**UJIAN AKHIR SEMESTER KRIPTOGRAFI**

**PENGERTIAN KRIPTOGRAFI DAN ALGORITMA KRIPTOGRAFI**

****

Disusun oleh :

Made Rio Efendi NIM : 2015-81-111

FAKULTAS ILMU KOMPUTER

TEKNIK INFORMATIKA

UNIVERSITAS ESA UNGGUL

2018

**BAB I**

**PENDAHULUAN**

1. Latar Belakang

Perkembangan teknologi informasi dewasa ini telah berpengaruh pada hampir semua aspek kehidupan manusia. Saat ini internet seolah-olah menghiraukan batas-batas geografis suatu negara, bukan hal yang sulit bagi seseorang untuk berkomunikasi jarak jauh, mengirimkan data, mencari informasi dan sebagainya. Semua hal tersebut dapat dilakukan dengan internet secara cepat, efisien dan relatif murah. Namun di sisi lain, ternyata internet merupakan jalur yang tidak terlalu aman karena merupakan media komunikasi umum yang dapat digunakan secara bebas oleh siapapun sehingga sangat rawan penyadapan informasi oleh pihak-pihak yang tidak absah. Oleh karena itu masalah keamanan, kenyamanan dan otorisasi dalam berinternet menjadi masalah krusial yang sampai hari ini masih menjadi perbincangan hangat. Banyak pertanyaan yang muncul, Apakah transaksi via internet bisa terjamin keamanannya? Apakah data yang dikirimkan via internet sampai pada tujuan yang tepat tanpa diketahui pihak yang tidak diinginkan? Bagaimana mencegah penyadapan informasi? Bagaimana mengantisipasi seseorang mencuri data?

Banyak metode yang dapat dilakukan untuk menjawab pertanyaan-pertanyaan diatas. Salah satunya dengan metode penyandian atau yang lebih dikenal dengan metode kriptografi. Banyak sekali metode penyandian atau metode kriptografi yang dikembangkan oleh pakar-pakar kriptografi hingga saat ini. Hal ini dilakukan dikarenakan penyadap dan pencuri informasi atau yang lebih dikenal dengan sebutan cracker semakin handal dalam mempenetrasi suatu sistem untuk menggali berbagai macam informasi. Oleh karena itu dalam rangka melawan tindakan keji tersebut, pakar-pakar kriptografi terus mengembangkan metode ini secara berkesinambungan. Banyak sekali metode penyandian yang telah diciptakan oleh pakar-pakar kriptografi dunia, sebut saja Algoritma DES, Algoritma 3DES, Algoritma IDEA, Algoritma Blowfish, Algotitma RSA, Algoritma MD4, Algotitma MD5, Algoritma SHA-1, Algoritma McEliecce dan sebagainya. Algoritma-algoritma diatas telah diuji kemampuannya oleh pakar-pakar kriptografi, namun tidak semua metode kriptografi diatas bertahan dari serangan para penyadap informasi atau dalam istilah kriptografi sering disebut dengan cryptanalist. Sebut saja algoritma DES, sempat bertahan cukup lama yakni selama 20 tahun akhirnya harus rela di-crack hanya dalam tempo 3,5jam dengan biaya 1 juta US Dollar pada tahun 1993. Namun para pakar kriptografi tidak berdiam diri, karena setelah kejadian itu para pakar kriptograhi mengembangkan varian baru dari algoritma DES seperti Algoritma 3DES, Algoritma RDES dan sebagainya, Makalah ini akan membahas DES, AES, MD5, 3DES, DIGITAL SIGNATURE.

1. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah di uraikan, dapat dirumuskan masalah-

masalah yang akan dibahas pada penulisan kali ini. Masalah yang dimaksud adalah

sebagai berikut :

1. Bagaimana sejarah kriptografi.
2. Apakah tujuan kriptografi.
3. Apakah pengertian fungsi hash.
4. Bagaimana sejarah MD5.
5. Bagaimana cara kerja MD5.
6. Bagaimana Aplikasi Algoritma MD5 Untuk Integritas Data.
7. Bagaimana sejarah AES.
8. Bagaimana cara kerja AES.
9. Bagaimana sejarah DES.
10. Bagaimana cara kerja DES.
11. Bagaimana cara kerja RSA.
12. Apa pengertian Digital Signature.
13. Bagaimana cara kerja Digital Signature.
14. Tujuan Pembuatan Makalah

Pembuatan makalah bermaksud untuk menganalisa proses penyandian pesan dan proses membuka kembali penyandian dalam kriptografi, Diharapkan dengan adanya makalah ini para pengguna jalur internet lebih ekstra hati-hati dalam penggunaan jalur publik ini..

**BAB II**

**LANDASAN TEORI**

1. Sejarah Kriptografi

Dalam The CodeBreaker yang ditulis oleh Kahn, terlihat bahwa kriptografi mempunyai sejarah yang panjang. Kriptografi sudah digunakan oleh bangsa Mesir Kuno sekitar 4000 tahun sampai abad 20 dimana kriptografi berperanan penting di dalam perang dunia pertama dan kedua. Juga diungkap bahwa latar belakang sejarah yang panjang itu sangat menentukan perkembangan ilmu kriptografi itu sendiri baik dari segi teoretik maupun aplikasinya.

Awalnya kriptografi sangat dominan digunakan dalam bidang-bidang yang berhubungan dengan militer, layanan diplomatik, dan pemerintahan secara umum. Dalam hal ini kriptografi digunakan sebagai suatu alat untuk melindungi strategi dan rahasia negara. Perkembangan sistem komunikasi dan komputer pada tahun 1960 an membawa kriptografi memasuki sektor swasta sebagai alat untuk melindungi informasi dalam bentuk dijitel dan untuk memberikan layanan keamanan.

Hasil kerja Feistel di IBM pada awal tahun 1970 an dan puncaknya ada tahun 1977, DES (Data Ecryption Standart) merupakan karya kriptografi yang paling terkenal di dalam sejarah. Karya ini menjadi alat keamanan komersial elektronik di banyak institusi keuangan di seluruh dunia hingga pertengahan tahun 1990-an. DES secara definitif terbukti tak-aman sejak Juli 1998. Walaupun demikian DES telah melandasi prinsip-prinsip sandi simetrik modern yang dewasa ini muncul produk-produk penggantinya seperti: AES (Advanced Ecryption Standart), Blowfish, 3DES, RC5, dan lain sebagainya.

Yang cukup signifikan selanjutnya adalah pada tahun 1976 ketika Diffie dan Hellman mempublikasikan suatu artikel dengan judul New Directions in Cryptography. Artikel ini memperkenalkan konsep revolusioner tentang kriptograpy kunci-publik (public-key cryptography) dan juga memberikan suatu metode baru untuk perubahan kunci dimana keamanan didasarkan pada pemecahan problem logaritme diskret. Walaupun penulis pada saat itu mengungkapkan hanya segi teoretiknya tanpa bentuk praktisnya, namun karya ini telah memberikan cakrawala baru bagi para ilmuwan kriptografi. Ini terbukti pada tahun 1978, Rivest, Shamir, dan Adleman menemukan bentuk praktis yang pertama untuk skema enkripsi dan penandaan kunci publik yang sekarang dikenal dengan skema RSA. Skema ini didasarkan pada problem matematika yang sulit lainnya, yaitu pemecahan masalah faktorisasi intejer besar. Bentuk praktis skema kunci-publik lainya ditemukan oleh ElGamal pada tahun 1985. Sebagaimana karya Di¢ e dan Hellman, skema ini juga didasarkan pada pemecahan problem logaritme diskret.

Sumbangan yang paling signifikan yang diberikan olen kriptografi kunci-publik adalah penandaan dijitel (digital signature). Pada tahun1991 standar internasional pertama untuk penandaan dijitel diadopsi dari ISO/IEC 9796. Standar internasional penandaan dijitel ini didasarkan ada skema kunci-publik RSA. Pada tahun 1994 pemerintah Amerika Serikat mengadopsi standar penandaan dijitel yang mekanismenya didasarkan pada skema kunci-publik ElGamal.

1. Tujuan Kriptografi

Secara umum tujuan ilmu kriptografi diciptakan adalah untuk merahasiakan suatu pesan agar tidak bisa dibaca oleh pihak yang tidak absah. berpedoman dari tujuan umum tersebut, terdapat empat tujuan mendasar ilmu kriptografi digunakan dalam bidang keamanan informasi yaitu :

1. Kerahasiaan

adalah layanan yang digunakan untuk menjaga isi informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.

1. Integritas data

adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.

1. Autentikasi

adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.

1. Non-repudiasi

adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

**BAB III**

**PEMBAHASAN**

1. Pengertian Fungsi Hash

Hash adalah suatu teknik "klasik" dalam Ilmu Komputer yang banyak digunakan dalam praktek secara mendalam. Hash merupakan suatu metode yang secara langsung mengakses record-record dalam suatu tabel dengan melakukan transformasi aritmatik pada key yang menjadi alamat dalam tabel tersebut. Key merupakan suatu input dari pemakai di mana pada umumnya berupa nilai atau string karakter.

Pelacakan dengan menggunakan Hash terdiri dari dua langkah utama, yaitu:

* Menghitung Fungsi Hash. Fungsi Hash adalah suatu fungsi yang mengubah key menjadi alamat dalam tabel. Fungsi Hash memetakan sebuah key ke suatu alamat dalam tabel. Idealnya, key-key yang berbeda seharusnya dipetakan ke alamat-alamat yang berbeda juga. Pada kenyataannya, tidak ada fungsi Hash yang sempurna. Kemungkinan besar yang terjadi adalah dua atau lebih key yang berbeda dipetakan ke alamat yang sama dalam tabel. Peristiwa ini disebut dengan collision (tabrakan). Karena itulah diperlukan langkah berikutnya, yaitu collision resolution (pemecahan tabrakan).
* Collision Resolution. Collision resolution merupakan proses untuk menangani kejadian dua atau lebih key di-hash ke alamat yang sama. Cara yang dilakukan jika terjadi collision adalah mencari lokasi yang kosong dalam tabel Hash secara terurut. Cara lainnya adalah dengan menggunakan fungsi Hash yang lain untuk mencari lokasi kosong tersebut.

1. Sejarah MD5

MD5 adalah salah satu dari serangkaian algortima(merupakan salah satu fungsi Hash) message digest yang didesain oleh Profesor Ronald Rivest dari MIT (Rivest, 1994). Saat kerja analitik menunjukkan bahwa pendahulu MD5, yaitu MD4 mulai tidak aman, MD5 kemudian didesain pada tahun 1991 sebagai pengganti dari MD4 (kelemahan MD4 ditemukan oleh Hans Dobbertin).

Pada tahun 1993, den Boer dan Bosselaers memberikan awal, bahkan terbatas, hasil dari penemuan pseudo-collision dari fungsi kompresi MD5. Dua vektor inisialisasi berbeda I dan J dengan beda 4-bit diantara keduanya.

MD5compress(I,X) = MD5compress(J,X)

Pada tahun 1996 Dobbertin mengumumkan sebuah kerusakan pada fungsi kompresi MD5. Dikarenakan hal ini bukanlah serangan terhadap fungsi hash MD5 sepenuhnya, hal ini menyebabkan para pengguna kriptografi menganjurkan pengganti seperti WHIRLPOOL, SHA-1 atau RIPEMD-160.

Ukuran dari hash — 128-bit — cukup kecil untuk terjadinya serangan brute force.. MD5CRK adalah proyek distribusi mulai Maret 2004 dengan tujuan untuk menunjukkan kelemahan dari MD5 dengan menemukan kerusakan kompresi menggunakan brute force attack. Bagaimanapun juga, MD5CRK berhenti pada tanggal 17 Agustus 2004, saat [[kerusakan hash]] pada MD5 diumumkan oleh Xiaoyun Wang, Dengguo Feng, Xuejia Lai dan Hongbo Yu. Serangan analitik mereka dikabarkan hanya memerlukan satu jam dengan menggunakan IBM P690 cluster.

Pada tanggal 1 Maret 2005, Arjen Lenstra, Xiaoyun Wang, and Benne de Weger mendemontrasikan kunstruksi dari dua buah sertifikat X.509 dengan public key yang berbeda dan hash MD5 yang sama, hasil dari demontrasi menunjukkan adanya kerusakan. Konstruksi tersebut melibatkan private key untuk kedua public key tersebut. Dan beberapa hari setelahnya, Vlastimil Klima menjabarkan dan mengembangkan algortima, mampu membuat kerusakan Md5 dalam beberapa jam dengan menggunakan sebuah komputer notebook. Hal ini menyebabkan MD5 tidak bebas dari kerusakan.

Dikarenakan MD5 hanya menggunakan satu langkah pada data, jika dua buah awalan dengan hash yang sama dapat dibangun, sebuah akhiran yang umum dapat ditambahkan pada keduanya untuk membuat kerusakan lebih masuk akal. Dan dikarenakan teknik penemuan kerusakan mengijinkan pendahuluan kondisi hash menjadi arbitari tertentu, sebuah kerusakan dapat ditemukan dengan awalan apapun. Proses tersebut memerlukan pembangkitan dua buah file perusak sebagai file template, dengan menggunakan blok 128-byte dari tatanan data pada 64-byte batasan, file-file tersebut dapat mengubah dengan bebas dengan menggunakan algoritma penemuan kerusakan.

Saat ini dapat diketahui, dengan beberapa jam kerja, bagaimana proses pembangkitan kerusakan MD5. Yaitu dengan membangkitkan dua byte string dengan hash yang sama. Dikarenakan terdapat bilangan yang terbatas pada keluaran MD5 (2128), tetapi terdapat bilangan yang tak terbatas sebagai masukannya, hal ini harus dipahami sebelum kerusakan dapat ditimbulkan, tapi hal ini telah diyakini benar bahwa menemukannya adalah hal yang sulit.

Sebagai hasilnya bahwa hash MD5 dari informasi tertentu tidak dapat lagi mengenalinya secara berbeda. Jika ditunjukkan informasi dari sebuah public key, hash MD5 tidak mengenalinya secara berbeda jika terdapat public key selanjutnya yang mempunyai hash MD5 yang sama.

Bagaimanapun juga, penyerangan tersebut memerlukan kemampuan untuk memilih kedua pesan kerusakan. Kedua pesan tersebut tidak dengan mudah untuk memberikan serangan preimage, menemukan pesan dengan hash MD5 yang sudah ditentukan, ataupun serangan preimage kedua, menemukan pesan dengan hash MD5 yang sama sebagai pesan yang diinginkan.

Hash MD5 lama, yang dibuat sebelum serangan-serangan tersebut diungkap, masih dinilai aman untuk saat ini. Khususnya pada digital signature lama masih dianggap layak pakai. Seorang user boleh saja tidak ingin membangkitkan atau mempercayai signature baru menggunakan MD5 jika masih ada kemungkinan kecil pada teks (kerusakan dilakukan dengan melibatkan pelompatan beberapa bit pada bagian 128-byte pada masukan hash) akan memberikan perubahan yang berarti. Penjaminan ini berdasar pada posisi saat ini dari kriptoanalisis. Situasi bisa saja berubah secara tiba-tiba, tetapi menemukan kerusakan dengan beberapa data yang belum-ada adalah permasalahan yang lebih susah lagi, dan akan selalu butuh waktu untuk terjadinya sebuah transisi.

