulisan tersebut. Dalam dunia Kriptografi, menyandi data agar hanya bisa diterjemahkan oleh orang-orang yang mengerti tentu saja akan semakin sulit untuk dipecahkan bagi orang yang ingin membaca arti tulisan tersebut.

Dalam mengimplementasikan maka dengan pendekatan aplikasi CTI Text Encryption. Penggunaan secara umum, dengan cara mengacak atau menyandi tulisan yang telah dibuat menjadi huruf acak, simbol maupun angka yang sulit untuk dipahami. Untuk implementasi penggabungan metoda enkripsi dimana mempergunakan metoda Enkripsi SH2, Enkripsi TwoFish dan Enkripsi Aes yang terbungkus dalam aplikasi CTI Text Encryption.

2. Konsep Enkripsi

Untuk melakukan pengacakan data, sebenarnya banyak sekali metode enkripsi yang di pergunakan namun untuk enkripsi yang di pergunakan kali ini dengan metode SHA-2, AES, dan Twofish.

a. SHA-2

SHA-2 adalah satu set fungsi hash kriptografi (SHA-224, SHA-256, SHA-384, SHA-512) yang dirancang oleh National Security Agency (NSA) dan diterbitkan pada tahun 2001 oleh NIST sebagai US Federal Information Processing Standard. Sebuah fungsi hash adalah suatu algoritma yang mengubah (hash) set sewenang-wenang elemen data ke dalam nilai tunggal panjang tetap (hash). Nilai hash dihitung kemudian dapat digunakan untuk memverifikasi integritas dari salinan data asli tanpa memberikan sarana untuk memperoleh sumber (ireversibel). Sebuah nilai hash sehingga dapat bebas didistribusikan atau disimpan karena hanya digunakan untuk tujuan perbandingan. SHA singkatan dari Secure Hash Algorithm. SHA-2 mencakup sejumlah besar perubahan dari pendahulunya, SHA-1. SHA-2 terdiri dari satu set empat fungsi hash dengan mencerna yang 224, 256, 384 atau 512 bit.

Keamanan yang disediakan oleh algoritma hashing sepenuhnya tergantung pada kemampuannya untuk menghasilkan nilai unik untuk setiap set data tertentu. Ketika fungsi hash menghasilkan nilai hash yang sama untuk dua set data yang berbeda maka tabrakan dikatakan terjadi. Tabrakan meningkatkan kemungkinan bahwa seorang penyerang mungkin dapat komputasi kerajinan set data yang menyediakan akses ke informasi yang dijamin oleh nilai-nilai hashed kode lulus atau untuk mengubah data file komputer dalam mode yang tidak akan mengubah nilai hash yang dihasilkan dan dengan demikian akan melarikan diri deteksi. Sebuah fungsi hash yang kuat adalah salah satu yang tahan terhadap serangan komputasi tersebut. Sebuah fungsi hash yang lemah adalah salah satu di mana pendekatan komputasi untuk menghasilkan tabrakan diyakini menjadi mungkin. Sebuah fungsi hash rusak adalah salah

satu di mana metode komputasi untuk menghasilkan tabrakan diketahui ada.

Pada tahun 2005, kelemahan keamanan diidentifikasi dalam SHA-1, yaitu bahwa kelemahan matematika mungkin ada, menunjukkan bahwa fungsi hash kuat akan diinginkan [2] Meskipun SHA-2 memiliki beberapa kesamaan dengan algoritma SHA-1. Serangan ini memiliki belum berhasil diperpanjang untuk SHA-2.

b. Twofish

Dalam kriptografi, Twofish merupakan blok cipher kunci simetris dengan ukuran blok 128 bit dan ukuran kunci hingga 256 bit. Itu adalah salah satu dari lima finalis dari kontes Advanced Encryption Standard, tapi itu tidak dipilih untuk standardisasi. Twofish berkaitan dengan blok cipher sebelumnya Blowfish.

Gambar 1 Twofish Encryption

Fitur khas Twofish ini adalah penggunaan pra-dihitung key-dependent S-kotak, dan jadwal kunci relatif kompleks. Satu setengah dari kunci n-bit digunakan sebagai kunci enkripsi yang sebenarnya dan setengah lainnya dari kunci n-bit digunakan untuk memodifikasi algoritma

enkripsi (key-dependent S-kotak). Twofish meminjam beberapa elemen dari desain lain, misalnya, pseudo-Hadamard transform (PHT) dari keluarga AMAN dari cipher. Twofish menggunakan struktur Feistel sama dengan DES.

