Aplikasi Kriptografi pada e-KTP

Muhammd Zen 13511060
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13511060@std.stei.itb.ac.id

***Abstrak*— Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga. Kriptografi juga digunakan pada e-KTP untuk menjaga keamanan data yang tersimpan di e-KTP tersebut. Pada makalah ini akan dibahas algoritma enkripsi yang digunakan di e-KTP.**

***Kata kunci*—Kriptografi, e-KTP, enkripsi**

I. PENDAHULUAN

Berdasarkan sosialisasi e-KTP di http://www.e-ktp.com/2011/06/hello-world/ dan PERATURAN MENTERI DALAM NEGERI TENTANG PERUBAHAN ATAS PERATURAN MENTERI DALAM NEGERI NOMOR 38 TAHUN 2009 TENTANG STANDAR DAN SPESIFIKASI PERANGKAT KERAS, PERANGKAT LUNAK DAN BLANGKO KARTU TANDA PENDUDUK BERBASIS NOMOR INDUK KEPENDUDUKAN SECARA NASIONAL disebutkan bahwa e-KTP menggunakan algoritma kriptografi untuk mengamankan data pada e-KTP. Berdasarkan informasi tersebut penulis menyusun makalah ini untuk menjelaskan algoritma kriptografi yang digunakan di e-KTP.

II. E-KTP

e-KTP atau KTP Elektronik adalah dokumen kependudukan yang memuat sistem keamanan / pengendalian baik dari sisi administrasi ataupun teknologi informasi dengan berbasis pada database kependudukan nasional.

Penduduk hanya diperbolehkan memiliki 1 (satu) KTP yang tercantum Nomor Induk Kependudukan (NIK). NIK merupakan identitas tunggal setiap penduduk dan berlaku seumur hidup

Nomor NIK yang ada di e-KTP nantinya akan dijadikan dasar dalam penerbitan Paspor, Surat Izin Mengemudi (SIM), Nomor Pokok Wajib Pajak (NPWP), Polis Asuransi, Sertifikat atas Hak Tanah dan penerbitan dokumen identitas lainnya (Pasal 13 UU No. 23 Tahun 2006 tentang Adminduk)

Autentikasi Kartu Identitas (e-ID) biasanya menggunakan biometrik yaitu verifikasi dan validasi sistem melalui pengenalan karakteristik fisik atau tingkah laku manusia. Ada banyak jenis pengamanan dengan cara ini, antara lain sidik jari (fingerprint), retina mata, DNA, bentuk wajah, dan bentuk gigi. Pada e-KTP, yang digunakan adalah sidik jari.

Penggunaan sidik jari e-KTP lebih canggih dari yang selama ini telah diterapkan untuk SIM (Surat Izin Mengemudi). Sidik jari tidak sekedar dicetak dalam bentuk gambar (format jpeg) seperti di SIM, tetapi juga dapat dikenali melalui chip yang terpasang di kartu. **Data yang disimpan di kartu tersebut telah dienkripsi dengan algoritma kriptografi tertentu**. Proses pengambilan sidik jari dari penduduk sampai dapat dikenali dari chip kartu adalah sebagai berikut:

Sidik jari yang direkam dari setiap wajib KTP adalah seluruh jari (berjumlah sepuluh), tetapi yang dimasukkan datanya dalam chip hanya dua jari, yaitu jempol dan telunjuk kanan. Sidik jari dipilih sebagai autentikasi untuk e-KTP karena alasan berikut:

1. Biaya paling murah, lebih ekonomis daripada biometrik yang lain
2. Bentuk dapat dijaga tidak berubah karena gurat-gurat sidik jari akan kembali ke bentuk semula walaupun kulit tergores
3. Unik, tidak ada kemungkinan sama walaupun orang kembar

Informasi penduduk yang dicantumkan dalam e-KTP ditunjukkan pada layout kasar berikut:

Untuk mendapatkan informasi di atas dari penduduk, wajib KTP harus mengisi formulir tipe F1.01.

Selain tujuan yang hendak dicapai, manfaat e-KTP diharapkan dapat dirasakan sebagai berikut:

1. Identitas jati diri tunggal
2. Tidak dapat dipalsukan
3. Tidak dapat digandakan
4. Dapat dipakai sebagai kartu suara dalam pemilu atau pilkada

Struktur e-KTP terdiri dari sembilan layer yang akan meningkatkan pengamanan dari KTP konvensional. Chip ditanam di antara plastik putih dan transparan pada dua layer teratas (dilihat dari depan). Chip ini memiliki antena didalamnya yang akan mengeluarkan gelombang jika digesek. Gelombang inilah yang akan dikenali oleh alat pendeteksi e-KTP sehingga dapat diketahui apakah KTP tersebut berada di tangan orang yang benar atau tidak. Untuk menciptakan e-KTP dengan sembilan layer,

tahap pembuatannya cukup banyak, diantaranya:

1. Hole punching, yaitu melubangi kartu sebagai tempat meletakkan chip
2. Pick and pressure, yaitu menempatkan chip di kartu
3. Implanter, yaitu pemasangan antenna (pola melingkar berulang menyerupai spiral)
4. Printing,yaitu pencetakan kartu
5. Spot welding, yaitu pengepresan kartu dengan aliran listrik
6. Laminating, yaitu penutupan kartu dengan plastik pengaman

e-KTP dilindungi dengan keamanan pencetakan seperti relief text, microtext, filter image, invisible ink dan warna yang berpendar di bawah sinar ultra violet serta anti copy design.