1. Cara Kerja MD5

Algoritma MD5 adalah algoritma yang menggunakan fungsi hash satu arah yang diciptakan oleh Ron Rivest. Algoritma merupakan pengembangan dari algoritma-algoritma sebelumnya yaitu algoritma MD2 dan algoritma MD4 karena kedua algoritma ini berhasil diserang para *cryptanalist*.

Cara kerja kriptografi algoritma MD5 adalah menerima input berupa pesan dengan ukuran sembarang dan menghasilkan message diggest yang memiliki panjang 128 bit. Berikut ilustrasi gambar dari pembuatan message diggest pada kriptografi algoritma MD5 :



Gambar 3.1 Pembuatan message digest dengan algoritma MD5

Menilik dari gambar diatas, secara garis besar pembuatan message digest ditempuh melalui empat langkah, yaitu :

1. Penambahan bit bit pengganjal

Proses pertama yang dilakukan adalah menambahkan pesan dengan sejumlah bit pengganjal sedemikian sehingga panjang pesan (dalam satuan bit) kongruen dengan 448 modulo 512. Ini berarti setelah menambahkan bit-bit pengganjal, kini panjang pesan adalah 64 bit kurang dari kelipatan 512. Hal yang perlu diingat adalah angka 512 muncul karena algoritma MD5 memproses pesan dalam blok-blok yang berukuran 512.

Apabila terdapat pesan dengan panjang 448 bit, maka pesan tersebut akan tetap ditambahkan dengan bit-bit pengganjal. Pesan akan ditambahkan dengan 512 bit menjadi 96 bit. Jadi panjang bit-bit pengganjal adalah antara 1 sampai 512. Lalu satu hal lagi yang perlu diperhatikan adalah bahwasanya bit-bit pengganjal terdiri dari sebuah bit 1 diikuti dengan sisanya bit 0.

1. Penambahan nilai panjang pesan semula

kemudian proses berikutnya adalah pesan ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula. Apabila panjang pesan lebih besar dari 264 maka yang diambil adalah panjangnya dalam modulo 264. dengan kata lain, jika pada awalnya panjang pesan sama dengan K bit, maka 64 bit yang ditambahkan menyatakan K modulo 264. sehingga setelah proses kedua ini selesai dilakukan maka panjang pesan sekarang adalah 512 bit.

1. Inisialisasi penyangga MD

Pada algoritma MD5 dibutuhkan empat buah penyangga atau buffer, secara berurut keempat nama penyangga diberi nama A, B, C dan D. Masing-masing penyangga memiliki panjang 32 bit. Sehingga panjang total :



Keempat penyangga ini menampung hasil antara dan hasil akhir. Setiap penyangga diinisialisasi dengan nilai-nilai (dalam notasi Hexadesimal) sebagai berikut :

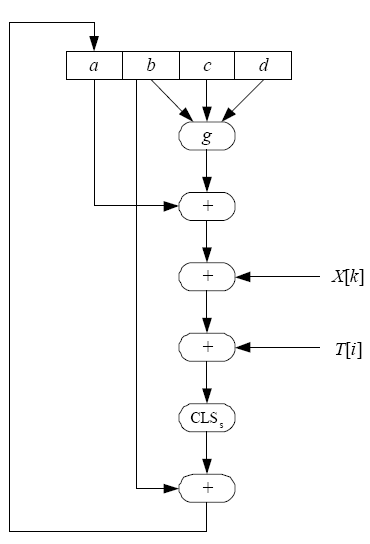


1. Pengolahan pesan dalam blok berukuran 512 bit

Proses berikutnya adalah pesan dibagi menjadi L buah blok yang masing-masing panjangnya 512 bit (Y0 sampai YL-1). Setelah itu setiap blok 512 bit diproses bersama dengan penyangga MD yang menghasilkan keluaran 128 bit, dan ini disebut HMD5. Berikut ini gambaran dari proses HMD5 :

Gambar 3.2 Pengolahan blok 512 bit (Proses HMD5)

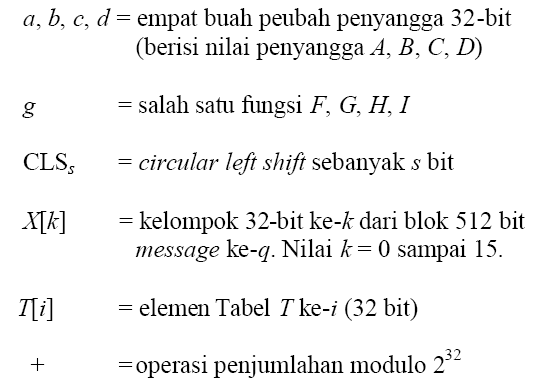
Dari gambar diatas dapat kita lihat bahwa proses HMD5 terdiri dari 4 buah putaran, dan masing-masing putaran melakukan opersi dasar MD5 sebanyak 16 kali. Dimana disetiap operasi dasar memakai sebuah elemen T. Sehingga setiap putaran memakai 16 elemen tabel T.

Pada gambar 3.2, Yq menyatakan blok 512 bit ke-q dari pesan yang telah ditambahkan dengan bit-bit pengganjal pada proses pertama dan tambahan 64 bit nilai panjang pesan semula pada proses kedua. MDq adalah nilai *message digest* 128 bit dari proses HMD5 ke-q. Pada awal proses , MDq berisi nilai inisialisasi penyangga MD. Kemudian fungsi fF, fG, fH, dan fI pada gambar, masing-masing berisi 16 kali operasi dasar terhadap input, setiap operasi dasar menggunakn elemen tabel T. Berikut ini ilustrasi gambar operasi dasar MD5 :

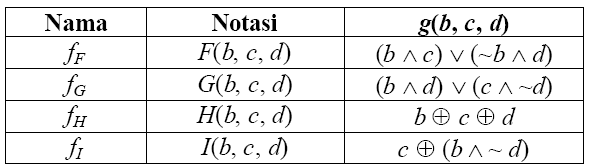
Gambar 3.3 Operasi dasar MD5

Operasi dasar MD5 yang diperlihatkan gambar diatas dapat dituliskan dengan persamaan berikut ini :

https://lh3.googleusercontent.com/yc1r6MUqLhvl51nGPUE-Rv5PUhO24HhD5F0jrYruf-7VTGyBPSu-ym-Qy8ogAS3LXakZf-AMHZ_rPlVJVf-lYQ4YnwGb6xeMpHmfJi_grcO8yi_Hknjyse1ArY7CFa4EDHaRF5Pxoeeup1mplg

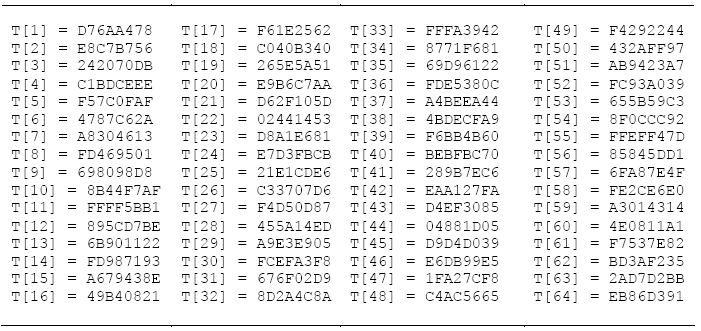
Dimana :

Fungsi fF, fG, fH, dan fI adalah fungsi untuk memanipulasi masukan a, b, c, dan d dengan ukuran 32-bit. Masing-masing fungsi dapat dilihat pada Tabel 1 dibawah ini :

Tabel 1 Fungsi-fungsi dasar MD5

NB: secara berurut peartor logika AND, OR, NOT dan XOR dilambangkan dengan ∧, ∨, ~, ⊕

Kemudian nilai T[i] dapat dilihat pada tabel dibawah ini. Tabel ini disusun oleh fungsi 232 x abs(sin(i)), i dalam radian.

Tabel 2. Nilai T[i]

Sebagaimana telah dijelaskan sebelumnya bahwa fungsi fF, fG, fH, dan fI melakukan 16 kali operasi dasar. Misalkan notasi berikut ini :

https://lh4.googleusercontent.com/KNYqQeuE7ee-uetp4ZzmTtRqVgftNTzZy211ck5GzAYW_1_4tMhQMI9wG3KR-igME2CAhzRDgnGkJACrAtM_w67L7xP8X40BzEdxfqcaGi4BiXtjXuQet5-ACWk8j80Q-7_FiJwRQmN_K8bmMg

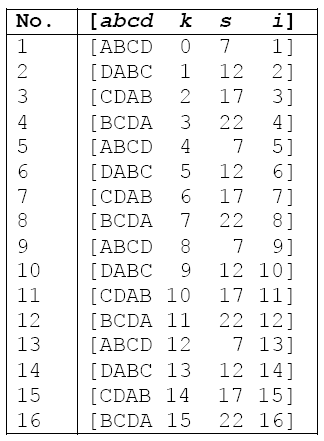
Meyatakan operasi

https://lh3.googleusercontent.com/3iFGid5GP_vunvEXdR1lGEOdFFKzIBDOUBO9X9Oi11mPpYIsFjjCwRqpfpBEpgLfYa-UIw6DqEe7YNzP7zQX1DER8SqeY6-Bk0yyuSynQ9_JmvaVvcDMFD01tvB1KfbefsmB4GqhcHsZdGHi8A

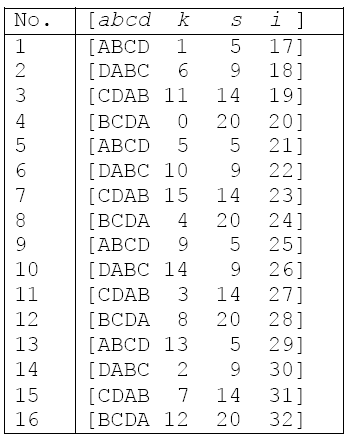
untuk operasi diatas, <<<s melambangkan opersi *circular left shift* 32 bit, maka operasi dasar pada masing-masing putaran dapat ditabulasikan sebagai berikut :

* putaran 1 : 16 kali operasi dasar dengan g(b, c, d) – F(b, c, d), dapat dilihat pada tabel berikut ini :

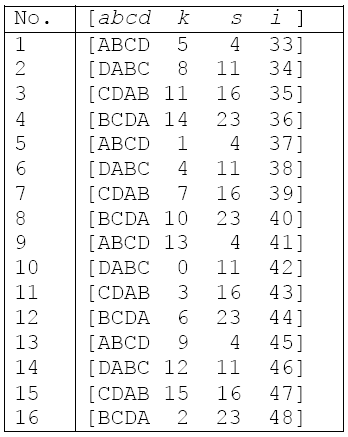
Tabel 3 Rincian operasi pada fungsi F(b, c, d)



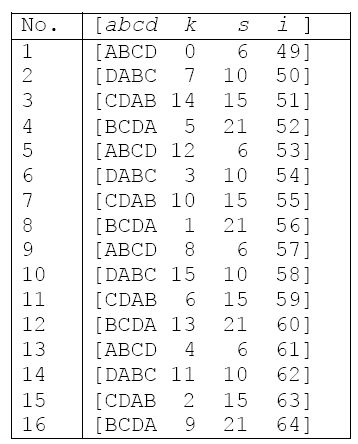
* putaran 2 : 16 kali operasi dasar dengan g(b, c, d) – G(b, c, d), dapat dilihat pada tabel berikut ini :

Tabel 4 Rincian operasi pada fungsi G(b, c, d)

* putaran 3 : 16 kali operasi dasar dengan g(b, c, d) – H(b, c, d), dapat dilihat pada tabel berikut ini

Tabel 5 Rincian operasi pada fungsi H(b, c, d)

* putaran 4 : 16 kali operasi dasar dengan g(b, c, d) – I(b, c, d), dapat dilihat pada tabel berikut ini :

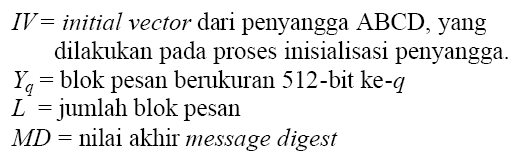
Tabel 6 Rincian operasi pada fungsi I(b, c, d)

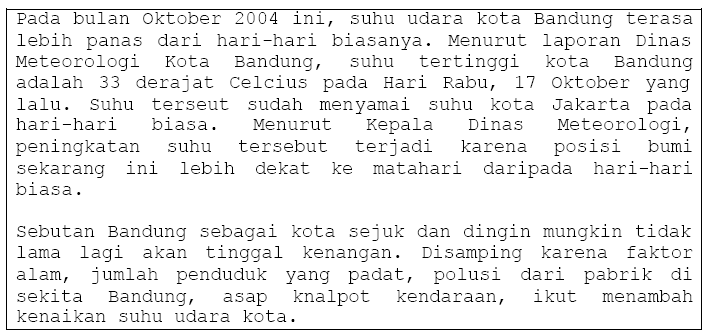
Setelah putaran keempat, a, b, c dan d di tambahkan ke A, B, C dan D yang selanjutnya algoritma akan memproses untuk blok data berikutnya (Yq+1). Output akhir dari algoritma MD5 adalah hasil penyambungan bit-bit di A, B, C dan D.

https://lh4.googleusercontent.com/TP3lYN8Ih19TPMiWo1LpuM_Y9NLRvsotnRzJKeryDgTUdhSQtGt6ZiKhK5KGCU5iaI3DkkRbsCpuOD3gUBtEiRHfqGajQctj8pySqPmmzdt6JZLF70Z_6_EvNeNnNfwwF9TcnsUPAUaDIN2fwQDari uraian diatas, secara umum fungsi hash MD5 dapat ditulis dalam persamaan matematis sebagai berikut :

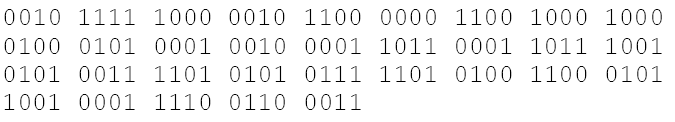
https://lh6.googleusercontent.com/fE3htJqzjmPU1ReBuwuV1-YmDqd378MA37gabrJWfSvtpbRsEDsdnjtF4q_LHGJxMQWUXfvrTVh-OyZD5-PSQHit2etBAhr1Ka9jyq0K7y0PY7I2xjErwA4bk91IwstIuf1LsKbpeL344VTvGg

https://lh6.googleusercontent.com/K7bVR59pKvc-YIIUU0oKrxmw0aF6KLlJiRVhDGuP_PA3NFYYgDo7EwZGj0gSEYhzuWx1m0-tRPOyubtNmuta7_1opt7KAqsrloHYBcWIdOdjBYUCg2ZwlaF_YkTz5mrmMCG4iBM5yYwsXJo3HQ

Dimana

Agar lebih mudah dipahami, berikut ini contoh penerapan kriptografi MD5 pada suatu pesan yang ingin dirahasiakan. Misalkan terdapat sebuah arsip dengan nama bandung.txt sebagai berikut :

Message digest yang dihasilkan dari arsip diatas adalah untuk 128 bit :

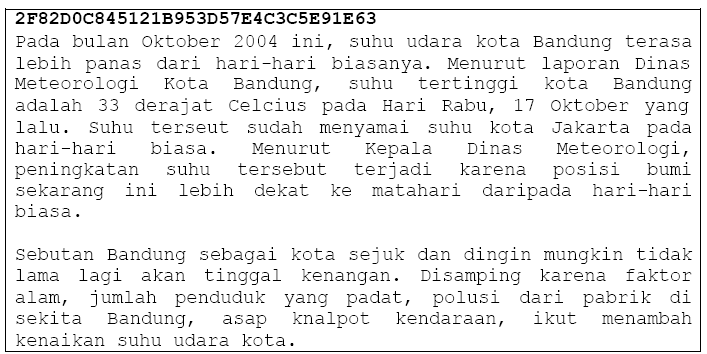


Lalu untuk notasi hexadesimal :

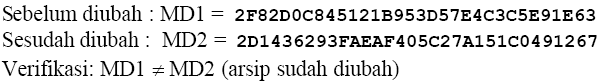
https://lh6.googleusercontent.com/3fEIyvxnVCkaeGo5n1x7sWPueH3dJkoI6ZvcBwu6TE1dRB2Wjdy-ow_WSVIwOUSQIycmeHlYf5TiwlAn37CDHP4n2Lumw9ufwNL-QUXPRDAEFQEksRFHkEBPBHqtbxanXVzeRyS_4r6LbGdSuA

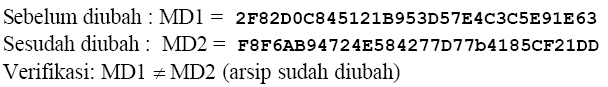
1. Aplikasi Algoritma MD5 Untuk Integritas Data

Aplikasi algoritma dapat digunakan untuk menjaga keintegritasan data. Dengan aplikasi fungsi hash yang menjadi asas algoritma MD5, perubahan kecil pada data sekalipun dapat terdeteksi. Langkah yang harus ditempuh adalah bangkitkan *message digest* dari isi arsip menggunakan algoritma MD5. kemudian gabung *message digest* ke dalam arsip. Verifikasi isi arsip dapat dilakukan secara berkala dengan membandingkan MD isi arsip sekarang dan MD dari arsip asli (MD yang telah disimpan sebelumnya). Jika hasilnya berbeda maka telah terjadi perubahan pada arsip.