Pada platform perangkat lunak yang paling Twofish adalah sedikit lebih lambat dari Rijndael (algoritma dipilih untuk Advanced Encryption Standard) untuk 128-bit kunci, tetapi agak lebih cepat untuk 256-bit kunci. [3]

Twofish dirancang oleh Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, dan Niels Ferguson, yang "Twofish tim diperpanjang" yang bertemu untuk melakukan cryptanalysis lanjut dari pendatang AES Twofish dan kontes termasuk keberuntungan Stefan, Tadayoshi Kohno, dan Mike tinggal.

Cipher Twofish belum dipatenkan dan implementasi referensi telah ditempatkan dalam domain publik. Akibatnya, algoritma Twofish gratis bagi siapa saja untuk menggunakan tanpa batasan apapun. Ini adalah salah satu dari beberapa cipher termasuk dalam standar OpenPGP (RFC 4880). Namun, Twofish telah melihat penggunaan lebih luas daripada Blowfish, yang telah tersedia lagi.

c. Advanced Encryption Standar (AES)

SHA-2 adalah satu set fungsi hash kriptografi (SHA-224, SHA-256, SHA-384, SHA-512) yang dirancang oleh National Security Agency (NSA) dan diterbitkan pada tahun 2001 oleh NIST sebagai US Federal Information Processing Standard. Sebuah fungsi hash adalah suatu algoritma yang mengubah (hash) set sewenang-wenang elemen data ke dalam nilai tunggal panjang tetap (hash). Nilai hash dihitung

kemudian dapat digunakan untuk memverifikasi integritas dari salinan data asli tanpa memberikan sarana untuk memperoleh sumber (ireversibel). Sebuah nilai hash sehingga dapat bebas didistribusikan atau disimpan karena hanya digunakan untuk tujuan perbandingan. SHA singkatan dari Secure Hash Algorithm. SHA-2 mencakup sejumlah besar perubahan dari pendahulunya, SHA-1. SHA-2 terdiri dari satu set empat fungsi hash dengan mencerna yang 224, 256, 384 atau 512 bit.

Gambar 2 AES Encryption

Keamanan yang disediakan oleh algoritma hashing sepenuhnya tergantung pada kemampuannya untuk menghasilkan nilai unik untuk setiap set data tertentu. Ketika fungsi hash menghasilkan nilai hash yang sama untuk dua set data yang berbeda maka tabrakan dikatakan terjadi. Tabrakan meningkatkan kemungkinan bahwa seorang penyerang mungkin dapat komputasi kerajinan set data yang menyediakan akses ke informasi yang dijamin oleh nilai-nilai hashed kode lulus atau untuk mengubah data file komputer dalam mode yang tidak akan mengubah nilai hash yang dihasilkan dan dengan demikian akan melarikan diri deteksi. Sebuah fungsi hash yang kuat adalah salah satu yang tahan terhadap serangan komputasi tersebut. Sebuah fungsi hash yang lemah adalah salah satu di mana pendekatan komputasi untuk menghasilkan tabrakan diyakini menjadi mungkin. Sebuah fungsi hash rusak adalah salah

satu di mana metode komputasi untuk menghasilkan tabrakan diketahui ada.

Pada tahun 2005, kelemahan keamanan diidentifikasi dalam SHA-1, yaitu bahwa kelemahan matematika mungkin ada, menunjukkan bahwa fungsi hash kuat akan diinginkan [2] Meskipun SHA-2 memiliki beberapa kesamaan dengan algoritma SHA-1, Serangan ini memiliki belum berhasil diperpanjang untuk SHA-2.

3. CTI Text Encryption

Agar menghasilkan pengkodean / enkripsi data yang lebih rumit maka ke-tiga jenis Enkripsi di satukan dalam satu aplikasi dengan harapan pengacakan data semakin baik dan tidak mudah di ketahui plaintext nya.

Untuk mempermudah pengguna menyimpan text yang akan di enkripsi dan didekripsi maka ada komponen dengan tipe text untuk menyimpan objek datanya.

Gambar 3 CTI Text Encryption

Sedangkan untuk enkripsi data dengan metode SHA-2, AES, dan Twofish dengan cara dianggap sebagai password. Sehingga saat di penggunaan untuk enkripsi dan dekripsi, kunci itu harus sama. Apabila salah satu kunci berbeda walaupun hanya satu karakter tidak berfungsi

atau tidak bisa mengembalikan / dekripsi data yang sudah dilakukan di enkripsi.