Aplikasi ini didasarkan pada kenyataan bahwa perubahan 1 bit pesan akan mengubah, secara rata-rata, setengah dari bit-bit *message digest*. Dengan kata lain, algoritma MD5 dengan fungsi hashnya sangat peka terhadap perubahan sekecil apapun pada data masukan. Sebagai contoh perhatikan arsip dibawah ini :

Untuk kasus pertama, apabila 33 derajat celcius diganti dengan 32, maka MD dari isi arsip adalah (tidak termasuk baris MD) :



Untuk kasus kedua, apabila ditambahkan spasi diantara ”33” dan ”derajat”, maka MD dari isi arsip adalah (tidak termasuk baris MD) :

Dari kedua kasus diatas sangat terlihat bahwa algoritma MD5 dengan fungsi hashnya sangatlah peka dengan perubahan walau kecil sekalipun.

1. Cara Kerja SHA1

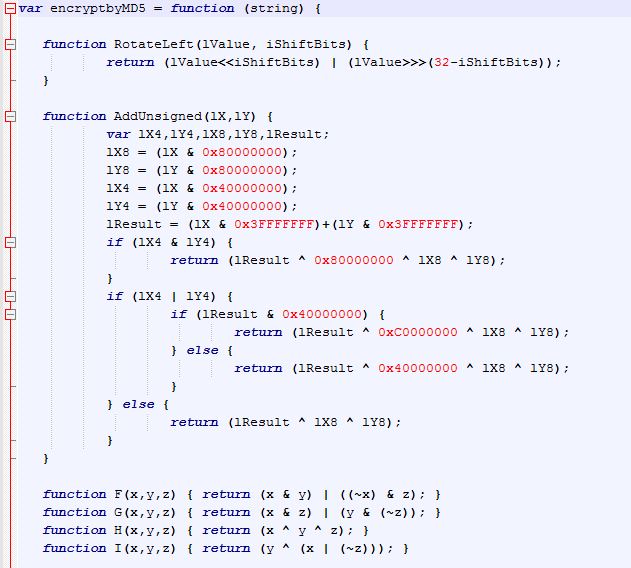
Pesan diberi tambahan untuk membuat panjangnya menjadi kelipatan 512 bit ( l x 512). Jumlah bit asal adalah k bit. Tambahkan bit secukupnya sampai 64 bit kurangnya dari kelipatan 512 ( 512 – 64 = 448 ), yang disebut juga kongruen dengan 448 ( mod 512 ). Kemudian tambahkan 64 bit yang menyatakan panjang pesan. Inisiasi 5 md variable dengan panjang 32 bit yaitu a,b,c,d,e. Pesan dibagi menjadi blok-blok berukuran 512 bit dan setiap blok diolah. Kemudian keluaran setiap blok digabungkan dengan keluaran blok berikutnya, sehingga diperoleh output ( diggest ). Fungsi kompresi yang digunakan oleh algoritma sha-1 adalah sebagai berikut : A,b,c,d,e ← ( e + f (t,b,c,d) + s5 (a) + wt + kt),a,s30(b),c,d.]

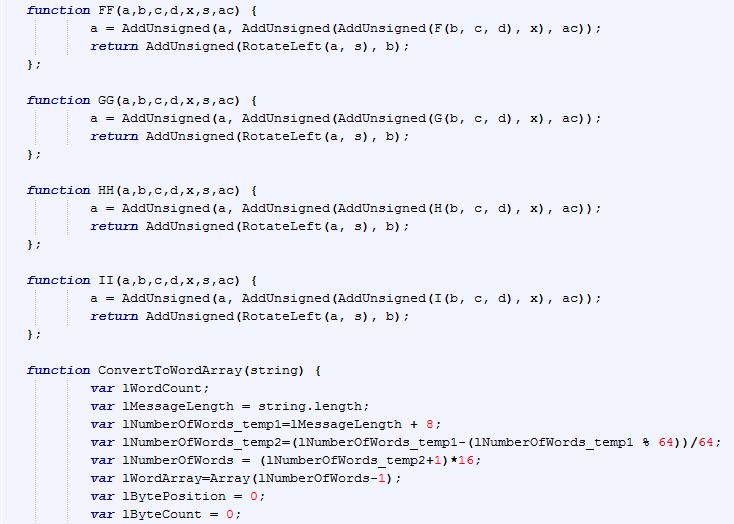
1. Kelebihan dan Kekurangan
2. Keamanan terhadapserangan brute-force. Hal yang paling penting adalah bahwa SHA-1 menghasilkan diggest 32-bit lebih panjang dari MD5. Dengan brute-force maka SHA-1 lebihkuat dibanding MD5.
3. Keamanan terhadap kriptanalisis. Kelemahan MD5 ada pada design sehingga lebih mudah dilakukan kriptanalisis dibandingkan SHA-1.
4. Kecepatan. Kedua algoritma bekerja pada modulo 232 sehingga keduanya bekerja baik pada arsitektur 32 bit. SHA-1 mempunyai langkah lebih banyak dibandingkan MD5 ( 80 dibanding MD5 64 ) dan harus memproses 160 bit buffer dibanding DM5 128 bit buffer, sehingga SHA-1 bekerja lebih lambat dibanding MD5 pada perangkat keras yang sama.

Simplicity. Kedua algoritma simple untuk dijelaskan dan mudah untuk diiemplementasikan karena tidak membutuhkan program yang besar atau table subtitusi yang besar pula.

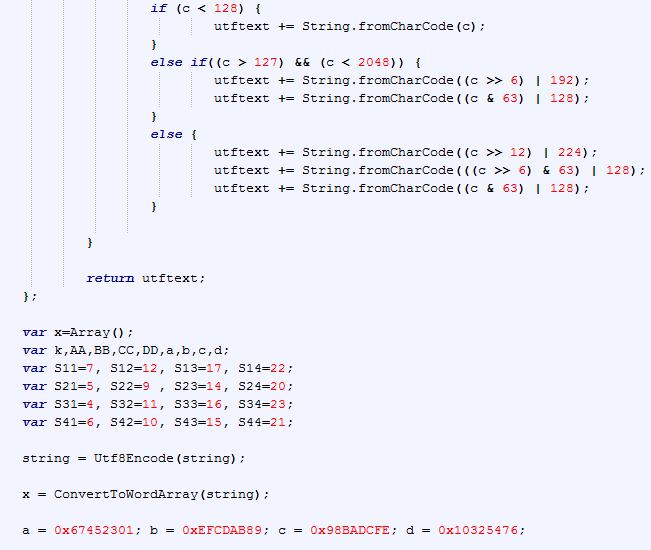
1. Souce Code MD5 dan SHA1

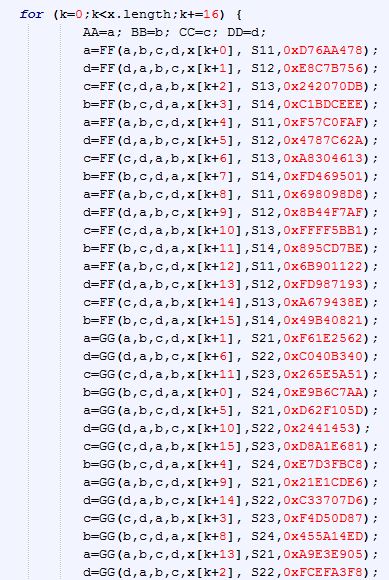
* MD5

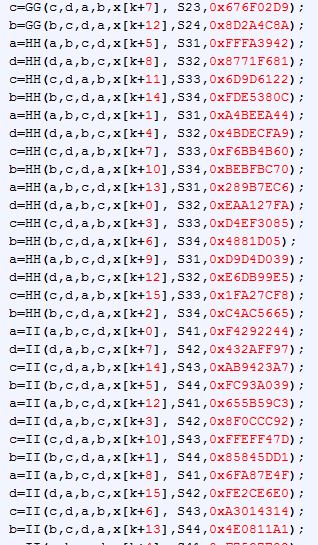


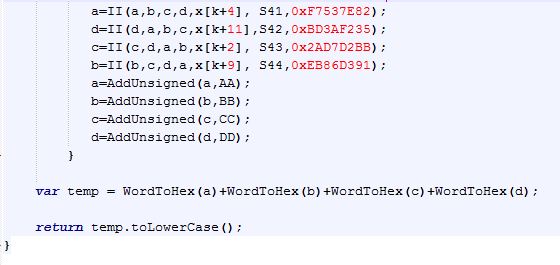




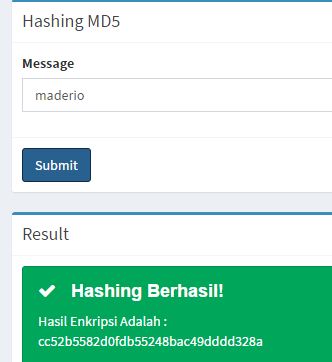






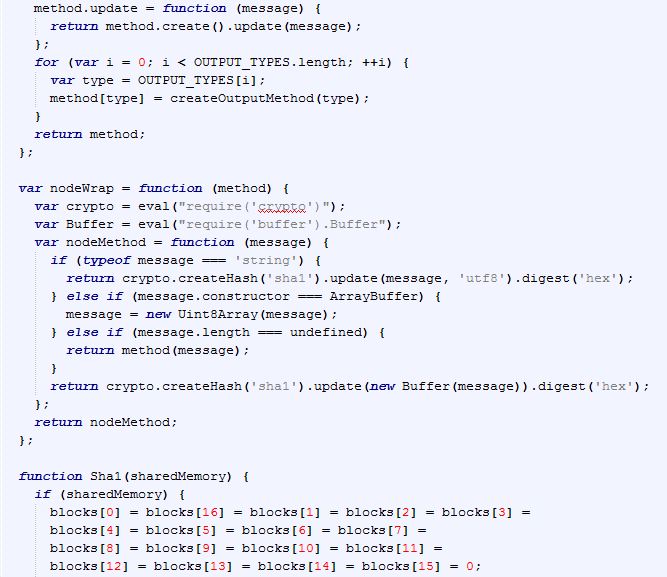


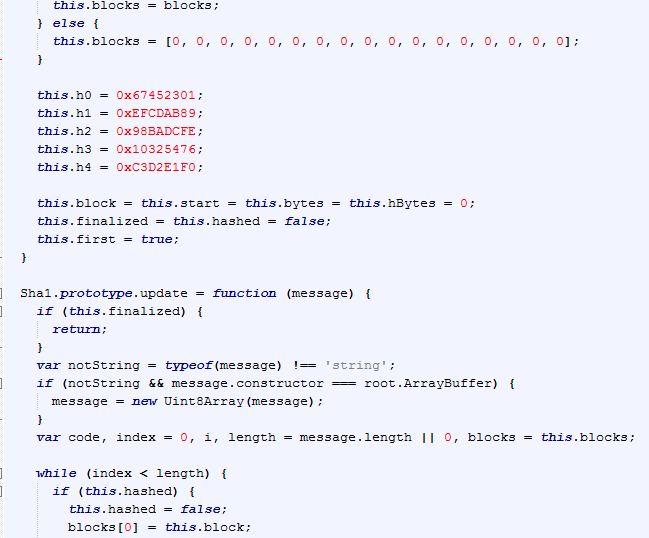
Hasil running :

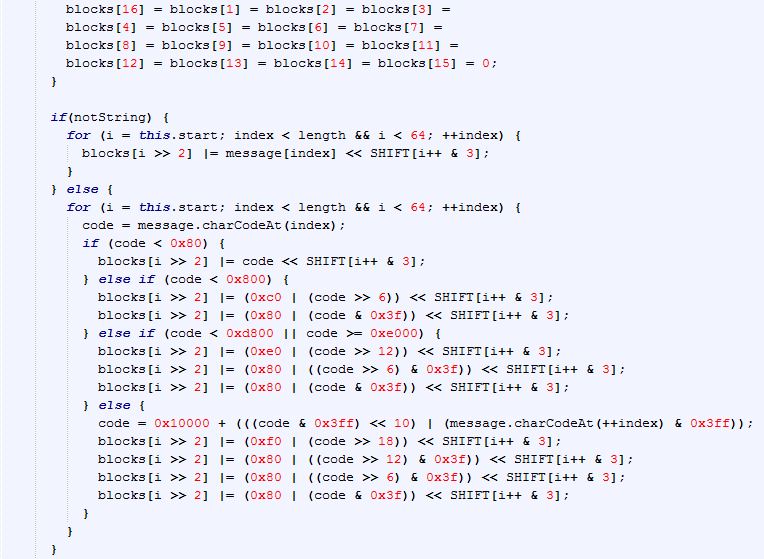


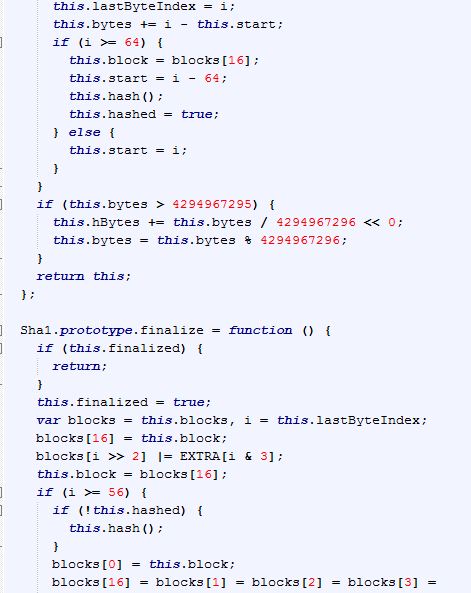
* SHA1

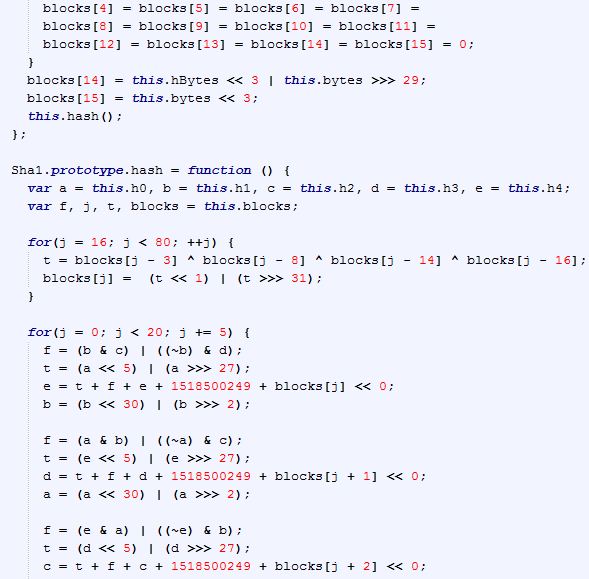


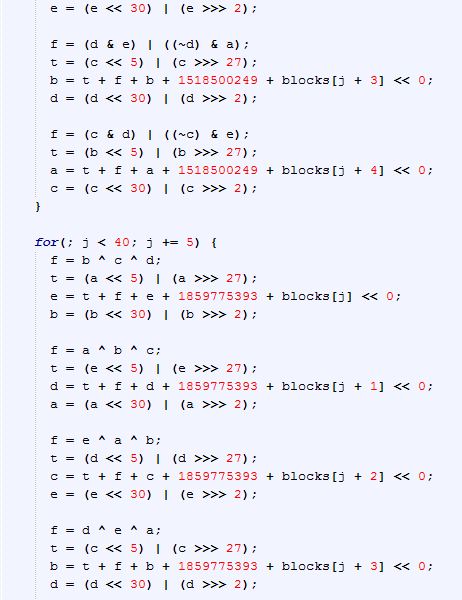


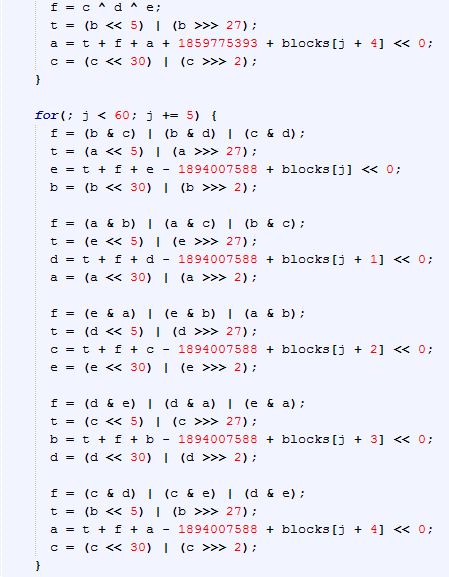


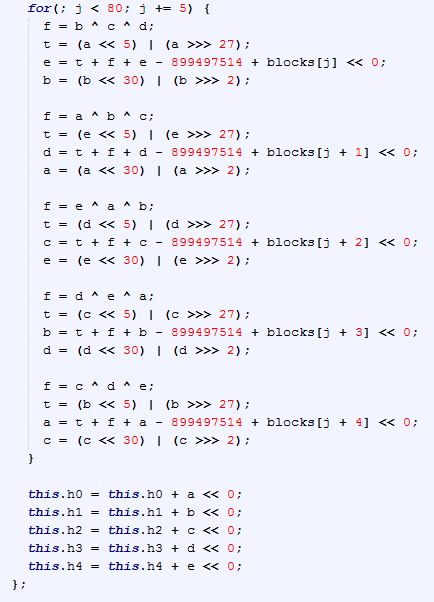


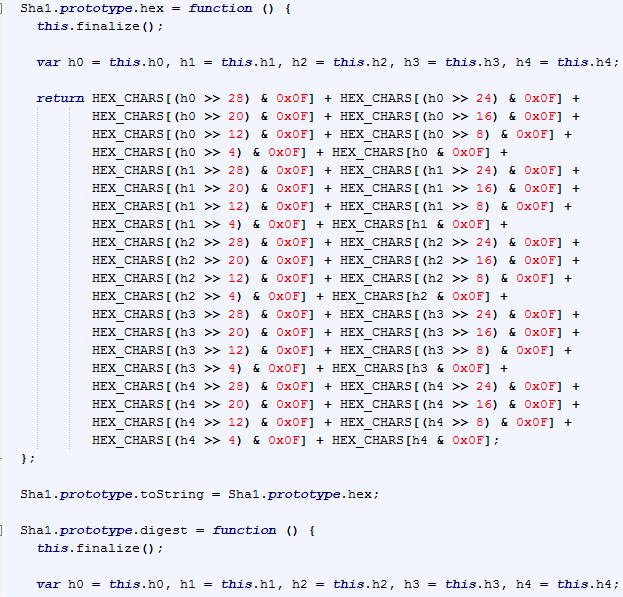


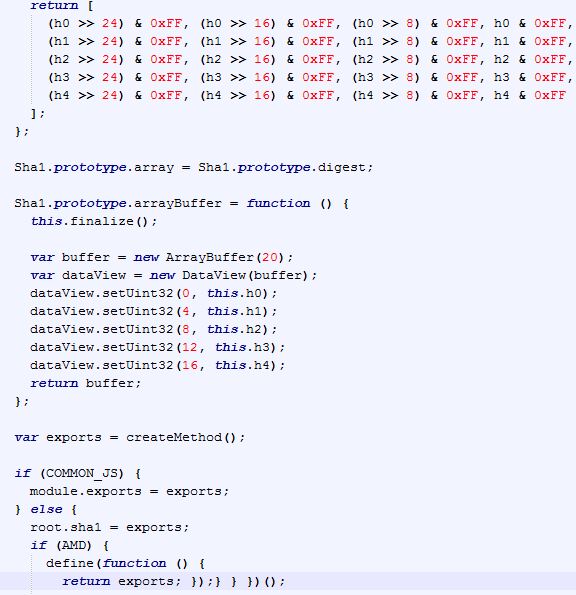


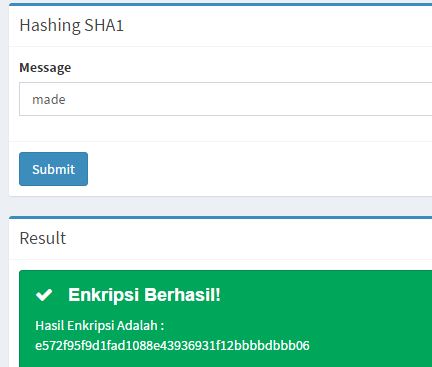












1. Data Encryption Standard (DES)

DES atau Singkatan dari Data Encryption Standard merupakan algoritma penyandian yang diadopsi dan dibakukan oleh NBS (National Bureau Standard) yang kini menjadi NIST (National Institute of Standards and Technology) pada tahun 1977 sebagai FIPS 46 (Federal Information Processing Standard).

DES bermula dari hasil riset Tuchman Meyer yang diajukan sebagai kandidat Sandi Standard Nasional yang diusulkan oleh NBS. Konon katanya, algoritma yang dikembangkan oleh Tuchman Meyer ini merupakan algoritma terbaik dari semua kandidat Sandi Standard Nasional. Pada mulanya, algoritma yang kini disebut DES, memiliki panjang kunci sandi 128 bit. Namun selama proses pengadopsian, NBS melibatkan NSA (National Security Agency), dan algoritma sandi ini mengalami pengurangan ukuran kunci sandi dari 128 bit menjadi 56 bit saja. Sebagian orang mungkin mengira bahwa pengurangan panjang kunci sandi ini merupakan usulan NSA untuk melemahkan algoritma Tuchman Meyer karena motif politik tertentu. Entah itu untuk mempermudah penyadapan atau untuk melemahkan pengamanan informasi lawan politik. Mungkin NSA menginginkan algoritma Tuchman Meyer ini “cukup aman” untuk digunakan warga sipil, tetapi mudah dipecahkan oleh organisasi besar semisal NSA dengan peralatan canggihnya. Bila dibandingkan dengan performa komputer personal pada saat itu, algoritma sandi dengan panjang kunci 56 bit dapat dikatakan cukup aman bila digunakan oleh orang-orang “biasa”, tapi dapat dengan mudah dipecahkan dengan peralatan canggih dan tentunya kepemilikan alat canggih ini hanya dapat dijangkau oleh organisasi elit seperti NSA. Dengan dukungan dana yang melimpah, pembuatan alat brute‐force DES bukanlah hal yang mustahil pada saat itu. Kini algoritma DES sudah usang dan keamanannya pun sudah tidak dapat dipertanggungjawabkan lagi. Kini komputer personal pun sudah cukup untuk membobol algoritma DES, apalagi dengan adanya teknologi parallel computing dan internet yang berkembang pesat. DES telah secara resmi digantikan fungsinya oleh AES (Advanced Encryption Standard) dengan panjang kunci sandi 128, 192 dan 256 bit.

Kendatipun kita telah mengetahui bahwa algoritma AES sudah kuno dan tidak aman, tidak ada salahnya jika kita mempelajari algoritma ini untuk tujuan hobi atau pendidikan. Perlahan tapi pasti, belajar dari algoritma yang sederhana dan perlahan‐lahan menuju algoritma lain yang lebih kompleks.

Algoritma DES merupakan algoritma enkripsi blok simetris. DES dikatakan enkripsi blok karena pemrosesan data baik enkripsi maupun dekripsi, diimplementasikan per blok (dalam hal ini 8 byte). DES dikatakan enkripsi simetris karena algoritma yang digunakan untuk enkripsi relatif atau bahkan sama persis dengan algoritma yang digunakan dalam proses dekripsi. Proses enkripsi dapat didefinisikan secara sederhana sebagai proses penterjemahan data “asli” yang “jelas” dan “kasat mata” yang dapat dipahami maknanya. secara langsung menjadi data lain yang terlihat “buram” atau “acak” sehingga tidak dapat dipahami secara langsung, sedemikian rupa sehingga makna informasi yang disembunyikan tidak lagi dapat diketahui secara langsung kecuali dengan mengembalikan informasi tersebut ke bentuk aslinya. Sedangkan proses dekripsi dapat didefinisikan secara sederhana sebagai proses pengembalian bentuk data, dari data “buram” atau “acak” menjadi data “asli” yang “jelas” dan “kasat mata” yang dapat dipahami maknanya. Algoritma enkripsi umumnya dilengkapi semacam kata sandi (password), untuk memvariasikan fungsi enkripsi tersebut. Data yang sama, kunci yang sama dan algoritma yang sama akan menghasilkan data enkripsi yang sama. Dalam algoritma penyandian DES, kunci yang digunakan dalam proses enkripsi dan dekripsi haruslah sama, supaya data dapat dikembalikan ke bentuk aslinya. Bisa jadi, karena “kesamaan” kunci inilah DES juga dinamakan algoritma enkripsi simetris. Inti dari proses enkripsi adalah penyembunyian data dengan mengaburkan data “asli” dan mengurangi keteraturan informasi, sehingga data tersebut tidak dapat “dibaca” kecuali oleh pihak yang berhak. Berbagai algoritma enkripsi sengaja dibuat untuk melindungi informasi dari penyadapan, karena ada kemungkinan terjadinya penyadapan saat data melewati media hantar (media hantar dapat berupa suara, surat, email, kabel, kertas, frekwensi radio atau apapun itu). Seandainya penyadap dapat menyadap semua informasi yang melalui media hantar, idealnya hasil sadapan tersebut hanya menghasilkan data “sampah” yang tidak berguna. Semua algoritma kriptografi diciptakan untuk mewujudkan kondisi ideal tersebut, tapi sayangya kondisi tersebut sangat sulit dicapai, karena selalu ada cara untuk membalikkan informasi sadapan ke bentuk aslinya.

1. Cara Kerja DES

Dalam DES, algoritma dekripsi tepatnya merupakan proses kebalikan (inverse) algoritma enkripsi. Dalam prakteknya proses pembalikan (proses dekripsi) ini diimplementasikan dengan membalikkan urutan sub kunci yang digunakan dalam proses enkripsi, selebihnya algoritma enkripsi dan dekripsi adalah sama. Algoritma enkripsi DES bekerja dengan mengolah blok data 8 byte (64 bit) dengan blok kunci 8 byte (64 bit). Proses penyandian dalam DES diawali dengan fungsi pengacakan bit yang dinamai IP (Initial Permutation) kemudian fungsi inti DES yang diulang sebanyak 16 kali dan terakhir ditutup dengan fungsi pengacakan bit lain yang dikenal denagn nama IP‐1 (Inverse Initial Permutation). Pada sisi lain algoritma penjadwalan sub kunci akan menghasilkan 16 sub kunci secara berurutan dari parameter kunci yang diberikan untuk digunakan pada setiap putaran fungsi inti DES. Sub kunci pertama untuk putaran pertama, sub kunci kedua untuk putaran kedua dan seterusnya hingga putaran ke 16.

Perlu diingat, kendatipun slot kunci yang disediakan digunakan berukuran 8 byte (64 bit), ternyata pada faktanya ukuran kunci yang digunakan hanya sebanyak 56 bit saja, karena bit paling signifikan (MSB) dari setiap bit diabaikan. Jadi sebenarnya ukuran kunci DES adalah 56 bit. Algoritma penjadwalan sub kunci dibentuk dari pengacakan bit dan pemutaran kiri ruas kanan dan kiri kunci. Pertama kali, bit-bit kunci diacak dengan Permutation Choice 1 dan dibagi dua menjadi ruas kiri dan ruas kanan. Kedua ruas tersebut kemudian diputar kiri dan diacak kembali dengan Permutation Choice 2 untuk menghasilkan sub kunci. Jumlah pemutaran ke kiri ditentukan secara spesifik untuk setiap sub kunci.

Rinciannya adalah sebagai berikut.

[](https://link.safelinkconverter.com/?id=aHR0cDovL3B1c2F0LW1ha2FsYWhrdS5ibG9nc3BvdC5jb20v&c=0&user=61204)

Untuk lebih lanjut, mari terlebih dahulu kita bahas detail algoritma penjadwalan sub kunci kemudian algoritma enkripsi dan terakhir algoritma dekripsi. Inti dari semua proses permutasi dalam DES adalah pengacakan bit. Sebagai contoh, jika masukan permutasi sebanyak n bit, maka akan ada sebanyak 2n kemungkinan masukan permutasi dan ada 2n kemungkinan hasil permutasi. Setiap satu kemungkinan masukan akan berpasangan dengan satu kemungkinan keluaran.