a. Enkripsi

Karena metoda enkripsi yang berbeda dan akan saat melakukan enkripsi, saat melakukan enkripsi pun di buat fungsi tersendiri dari ketiga enkripsi tersebut dengan di lakukan inisiasi.

```
public bool InitializeEngine()
{
    if (_algorithm ==
        AlgorithmType.None)
        return false;

    if (_algorithm ==
        AlgorithmType.BlowFish ||
        _algorithm == AlgorithmType.DES ||
        _algorithm == AlgorithmType.RC2 ||
        _algorithm ==
        AlgorithmType.Rijndael ||
        _algorithm ==
        AlgorithmType.TripleDES ||
        _algorithm ==
        AlgorithmType.Twofish)
    {
        _isSymmetric = true;
        _isHash = false;
    }
    else
    if (_algorithm ==
        AlgorithmType.MD5 ||
        _algorithm == AlgorithmType.SHA
        ||
        _algorithm ==
        AlgorithmType.SHA256 ||
        _algorithm ==
        AlgorithmType.SHA384 ||
        _algorithm ==
        AlgorithmType.SHA512)
    {
        _isSymmetric = false;
        _isHash = true;
    }
}
```

b. Dekripsi

Dalam melakukan dekripsi data yang sudah teracak datanya, maka dalam melakukan dekripsi data, dalam program ini dibuat suatu fungsi yang bisa membaca data dalam objek menyimpan data dan memberikan ke fungsi dekripsi untuk melakukan mengembalikan data yang sudah di enkripsi.

```
public int Decrypt
(byte[] dataIn,
 byte[] dataOut,
 int nPosIn,
 int nPosOut,
 int nCount)
{
    // count in blocks
    nCount >>= 3;
    for (int nl = 0; nl < nCount; nl++)
    {
        // load the bytes into two 32bit words,
        which together represent the block

        uint unHi = (((uint)dataIn[nPosIn]) << 24) |
            (((uint)dataIn[nPosIn + 1]) << 16) |
            (((uint)dataIn[nPosIn + 2]) << 8) |
            dataIn[nPosIn + 3];

        uint unLo = (((uint)dataIn[nPosIn + 4]) <<
            24) |
            (((uint)dataIn[nPosIn + 5]) << 16) |
            (((uint)dataIn[nPosIn + 6]) << 8) |
            dataIn[nPosIn + 7];

        // encrypt that construct
        Decrypt(ref unHi, ref unLo);
        // write the encrypted block back

        dataOut[nPosOut] = (byte)(unHi >> 24);
        dataOut[nPosOut + 1] = (byte)(unHi >>
            16);
        dataOut[nPosOut + 2] = (byte)(unHi >> 8);
        dataOut[nPosOut + 3] = (byte)unHi;
```

```
dataOut[nPosOut + 4] = (byte)(unLo >>
    24);
dataOut[nPosOut + 5] = (byte)(unLo >>
    16);
dataOut[nPosOut + 6] = (byte)(unLo >> 8);
dataOut[nPosOut + 7] = (byte)unLo;
```

```
nPosIn += 8;
nPosOut += 8;
    }
    return nCount;
}
}
```

4. Penutup

Dengan melakukan enkripsi data dengan cara kombinasi metode enkripsi sangat membuat tingkat kesulitan yang lebih banyak, yang berbeda dengan melakukan kunci untuk satu metode enkripsi.

a. Kesimpulan

CTI Text Encryption cukup baik dalam membuat aplikasi ini, karena cukup memadai baik secara enkripsi data dengan algoritma sehingga tidak ada kesalahan dalam melakukan enkripsi maupun dekripsi. Di lengkapi dengan fitur menghilangkan / hidden text yang di enkripsi hal ini membantu pencurian data secara visual karena terlihat secara sengaja ataupun tidak di sengaja.

b. Saran

Karena aplikasi cukup lumayan baik dan sifatnya terbuka (*open source*) yang di ambil di *codeplex* aplikasi yang sifatnya terbuka untuk aplikasi berbasis windows, aplikasi bisa di kembangkan dengan metode enkripsi yang berbeda atau di kembangkan dalam bentuk berbasis web base.

5. Daftar Pustaka

- [a] Advanced Encryption Standard,
<http://en.wikipedia.org/wiki/Rijndael>, di
akses 27 Maret 2013, jam 16.00
- [b] CTI Text Encryption,
<http://ctiencrypt.codeplex.com/wikipage?title=History%20%28Change%20Log%29> di
akses 27 Maret 2013, jam 16.00
- [c] Enkripsi Teks Penting dengan CTI Text
Encryption <http://pri-oftware.blogspot.com/2013/01/enkripsi-teks-penting-dengan-cti-text.html> di akses
27 Maret 2013, jam 16.00
- [d] SHA-2 Encryption,
<http://en.wikipedia.org/wiki/SHA-2>, di akses
27 Maret 2013, jam 16.00
- [e] Twofish,
<http://en.wikipedia.org/wiki/Twofish>, di
akses 27 Maret 2013, jam 16.00
- [f] XCrypt - encryption & decryption class
wrapper,
<http://www.codeproject.com/Articles/4797/XCrypt-encryption-decryption-class-wrapper>,
di akses 27 Maret 2013, jam 16.00