Sebelum proses penjadwalan kunci dimulai, kunci terlebih dahulu dipetakan menjadi matriks 8x8 dan diberi indeks. Dalam setiap byte, indeks paling kecil melambangkan LSB dan indeks paling besar melambangkan MSB. Sebagai contoh, indeks ke 1 melambangkan LSB byte pertama, index ke 8 melambangkan MSB byte pertama, indeks ke 9 melambangkan LSB byte kedua, indeks ke 16 melambangkan MSB byte kedua dan seterusnya hingga indeks ke 64 yang melambangkan MSB byte ke 8. Mari kita perhatikan contoh dibawah ini. Kunci = 0x01, 0x23, 0x45, 0x67, 0x89, 0xab, 0xcd, 0xef

[](https://link.safelinkconverter.com/?id=aHR0cDovL3B1c2F0LW1ha2FsYWhrdS5ibG9nc3BvdC5jb20v&c=0&user=61204)

Setelah kunci diacak dengan Permutation Choice 1, hasil pengacakan bit tersebut kemudian dibagi 2, yakni ruas kiri dan ruas kanan, masing masing berukuran 28 bit (ditandai dengan garis tebal pada hasil Permutation Choice 1). Selanjutnya kedua ruas tersebut kemudian mengalami pemutaran kiri sebanyak jumlah yang tertera pada tabel penjadwalan jumlah pemutaran yang telah kita bahas sebelumnya. Berikut ini adalah ilustrasi pemutaran ke kiri sebanyak 1 kali (untuk ruas kiri atau ruas kanan kunci yang panjangnya 28 bit). Untuk pemutaran ke kiri dengan jumlah yang lebih besar, cukup mengulangi proses ini sebanyak yang diinginkan.

[](https://link.safelinkconverter.com/?id=aHR0cDovL3B1c2F0LW1ha2FsYWhrdS5ibG9nc3BvdC5jb20v&c=0&user=61204)

Setelah ruas kiri dan ruas kanan diputar kiri dengan jumlah tertentu, selanjutnya hasil pemutaran tersebut digabungkan kembali menjadi 56 bit dan diacak dengan Permutation Choice‐2 untuk menghasilkan sub kunci. Rincian Permutation Choice 2 adalah sebagai berikut.

[](https://link.safelinkconverter.com/?id=aHR0cDovL3B1c2F0LW1ha2FsYWhrdS5ibG9nc3BvdC5jb20v&c=0&user=61204)

Karena jumlah keluaran Permutation Choice 48 bit sementara masukannya 56 bit, dengan demikian ada 8 bit yang “dihilangkan”. Bit-bit yang tidak muncul dalam keluaran Permutation Choice 2 diwarnai abuabu. Selanjutnya, mari kita perjelas algoritma enkripsinya. Setelah melihat diagram blok secara keseluruhan proses enkripsi, ada tiga hal yang perlu digarisbawahi dan dibahas lebih lanjut yaitu, pertama IP (Initial Permutation), kedua detail fungsi F dan IP 1 (Inverse Initial Permutation).

Selama proses enkripsi, pertama data dipetakan dan diberi indeks dengan prosedur sama persis seperti pemberian indeks pada penjadwalan kunci yang telah didiskusikan sebelumnya. Selanjutnya hasil pemetaan diacak dengan menggunakan Initial Permutation dengan rincian sebagai berikut :

[](https://link.safelinkconverter.com/?id=aHR0cDovL3B1c2F0LW1ha2FsYWhrdS5ibG9nc3BvdC5jb20v&c=0&user=61204)

Setelah data melewati proses Initial Permutation, data yang akan disandikan kemudian dibagi menjadi dua ruas, yaitu ruas kiri dan ruas kanan yang masIng-masing lebarnya 4 byte (32 bit). Pada setiap putaran, ruas kanan dan sub kunci yang bersangkutan diproses dalam fungsi F dan hasilnya di XOR dengan ruas kiri ruas kanan dan kiri dipertukarkan. Proses ini diulang sebanyak 16 kali.

Pada putaran terakhir ruas kiri dan ruas kanan dipertukarkan kembali untuk menghilangkan efek pertukaran pada putaran terakhir. Hasil ini kemudian diacak kembali dengan menggunakan IP 1 (Inverse Initial Permutation). XOR merupakan fungsi Boolean yang didefinisikan dengan table benaran berikut.

[](https://link.safelinkconverter.com/?id=aHR0cDovL3B1c2F0LW1ha2FsYWhrdS5ibG9nc3BvdC5jb20v&c=0&user=61204)

Initial Permutatrion dan Inverse Initial Permutation memiliki karakter saling menetralkan. Dalam notasi matematika, IP−1(IP(A))= A. Inverse Initial Permutation didefinisikan sebagai berikut :

[](https://link.safelinkconverter.com/?id=aHR0cDovL3B1c2F0LW1ha2FsYWhrdS5ibG9nc3BvdC5jb20v&c=0&user=61204)

Fungsi F merupakan fungsi inti kompleks yang terdiri dari beberapa proses. Fungsi F menerima dua parameter, yaitu sub kunci dan ruas kanan data yang akan dienkripsi. Berikut ini adalah diagram blok detail fungsi F. Pada fungsi F, ruas kanan (32 bit) diacak sekaligus diperluas dengan permutasi E menjadi 48 bit. Hasil pengacakan tersebut kemudian di XOR dengan sub kunci yang telah ditetapkan dengan putaran yang bersangkutan. Hasil XOR kemudian dipecah menjadi 8 unit yang masing‐masing lebarnya 6 bit. Setiap unit tersebut kemudian disubstitusikan dalam SBOX S1 hingga S8. 6 bit paling kiri disubstitusikan ke dalam S1 dan 6 bit paling kanan disubstitusikan ke dalam S8. Hasil setiap substitusi kemudian digabungkan menjadi data selebar 48 bit yang kemudian diacak dan diperpendek dengan permutasi P menjadi 32 bit. Hasil permutasi P kemudian dinyatakan sebagai keluaran fungsi F yang nantinya akan di XOR kan dengan ruas kiri data yang akan dienkripsi.

[](https://link.safelinkconverter.com/?id=aHR0cDovL3B1c2F0LW1ha2FsYWhrdS5ibG9nc3BvdC5jb20v&c=0&user=61204)

Sekarang mari kita bahas detail fungsi F satu per satu. Pertama, permutasi E memetakan 32 bit masukan menjadi 48 bit keluaran. Karena lebar keluaran lebih besar dari lebar masukan, maka ada beberapa bit masukan yang digandakan untuk mengisi kekosongan. Permutasi E didefinisikan sebagai berikut :

[](https://link.safelinkconverter.com/?id=aHR0cDovL3B1c2F0LW1ha2FsYWhrdS5ibG9nc3BvdC5jb20v&c=0&user=61204)

Kedua, DES memiliki 8 buah SBOX (S1 hingga S8) yang memiliki masukan selebar 6 bit dan keluaran selebar 4 bit. Karena lebar keluaran SBOX lebih kecil daripada lebar masukannya, maka adakemungkinan beberapa kombinasi masukan yang berbeda akan menghasilkan keluaran yang sama. Seandainya masukan setiap SBOX adalah 1 2 3 4 5 6 x x x x x x maka S1 hingga hingga S8 didefinisikan sebagai berikut.

[](https://link.safelinkconverter.com/?id=aHR0cDovL3B1c2F0LW1ha2FsYWhrdS5ibG9nc3BvdC5jb20v&c=0&user=61204)

[](https://link.safelinkconverter.com/?id=aHR0cDovL3B1c2F0LW1ha2FsYWhrdS5ibG9nc3BvdC5jb20v&c=0&user=61204)

Ketiga, hasil substitusi SBOX kemudian digabungkan menjadi 32 bit dan diacak dengan permutasi P dan hasil permutasi P merupakan keluaran fungsi F yang nantinya di XOR dengan ruas kiri. Permutasi P didefinisikan sebagai berikut.

[](https://link.safelinkconverter.com/?id=aHR0cDovL3B1c2F0LW1ha2FsYWhrdS5ibG9nc3BvdC5jb20v&c=0&user=61204)

Dalam DES, algoritma yang digunakan dalam proses enkripsi sama persis dengan algoritma yang digunakan dalam proses dekripsi, hanya saja penggunaan sub kuncinya saja yang berbeda. Dalam proses dekripsi, urutan sub kunci yang digunakan merupakan kebalikan urutan sub kunci yang digunakan dalam proses enkripsi.

1. Implementasi DES

Operasi yang digunakan dalam algoritma DES merupakan operasi‐operasi sederhana semisal move, bit copy, XOR, lookup, shift dan rotate. Semua operasi tersebut tersedia dalam mikroprosesor/mikrokontroler 8 bit. Dengan demikian dapat kita simpulkan bahwa DES dapat diterapkan dalam platform 8 bit.

Semua operasi permutasi dalam DES, baik IP, IP 1, PC 1, PC 2, E dan P, pada intinya hanyalah operasi penyalinan bit. Jika instruksi penyalinan bit tidak tersedia, maka permutasi juga dapat diimplementasikan dengan operasi shift, dengan memanfaatkan carry yang timbul dari setiap instruksi shift. Selain itu, operasi substitusi dengan SBOX juga dapat dengan mudah diimplementasikan menggunakan table lookup dengan ukuran yang masih dapat dijangkau. Dalam platform 32 atau 64 bit, DES dapat diimplementasikan lebih effektif lagi, tapi sayangnya operasi bit per bit seperti permutasi mungkin sedikit menyita performa prosesor dan memperlambat laju enkripsi per detik.

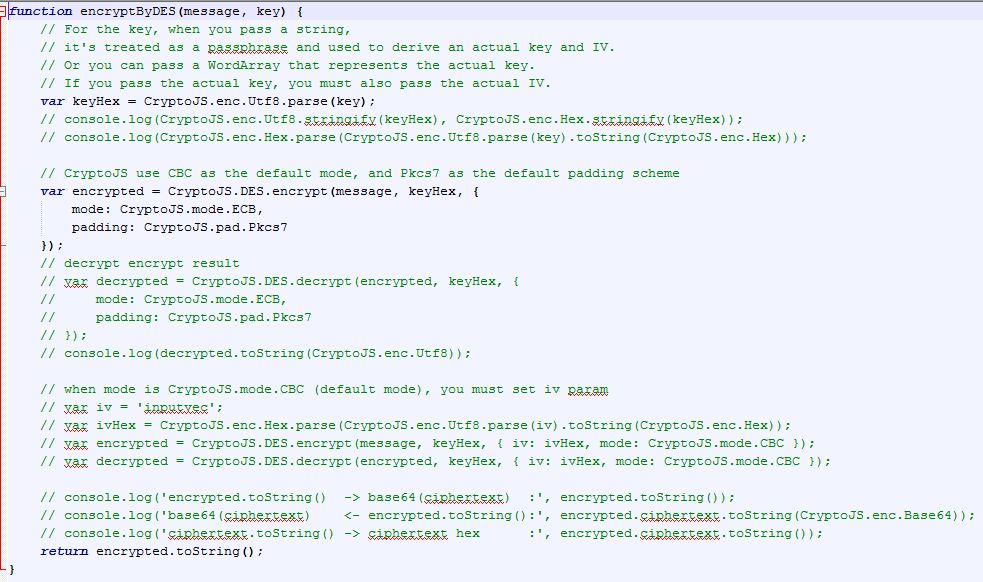
1. Kelebihan dan Kekurangan DES

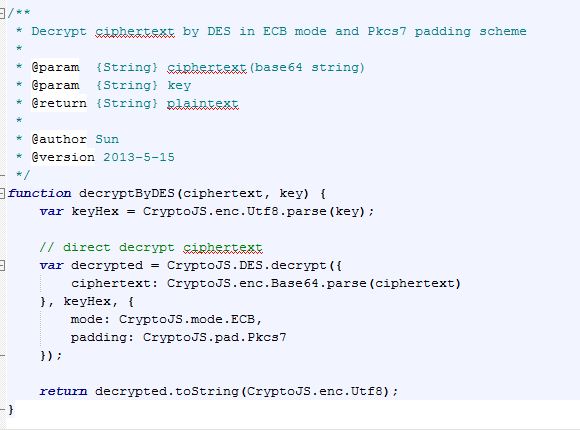
* Kelebihan DES

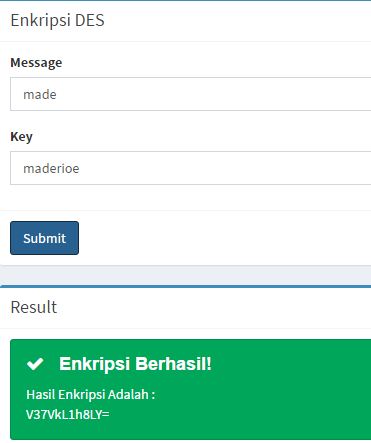
Kelebihan DES yaitu, sistem sandi lebih kompleks sehingga membutuhkan waktu yang tidak cepat untuk menembus enkripsi DES dan sulit untuk diketahui dari pihak luar. Tetapi setelah berkembangannya jaman DES tidak digunakan karena ukuran kunci yang terlalu kecil, sehingga mudah untuk ditembus.

* Kekurangan DES

Data Encryption Standard ( DES ) pun seperti sistem yang lainnya, memiliki kekurangan dan kelebihan. Salah satu kekurangan DES adalah proses yang lebih lama dalam melakukan proses dekripsi dan enkripsi.

1. Source Code DES



Hasil Running :

1. Definisi Algoritma AES

AES (Advanced Encryption Standard) merupakan algoritma cryptographic yang dapat digunakan untuk mengamankan data. Algoritma AES adalah blok chipertext simetrik yang dapat mengenkripsi (enchiper) dan deskripsi (decipher) informasi. Enkripsi merubah data yang tidak dapat dibaca lagi disebut ciphertext, sebaliknya deskripsi adalah merubah ciphertext data menjadi bentuk semula yang kita kenal sebagai plaintext. Algoritma AES menggunakan kunci kriptografi 128, 192, dan 256 bit.



Gambar 3.5 - Panjang Kunci Algoritma Kriptografi AES

AES adalah lanjutan dari algoritma enkripsi standar DES (Data Encryption Standard) yang masa berlakunya dianggap telah usai karena faktor keamanan. Kecepatan komputer yang sangat pesat dianggap sangat membahayakan DES, sehingga pada tanggal 2 Maret tahun 2001 ditetapkanlah algoritma baru Rinjadael sebagai AES. Kriteria pemilihan AES didasarkan pada 3 kriteria utama yaitu: keamanan, harga, dan karakteristik algoritma beserta implementasinya. Keamanan merupakan faktor penting dalam evaluasi (minimal semanan Triple DES), yang meliputi ketahanan terhadap semua analisis sandi yang telah diketahui dan diharapkan dapat menghadapi analisis sandi yang belum diketahui. Di samping itu, AES juga harus dapat digunakan secara bebas tanpa harus membayar royalti, dan juga murah untuk diimplementasikan pada smart card yang memiliki ukuran memori kecil. AES juga harus efisien dan cepat (minimal secepat Triple DES) dijalankan dalam berbagai mesin 8 bit hingga 64 bit, dan berbagai perangkat lunak. DES menggunakan stuktur Feistel yang memiliki kelebihan bahwa struktur enkripsi dan dekripsinya sama, meskipun menggunakan fungsi F yang tidak invertibel. Kelemahan Feistel yang utama adalah bahwa pada setiap ronde, hanya setengah data yang diolah. Sedangkan AES menggunakan struktur SPN (Substitution Permutation Network) yang memiliki derajat paralelisme yang lebih besar, sehingga diharapkan lebih cepat dari pada Feistel.

Kelemahan SPN pada umumnya (termasuk pada Rijndael) adalah berbedanya struktur enkripsi dan dekripsi sehingga diperlukan dua algoritma yang berbeda untuk enkripsi dan dekripsi. Dan tentu pula tingkat keamanan enkripsi dan dekripsinya menjadi berbeda. AES memiliki blok masukan dan keluaran serta kunci 128 bit. Untuk tingkat keamanan yang lebih tinggi, AES dapat menggunakan kunci 192 dan 256 bit. Setiap masukan 128 bit plaintext dimasukkan ke dalam state yang berbentuk bujursangkar berukuran 4×4 byte. State ini di-XOR dengan key dan selanjutnya diolah 10 kali dengan subtitusi-transformasi linear-Addkey. Dan di akhir diperoleh ciphertext.

Berikut ini adalah operasi Rijndael (AES) yang menggunakan 128 bit kunci:

1. Ekspansi kunci utama (dari 128 bit menjadi 1408 bit)

2. Pencampuran subkey

3. Ulang dari i=1 sampai i=10 Transformasi : ByteSub (subtitusi per byte), ShiftRow (Pergeseren byte perbaris), MixColumn (Operasi perkalian GF(2) per kolom)

4. Pencampuran subkey (dengan XOR)

5. Transformasi : ByteSub dan ShiftRow

6. Pencampuran subkey

1. Sejarah Advanced Encryption Standard

Pada rahun 1997, National Institute of Standard and Technology (NIST) of United States mengeluarkan Advanced Encryption Standard (AES) untuk menggantikan Data Encryption Standard (DES). AES dibangun dengan maksud untuk mengamankan pemerintahan di berbagai bidang. Algoritma AES di desain menggunakan blok chiper minimal dari blok 128 bit input dan mendukung ukuran 3 kunci (3-key-sizes), yaitu kunci 128 bit, 192 bit, dan 256 bit. Pada Agustus 1998, NIST mengumumkan bahwa ada 15 proposal AES yang telah diterima dan dievaluasi, setelah mengalami proses seleksi terhadap algoritma yang masuk, NIST mengumumkan pada tahun 1999 bahwa hanya ada 5 algoritma yang diterima, algoritma tersebut adalah:

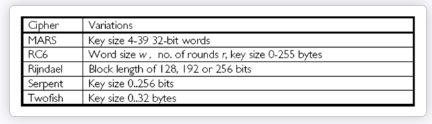
a. MARS

b. RC6

c. Rijndael

d. Serpent

f. Twofish



Gambar 3.6 Algoritma yang diterima oleh NIST

Algoritma-algoritma tersebut manjalani berbagai macam pengetesan. Pada bulan Oktober 2000, NIST mengumumkan bahwa Rijndael sebagai algoritma yang terpilih untuk standar AES yang baru. Baru pada Februari 2001 NIST mengirimkan draft kepada Federal Information Processing Standards (FIPS) untuk standar AES. Kemudian pada 26 November 2001, NIST mengumumkan produk akhir dari Advanced Encryption Standard (AES).

1. Cara Kerja AES

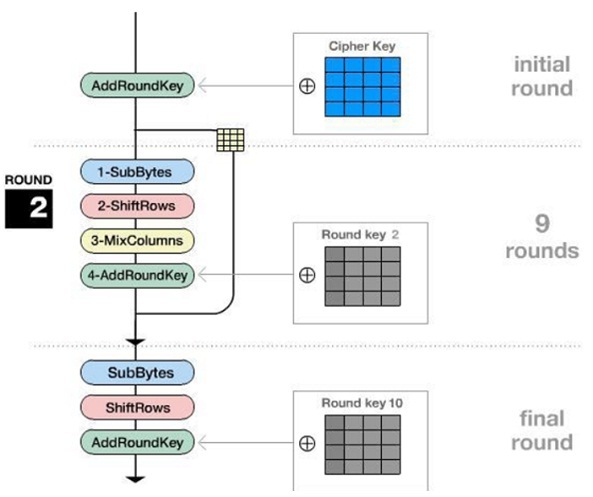
Algoritma kriptografi bernama Rijndael yang didesain oleh oleh Vincent Rijmen dan John Daemen asal Belgia keluar sebagai pemenang kontes algoritma kriptografi pengganti DES yang diadakan oleh NIST (National Institutes of Standards and Technology) milik pemerintah Amerika Serikat pada 26 November 2001. Algoritma Rijndael inilah yang kemudian dikenal dengan Advanced Encryption Standard (AES). Setelah mengalami beberapa proses standardisasi oleh NIST, Rijndael kemudian diadopsi menjadi standard algoritma kriptografi secara resmi pada 22 Mei 2002. Pada 2006, AES merupakan salah satu algoritma terpopuler yang digunakan dalam kriptografi kunci simetrik.

AES ini merupakan algoritma block cipher dengan menggunakan sistem permutasi dan substitusi (P-Box dan S-Box) bukan dengan jaringan Feistel sebagaimana block cipher pada umumnya. Jenis AES terbagi menjadi 3, yaitu:

1. AES-128
2. AES-192
3. AES-256

Pengelompokkan jenis AES ini adalah berdasarkan panjang kunci yang digunakan. Angka-angka di belakang kata AES menggambarkan panjang kunci yang digunakan pada tiap-tiap AES. Selain itu, hal yang membedakan dari masing-masing AES ini adalah banyaknya round yang dipakai. AES-128 menggunakan 10 round, AES-192 sebanyak 12 round, dan AES-256 sebanyak 14 round.

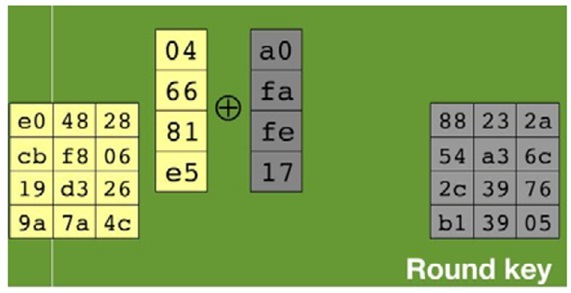
AES memiliki ukuran block yang tetap sepanjang 128 bit dan ukuran kunci sepanjang 128, 192, atau 256 bit. Tidak seperti Rijndael yang block dan kuncinya dapat berukuran kelipatan 32 bit dengan ukuran minimum 128 bit dan maksimum 256 bit. Berdasarkan ukuran block yang tetap, AES bekerja pada matriks berukuran 4x4 di mana tiap-tiap sel matriks terdiri atas 1 byte (8 bit). Sedangkan Rijndael sendiri dapat mempunyai ukuran matriks yang lebih dari itu dengan menambahkan kolom sebanyak yang diperlukan. Blok chiper tersebut dalam pembahasan ini akan diasumsikan sebagai sebuah kotak. Setiap plainteks akan dikonversikan terlebih dahulu ke dalam blok-blok tersebut dalam bentuk heksadesimal. Barulah kemudian blok itu akan diproses dengan metode yang akan dijelaskan. Secara umum metode yang digunakan dalam pemrosesan enkripsi dalam algoritma ini dapat dilihat melalui Gambar 3.7.



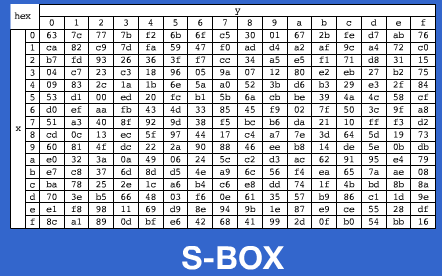
Gambar 3.7 Diagram AES

1. Add Round Key

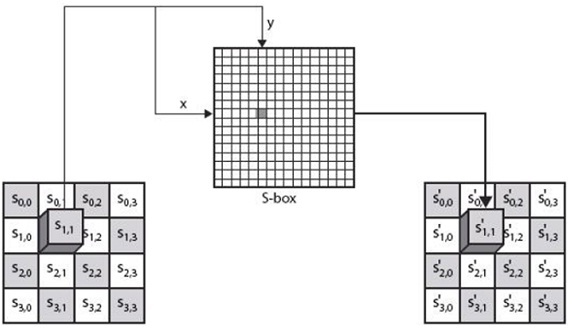
Add Round Key merupakan transformasi yang melakukan operasi XOR terhadap sebuah round key dengan array state dan hasilnya disimpan di array state.

Gambar 3.8 Ilustrasi Add Round Key

1. Sub Bytes

Prinsip dari Sub Bytes adalah menukar isi matriks/tabel yang ada dengan matriks/tabel lain yang disebut dengan Rijndael S-Box. Dibawah ini adalah contoh Sub Bytes dan Rijndael S-Box.

Gambar 3.9 Rjindael S-Box



Gambar 3.10. Ilustrasi Sub Bytes

Keterangan: Gambar 3.10 adalah contoh dari Rijndael S-Box, di sana terdapat nomor kolom dan nomor baris. Seperti yang telah disebutkan sebelumnya, tiap isi kotak dari blok chiper berisi informasi dalam bentuk heksadesimal yang terdiri dari dua digit, bisa angka-angka, angka-huruf, ataupun huruf-angka yang semuanya tercantum dalam Rijndael S-Box. Langkahnya adalah mengambil salah satu isi kotak matriks, mencocokkannya dengan digit kiri sebagai baris dan digit kanan sebagai kolom. Kemudian dengan mengetahui kolom dan baris, kita dapat mengambil sebuah isi tabel dari Rijndael S-Box. Langkah terakhir adalah mengubah keseluruhan blok chiper menjadi blok yang baru yang isinya adalah hasil penukaran semua isi blok dengan isi langkah yang disebutkan sebelumnya.

1. Shift Rows

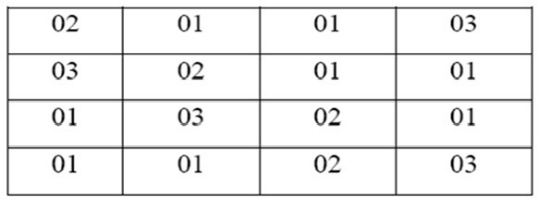
****Shift Rows seperti namanya adalah sebuah proses yang melakukan shift atau pergeseran pada setiap elemen blok/tabel yang dilakukan per barisnya. Yaitu baris pertama tidak dilakukan pergeseran, baris kedua dilakukan pergeseran 1 byte, baris ketiga dilakukan pergeseran 2 byte, dan baris keempat dilakukan pergeseran 3 byte. Pergeseran tersebut terlihat dalam sebuah blok adalah sebuah pergeseran tiap elemen ke kiri tergantung berapa byte tergesernya, tiap pergeseran 1 byte berarti bergeser ke kiri sebanyak satu kali. Ilustrasi dari tahap ini diperlihatkan oleh Gambar 3.11 Ilustrasi Shift Rows di bawah ini.

**** Gambar 3.11 Ilustrasi Shift Rows

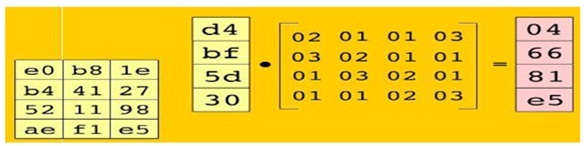
Seperti yang terlihat pada Gambar 3.11, tahap Shift Row sama sekali tidaklah rumit, karena ini adalah proses standar yang hanya berupa pergeseran. Langkah terakhir adalah Mix Column.

1. Mix Column

Yang terjadi saat *Mix Column* adalah mengalikan tiap elemen dari blok *chiper* dengan matriks yang ditunjukkan oleh tabel ..... . Tabel sudah ditentukan dan siap pakai. Pengalian dilakukan seperti perkalian matriks biasa yaitu menggunakan *dot product* lalu perkalian keduanya dimasukkan ke dalam sebuah blok chiper baru. Ilustrasi dalam Gambar 7 akan menjelaskan mengenai bagaimana perkalian ini seharusnya dilakukan. Dengan begitu seluruh rangkaian proses yang terjadi pada AES telah dijelaskan dan selanjutnya adalah menerangkan mengenai penggunaan tiap-tiap proses tersebut.



Tabel untuk Mix Column

****

Gambar 3.12 Ilustrasi Mix Column

1. Diagram alir AES

******Kembali melihat diagram yang ditunjukkan oleh Gambar 1. Seperti yang terlihat semua proses yang telah dijelaskan sebelumnya terdapat pada diagram tersebut. Yang artinya adalah mulai dari ronde dua, dilakukan pengulangan terus menerus dengan rangkaian proses Sub Bytes, Shift Rows, Mix Columns, dan Add Round Key, setelah itu hasil dari ronde tersebut akan digunakan pada ronde berikutnya dengan metode yang sama. Namun pada ronde kesepuluh, proses Mix Columns tidak dilakukan, dengan kata lain urutan proses yang dilakukan adalah Sub Bytes, Shift Rows, dan Add Round Key, hasil dari Add Round Key inilah yang dijadikan sebagai chiperteks dari AES. Lebih jelasnya bisa dilihat dengan Gambar 8 dan 9 yang akan menerangkan mengenai kasus tersebut.

Gambar 3.13 Ilustrasi Ronde 2 hingga Ronde 6

****

Gambar 3.14 Ilustrasi Ronde 7 hingga Ronde 10

Dengan mengetahui semua proses yang ada pada AES, maka kita dapat menggunakannya dalam contoh kasus yang muncul di kehidupan sehari-hari.

1. Implementasi Advanced Encryption Standard (AES)

AES atau algoritma Rijndael sebagai salah satu algoritma yang penting tentu memiliki berbagai kegunaan yang sudah diaplikasikan atau diimplementasikan di kehidupan sehari-hari yang tentu saja membutuhkan suatu perlindungan atau penyembunyian informasi di dalam prosesnya. Salah satu contoh penggunaan AES adalah pada kompresi 7-Zip. Salah satu proses di dalam 7-Zip adalah mengenkripsi isi dari data dengan menggunakan metode AES-256. Yang kuncinya dihasilkan melalui fungsi Hash. Perpaduan ini membuat suatu informasi yang terlindungi dan tidak mudah rusak terutama oleh virus yang merupakan salah satu musuh besar dalam dunia komputer dan informasi karena sifatnya adalah merusak sebuah data.

Hal yang serupa digunakan pada WinZip sebagai salah satu perangkat lunak yang digunakan untuk melakukan kompresi. Tapi prinsip kompresi pun tidak sama dengan prinsip enkripsi. Karena kompresi adalah mengecilkan ukuran suatu data, biasanya digunakan kode Huffman dalam melakukan hal tersebut. Contoh penggunaan lain adalah pada perangkat lunak DiskCryptor yang kegunaannya adalah mengenkripsi keseluruhan isi disk/partisi pada sebuah komputer. Metode enkripsi yang ditawarkan adalah menggunakan AES-256, Twofish, atau Serpent.

1. RSA

Sebagai algoritma kunci publik, RSA mempunyai dua kunci, yaitu kunci publik dan kunci rahasia. Kunci publik boleh diketahui oleh siapa saja, dan digunakan untuk proses enkripsi. Sedangkan kunci rahasia hanya pihak-pihak tertentu saja yang boleh mengetahuinya, dan digunakan untuk proses dekripsi. Keamanan sandi RSA terletak pada sulitnya memfaktorkan bilangan yang besar. Sampai saat ini RSA masih dipercaya dan digunakan secara luas di internet.

Sandi RSA terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses deskripsi. Sebelumnya diberikan terlebih dahulu beberapa konsep perhitungan matematis yang digunakan RSA.

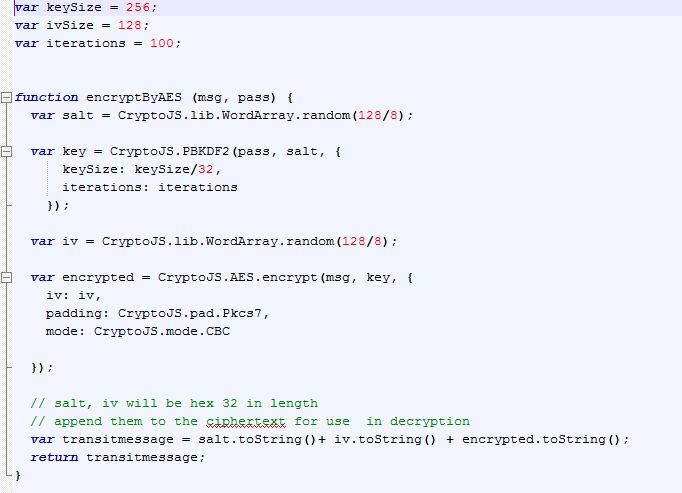
1. Kelebihan Dan Kekurangan

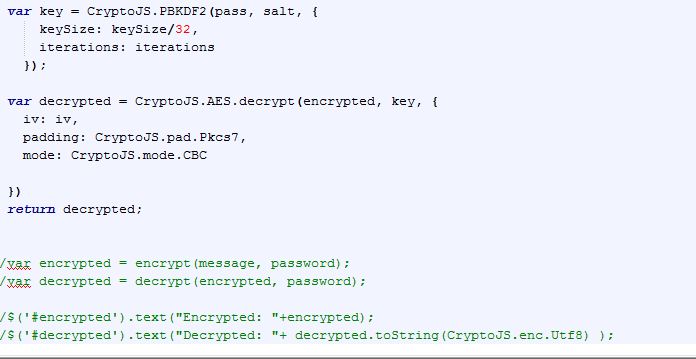
* Kelebihan

Kelebihan menggunakan struktur SPN (Substitution Permutation Network) yang memiliki derajat paralelisme yang lebih besar, sehingga diharapkan lebih cepat dari pada Feistel.

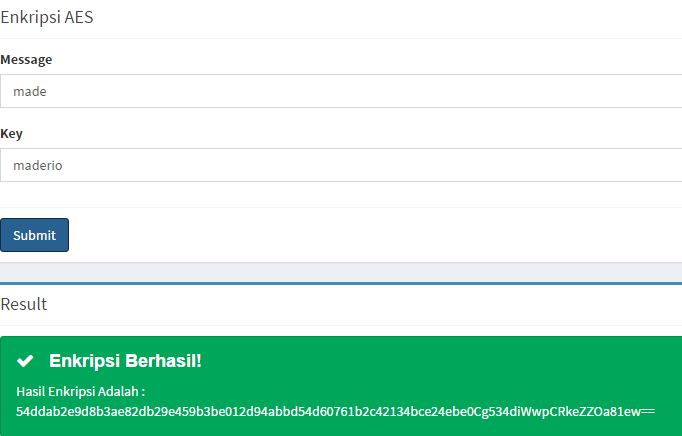
* Kekurangan

Kekurangan pada umumnya (termasuk pada Rijndael) adalah berbedanya struktur enkripsi dan dekripsi sehingga diperlukan dua algoritma yang berbeda untuk enkripsi dan dekripsi. Dan tentu pula tingkat keamanan enkripsi dan dekripsinya menjadi berbeda. AES memiliki blok masukan dan keluaran serta kunci 128 bit.

1. Source Code RSA dan AES



Hasil running :



1. Pengertila Triple DES

Menurut Hidayat, 2010 Triple DES adalah sebuah cipher blok yang dibentuk oleh DES dengan menggunakannya tiga kali. Triple DES atau TDES atau 3DES menggunakan DES 3 kali. Penggunaan tiga langkah ini penting untuk mencegah meet – in – middle attack sebagaimana pada Double DES.

Bentuk sederhana dari Triple DES adalah :

Enkripsi : C = Ek3 (Ek2(Ek1(P)))

Dekripsi : P = Dk1 (Dk2(Dk3(C)))

Perlu diingat bahwa DES bukanlah sebuah grup (dalam matematika), karena jika merupakan grup, pembangunan Triple DES akan ekivalen dengan operasi Single DES yang berarti tidak lagi aman Varian ini umum dikenal dengan dengan mode EEE (untuk enkripsi) karena pada proses enkripsi semuanya menggunakan enkripsi. Untuk menyederhanakan interoperability antara DES dan Triple DES, maka pada langkah di tengah (pada proses enkripsi TDES) diganti dengan dekripsi (mode EDE). Dengan pengubahan ini, maka dibuat beberapa versi TDES.

Versi pertama Triple DES menggunakan 2 buah kunci, yaitu k1 dan k2.

Enkripsi : C = Ek1 (Dk2(Ek1(P)))

Dekripsi : P = Dk1 (Ek2(Dk1(C)))

Enkripsi DES tunggal dengan kunci K dapat dinyatakan sebagai TDES – EDE with K1 = K2 = K. Gambar di bawah ini memperlihatkan versi TDES dengan 2 buah kunci. Penggunaan enkripsi pada langkah di tengah tidak mempengaruhi keamanan algoritma.



Gambar 3.15 Diagram enkripsi dan dekripsi Triple DES

Secara umum, Triple DES dengan 2 buah kunci mempunyai panjang kunci 2 x 56 bit = 112 bit, jauh lebih pendek daripada Triple DES yang mempunyai panjang kunci 3 x 56 bit = 168 bit.

1. Kriptoanalis Pada Triple DES

Kriptoanalisis pada Triple DES dapat menggunakan key search attack dan exploit known (atau chosen) pairs of plainteks dan cipherteks. Pengukuran keberhasilan dan kompleksitas dari serangan kriptoanalisis adalah sebagai berikut :

1. Jumlah pasangan known plainteks-cipherteks.
2. Storage space yang dibutuhkan untuk serangan.
3. Jumlah single encryption.
4. Jumlah keseluruhan operasi/step untuk serngan.

Cara yang paling ampuh untuk serangan Triple DES adalah dengan menggunakan MITM. Jika pasangan plainteks dan cipherteks (p,c) diberikan, dapat dilakukan proses sebagai berikut :

* Hitung semua nilai bN = D3N(c), N{0,1}k, dan simpan pasangan (bN,N) dalam tabel, dengan indeks bN.
* Hitung semua nilai bLM = E2M(E1L(p)) dengan L,M {0,1}k, dan cari (bL,M,N) di tabel pasangan (bN,N) yang telah dihitung sebelumnya.
* Tes seluruh triple (L,M,N) dengan bLM = bN sampai hanya tinggal satu dri triple yang ada.

1. Kelebihan dan Kekurangan Triple DES

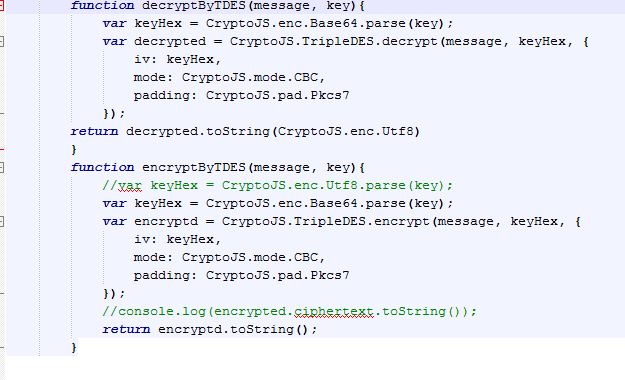
* Kelebihan Triple DES

Kelebihan yang dimiliki oleh Triple DES adalah membuat menjebol enkripsinya sangat lama.

* Kekurangan Triple DES

Triple DES memiliki kekurangan pada performansi yang lambat dalam software.Kelebihan yang dimiliki oleh Triple DES adalah membuat menjebol enkripsinya sangat lama.

1. Source Code Triple DES



1. Pengertian Tanda Tangan Digital (Digital Signature)

Tanda tangan digital adalah mekanisme otentikasi yang mengijinkan pemilik pesan membubuhkan sebuah sandi pada pesannya yang bertindak sebagai tanda tangan. Tanda tangan dibentuk dengan mengambil nilai hash dari pesan dan mengenkripsi nilai hash pesan tersebut dengan kunci privat pemilik pesan (Stallings, 2005). Prinsip yang digunakan dalam tanda tangan digital ini adalah dokumen yang dikirimkan harus ditandatangani oleh pengirim dan tanda tangan bisa diperiksa oleh penerima untuk memastikan keaslian dokumen yang dikirimkan. Fungsinya adalah untuk melakukan validasi terhadap data yang dikirim. Tanda tangan digital menggunakan algoritma yang disebut dengan istilah hashing algorithm. Fungsi tersebut akan menghasilkan sebuah kombinasi karakter yang yang unik yang disebut Message Digest. dengan cara ini pengirim bertanggungjawab terhadap isi dokumen dan dapat di cek keaslian dokumen oleh penerima. Keunikannya adalah jika di tengah perjalanan data mengalami modifikasi, penghapusan maupun di sadap diam-diam oleh hacker walaupun hanya 1 karakter saja, maka message digest yang berada pada si penerima akan berbeda dengan yang dikirimkan pada awalnya. Keunikan lainnya adalah message digest tersebut tidak bisa dikembalikan lagi ke dalam bentuk awal seperti sebelum disentuh dengan fungsi algoritma, sehingga disebutlah sebagai one-way hash (Sianturi, 2008). Fungsi utama dari tanda tangan digital pada pada aspek keamanan kriptografi adalah non-repudiation atau anti penyangkalan dimana apabila dokumen valid maka pengirim tidak bisa menyangkal bahwa keberadaan dokumen benar dikirim oleh pengirim yang bersangkutan. Suatu tanda tangan digital dapat digunakan di segala macam pesan, apakah itu terenkripsi maupun tidak, sehingga penerima dapat memastikan identitas pengirim itu dan pesan tiba secara utuh.

**Keunggulan Digital Signature**

* Memberikan keaslian, integritas dan ketidaktaatan pada dokumen elektronik
* Untuk menggunakan Internet sebagai media yang aman dan aman untuk e-Governance dan e-Commerce

**Sifat – sifat yang dimiliki tanda tangan digital**

adalah sebagai berikut :

1. Otentikasi

Otentik berarti tidak bisa bahkan sulit untuk ditiru oleh orang lain.Pesan dan tanda tangan pesan tersebut juga dapat menjadi barang bukti sehingga penanda tangan takbisa menyangkal bahwa dulu ia tidak pernah menandatanganinya . Meskipun pesan seringkali dapat mencakup informasi tentang entitas mengirim pesan, bahwa informasi mungkin tidak akurat. Tanda tangan digital dapat digunakan untuk otentikasi sumber pesan. Ketika kepemilikan kunci rahasia tanda tangan digital terikat kepada pengguna tertentu, tanda tangan yang sah menunjukkan bahwa pesan yang dikirim oleh pengguna tersebut. Pentingnya kepercayaan yang tinggi dalam otentisitas pengirim ini terutama jelas dalam konteks keuangan. Misalnya, kantor cabang bank mengirimkan instruksi ke kantor pusat meminta perubahan saldo account. Apabila kantor pusat tidak yakin bahwa pesan tersebut benar-benar dikirim dari sumber resmi, bertindak atas permintaan semacam itu bisa menjadi kesalahan besar.

1. Integritas

Dalam skenario banyak, pengirim dan penerima pesan mungkin memiliki kebutuhan untuk keyakinan bahwa pesan belum diubah selama transmisi. Meskipun menyembunyikan enkripsi isi pesan, dimungkinkan untuk mengubah sebuah pesan terenkripsi tanpa memahaminya. (Algoritma enkripsi Beberapa, yang dikenal sebagai nonmalleable yang, mencegah hal ini, tetapi yang lain tidak.) Namun, jika pesan secara digital ditandatangani, setiap perubahan dalam pesan setelah tanda tangan akan membatalkan tanda tangannya. Selain itu, tidak ada cara yang efisien untuk memodifikasi pesan dan tanda tangan untuk menghasilkan pesan baru dengan tanda tangan yang sah, karena ini masih dianggap layak oleh sebagian besar komputasi fungsi hash kriptografi (lihat resistensi tabrakan ).

1. Non-repudiation

Non-repudiation , atau lebih khusus non-repudiation asal, merupakan aspek penting dari tanda tangan digital. Dengan properti ini suatu entitas yang telah menandatangani beberapa informasi tidak dapat di lain waktu menyangkal memiliki menandatanganinya. Demikian pula, akses ke kunci publik hanya tidak memungkinkan pihak penipuan untuk palsu tanda tangan valid.

1. Sejarah Digital Signature

Pada tahun 1976, Whitfield Diffie dan Martin Hellman pertama kali menggambarkan gagasan tentang skema tanda tangan digital, walaupun mereka hanya menduga bahwa skema semacam itu ada berdasarkan fungsi peralihan satu arah jalan jebakan. Segera setelah itu, Ronald Rivest, Adi Shamir, dan Len Adleman menemukan algoritma RSA, yang dapat digunakan untuk menghasilkan tanda tangan digital primitif (walaupun hanya sebagai bukti-konsep - - tanda tangan RSA "polos" tidak aman). Paket perangkat lunak pertama yang dipasarkan untuk menawarkan tanda tangan digital adalah Lotus Notes 1.0, dirilis pada tahun 1989, yang menggunakan algoritma RSA. Skema tanda tangan digital lainnya segera dikembangkan setelah RSA, tanda tangan Lamport yang paling awal, tanda tangan Merkle (juga dikenal sebagai "pohon Merkle" atau hanya "pohon Hash"), dan tanda tangan Rabin.

Sedangkan pada tahun 1984, Shafi Goldwasser, Silvio Micali, dan Ronald Rivest menjadi orang pertama yang secara ketat menentukan persyaratan keamanan skema tanda tangan digital. Mereka menggambarkan sebuah hirarki model serangan untuk skema tanda tangan, dan juga mempresentasikan skema tanda tangan GMR, yang pertama yang dapat terbukti mencegah pemalsuan eksistensial terhadap serangan pesan terpilih yang merupakan definisi keamanan yang berlaku saat ini untuk skema tanda tangan. Skema pertama yang tidak dibangun pada fungsi trapdoor melainkan pada keluarga fungsi dengan properti permutasi satu arah yang jauh lebih lemah disajikan oleh Moni Naor dan Moti Yung.

1. Cara kerja dari digital signature

Dimulai dari pencetus pesan menggunakan tombol tanda tangan (Kunci Pribadi) untuk menandatangani pesan dan mengirim pesan dan tanda tangan digital ke penerima. Penerima menggunakan kunci verifikasi (Kunci Publik) untuk memverifikasi asal pesan dan pesan tersebut tidak dirusak saat dalam perjalanan.

Tanda tangan digital menggunakan tipe Asymmetric Cryptography. Skema biasanya terdiri dari tiga Algoritma yakni:

1. A key generation algorithm, yang memilih kunci pribadi secara seragam secara acak dari sekumpulan kunci pribadi yang mungkin. Algoritma mengeluarkan kunci privat dan kunci publik yang sesuai.
2. A signing algorithm, yang mana jika diberi pesan dan kunci privat, menghasilkan tanda tangan.
3. A signature verifying algorithm, yang mana ketika diberi pesan, kunci publik dan tanda tangan, akan menerima atau menolak klaim pesan tersebut atas keasliannya.

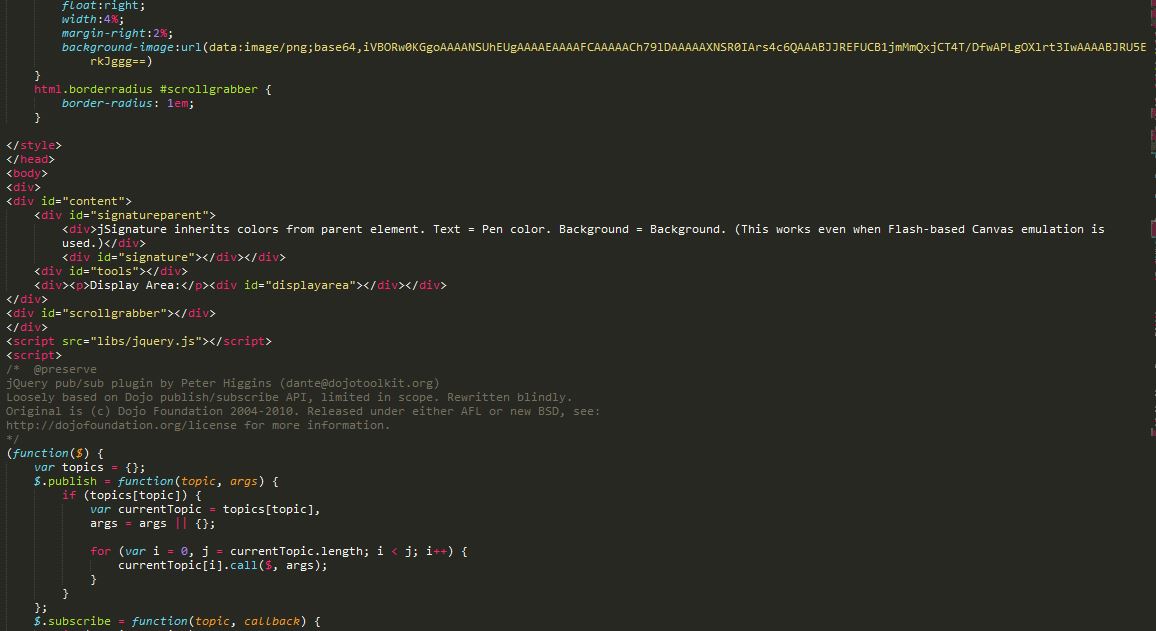
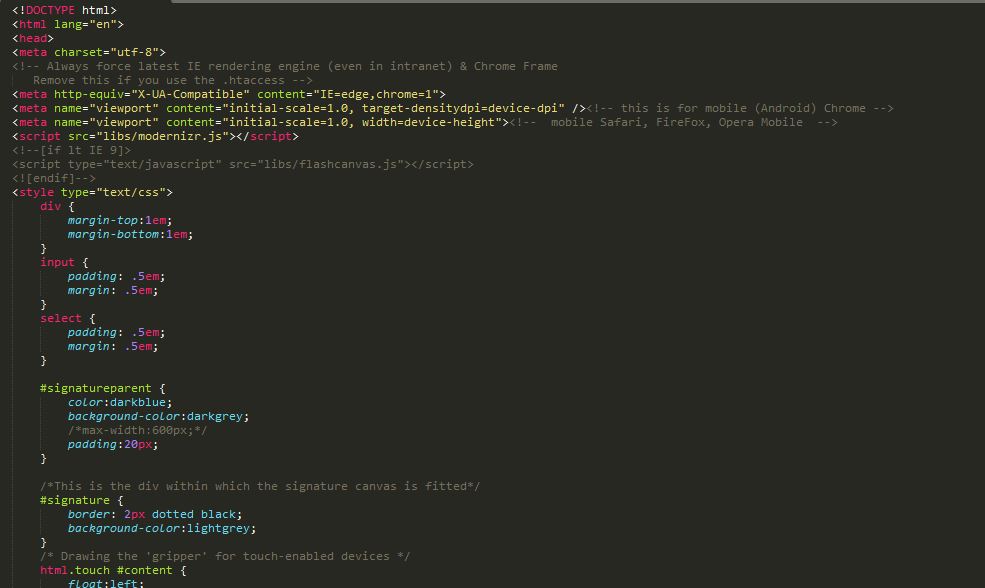
Cara kerja Digital Signature dengan memanfaatkan dua buah kunci, yaitu kunci publik dan kunci privat. Kunci publik digunakan untuk mengenkripsi data, sedangkan kunci privat digunakan untuk mendekripsi data. Pertama, dokumen di-hash dan menghasilkan Message Digest. Kemudian, Message Digest dienkripsi oleh kunci publik menjadi Digital Signature. Untuk membuka Digital Signature tersebut diperlukan kunci privat. Bila data telah diubah oleh pihak luar, maka Digital Signature juga ikut berubah sehingga kunci privat yang ada tidak akan bisa membukanya. Ini merupakan salah satu syarat keaman jaringan, yaitu Authenticity. Artinya adalah, keaslian data dapat terjamin dari perubahan-perubahan yang dilakukan pihak luar. Dengan cara yang sama, pengirim data tidak dapat menyangkal data yang telah dikirimkannya. Bila Digital Signature cocok dengan kunci privat yang dipegang oleh penerima data, maka dapat dipastikan bahwa pengirim adalah pemegang kunci privat yang sama. Ini berarti Digital Signature memenuhi salah satu syarat keamanan jaringan, yaitu Nonrepudiation atau non-penyangkalan.

1. Penggunaan Tanda Tangan Digital

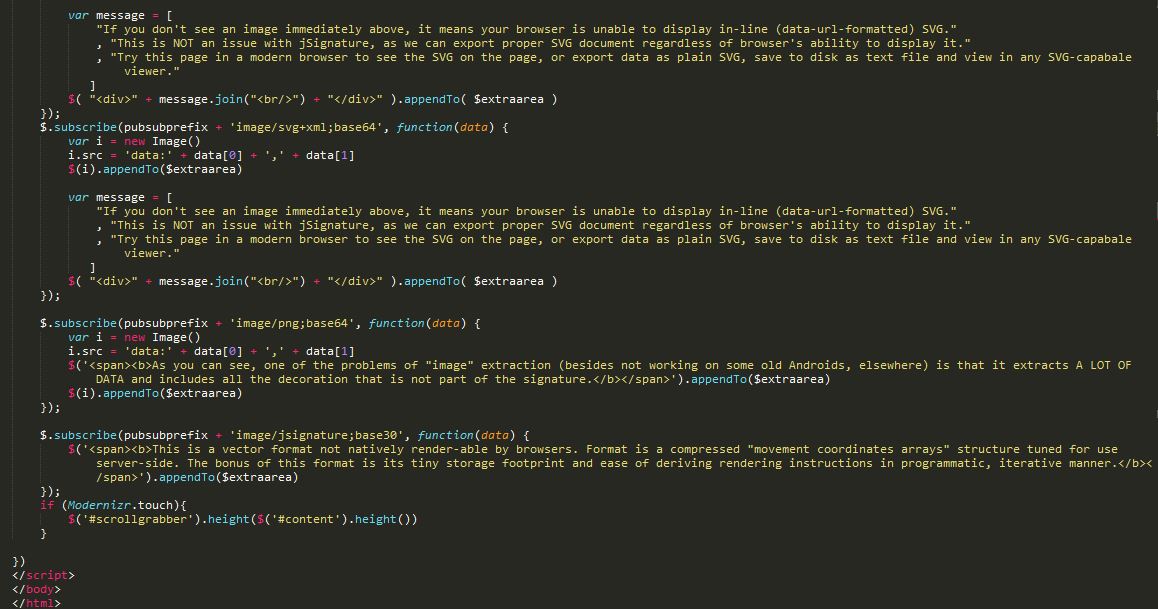
Salah satu cara yang digunakan untuk memastikan surat tersebut adalah dengan mengecek tanda tangan yang ada di dalam surat tersebut dan stempel yang menunjukkan keaslian pengirim surat. Tanda tangan digital atau yang lebih dikenal dengan digital signature mempunyai fungsi yang sama dengan tanda tangan analog yang ditulis di atas kertas. Tanda tangan digital harus unik sehingga dapat membedakanpengirim yang satu degan yang lainnya. Tanda tangan digital juga harus sulit untuk ditiru dan dipalsukan sehingga integritas dan keabsahan pesan dapat terjaga. Dengan demikian diharapkan pencatutan identitas ketika pesan atau email tersebut dikirim dapat dihindari. Tidak hanya pencatutan Untuk keperluan yang penting ini, tersedia alat bantu yang dapat diperoleh secara cumacuma, yakni Pretty Good Privacy (PGP) dan Gnu Privacy Guard atau GPG. Tentu saja masih terdapat penyedia layanan tanda tangan digital lainnya, namun PGP dan GPG lebih dikenal luas. GPG adalah produk Open Source yang dapat diperoleh secara gratis tanpa harus membayar lisensi. Penggunaaan PGP di luar

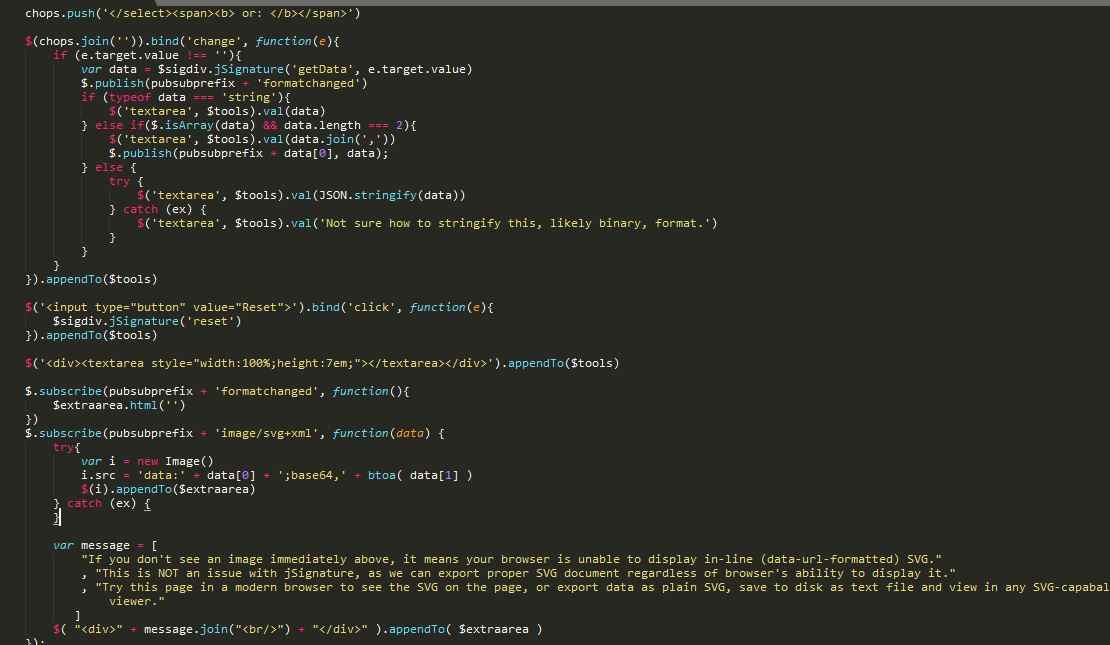
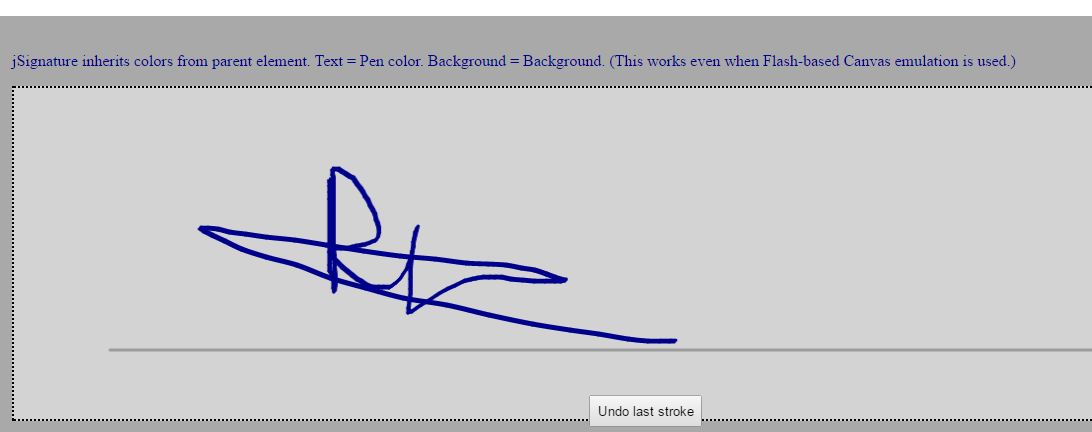
Amerika Serikat harus menggunakan versi internasional. Sedangkan GPG sendiri karena dikembangkan di luar wilayah hukum Amerika Serikat, maka bebas digunakan oleh siapapun. Restriksi ini berkaitan dengan aturan ekspor produk enkripsi yang berkait dengan pemakaian kunci sandi untuk pemakaian tanda tangan digital ini [DIR04]. Penggunaan tanda tangan digital ini tidak terlalu sulit. Kedua belah pihak yang akan berkomunikasi harus menyiapkan sepasang kunci, yaitu kunci privat (private key) dan kunci publik (public key). Kunci privat hanya dipegang oleh pemiliknya sendiri. Sedangkan kunci publik dapat diberikan kepada siapapun yang memerlukannya.

1. Source Code digital Signature









**BAB IV**

**PENUTUP**

1. Kesimpulan

Kriptografi merupakan salah satu dari media komunikasi dan informasi kuno yang masih dimanfaatkan hingga saat ini. Kriptografi di Indonesia disebut persandian yaitu secara singkat dapat berarti seni melindungi data dan informasi dari pihak-pihak yang tidak dikehendaki baik saat ditransmisikan maupun saat disimpan. Sedangkan ilmu persandiannya disebut kriptologi yaitu ilmu yang mempelajari tentang bagaimana tehnik melindungi data dan informasi tersebut beserta seluruh ikutannya.Pengguna diberikan ID dan password untuk mengakses sistem yang ada. Password dienkripsi untuk mencegah terjadinya akses illegal terhadap sistem misalnya pencurian data-data penting oleh mereka yang tidak berhak. Demikian juga enkripsi pada file-file penting dapat dilakukan (misalnya file yang berisi data keuangan). Metode enkripsi yang digunakan dapat berbentuk enkripsi kunci simetris, misalnya menggunakan algoritma DES, RSA, dll. Untuk mendapatkan algoritma enkripisi ini tidak dibutuhkan biaya karena telah dipublikasikan secara umum. Oleh karena itu, dapat disimpulkan bahwakriptografi masih merupakan sistem yang efektif dalam hal keamanan dan proteksi serta dapat digunakan secara luas di berbagai bidang usaha dan teknologi.

**DAFTAR PUSTAKA**

1. Contoh algoritma kriptografi “https://github.com/NaufalHSyahputra/CryptoTools ”