Penyimpanan data di dalam chip sesuai dengan standar internasional NISTIR 7123 dan Machine Readable Travel Documents ICAO 9303 serta EU Passport Specification 2006. Bentuk KTP elektronik sesuai dengan ISO 7810 dengan form factor ukuran kartu kredit yaitu 53,98 mm x 85,60 mm.

III. PERATURAN MENTERI DALAM NEGERI NOMOR 6 TAHUN 2011 TENTANG PERUBAHAN ATAS PERATURAN MENTERI DALAM NEGERI NOMOR 38 TAHUN 2009 TENTANG STANDAR DAN SPESIFIKASI PERANGKAT KERAS, PERANGKAT LUNAK DAN BLANGKO KARTU TANDA PENDUDUK BERBASIS NOMOR INDUK KEPENDUDUKAN SECARA NASIONAL

- Bagian A poin 1.a.4c

Tanda tangan elektronik (Digital Signature) berdasarkan standar Elliptic Curve Digital Signature Algorithm paling rendah 256 bit atau RSA 2048 bit dan Hash Algorithm SHA-256

- Bagian A poi 1.d.4

Algoritma Keamanan (Security Algorithm) bersifat simetris (symmetric) berdasarkan algoritma: 3DES dengan panjang kunci 168 bit, AES 128 bit, atau setara;

IV. KRIPTOGRAFI

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga. Menurut Bruce Scheiner dalam bukunya "Applied Cryptography", kriptografi adalah ilmu pengetahuan dan seni menjaga message-message agar tetap aman (secure).

Konsep kriptografi sendiri telah lama digunakan oleh manusia misalnya pada peradaban Mesir dan Romawi walau masih sangat sederhana. Prinsip-prinsip yang mendasari kriptografi yakni:

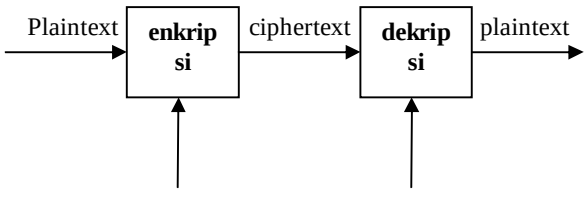
- Confidelity (kerahasiaan) yaitu layanan agar isi pesan yang dikirimkan tetap rahasia dan tidak diketahui oleh pihak lain (kecuali pihak pengirim, pihak penerima / pihak-pihak memiliki ijin). Umumnya hal ini dilakukan dengan cara membuat suatu algoritma matematis yang mampu mengubah data hingga menjadi sulit untuk dibaca dan dipahami.
- Data integrity (keutuhan data) yaitu layanan yang mampu mengenali/mendeteksi adanya manipulasi (penghapusan, pengubahan atau penambahan) data yang tidak sah (oleh pihak lain).
- Authentication (keotentikan) yaitu layanan yang berhubungan dengan identifikasi. Baik otentikasi pihak-pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian data/informasi.
- Non-repudiation (anti-penyangkalan) yaitu layanan yang dapat mencegah suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya (menyangkal bahwa pesan tersebut berasal dirinya).

Berbeda dengan kriptografi klasik yang menitikberatkan kekuatan pada kerahasiaan algoritma yang digunakan (yang artinya apabila algoritma yang digunakan telah diketahui maka pesan sudah jelas "bocor" dan dapat diketahui isinya oleh siapa saja yang mengetahui algoritma tersebut), kriptografi modern lebih menitikberatkan pada kerahasiaan kunci yang digunakan pada algoritma tersebut (oleh pemakainya) sehingga algoritma tersebut dapat saja disebarkan ke kalangan masyarakat tanpa takut kehilangan kerahasiaan bagi para pemakainya.

Berikut adalah istilah-istilah yang digunakan dalam bidang kriptografi :

- Plaintext (M) adalah pesan yang hendak dikirimkan (berisi data asli).
- Ciphertext (C) adalah pesan ter-enkrip (tersandi) yang merupakan hasil enkripsi.
- Enkripsi (fungsi E) adalah proses pengubahan plaintext menjadi ciphertext.
- Dekripsi (fungsi D) adalah kebalikan dari enkripsi yakni mengubah ciphertext menjadi plaintext, sehingga berupa data awal/asli.
- Kunci adalah suatu bilangan yang dirahasiakan yang digunakan dalam proses enkripsi dan dekripsi.

Kriptografi itu sendiri terdiri dari dua proses utama yakni proses enkripsi dan proses dekripsi. Seperti yang telah dijelaskan di atas, proses enkripsi mengubah plaintext menjadi ciphertext (dengan menggunakan kunci tertentu) sehingga isi informasi pada pesan tersebut sukar dimengerti.



Gambar 4.1 Diagram proses enkripsi dan dekripsi

Peranan kunci sangatlah penting dalam proses enkripsi dan dekripsi (disamping pula algoritma yang digunakan) sehingga kerahasiaannya sangatlah penting, apabila kerahasiaannya terbongkar, maka isi dari pesan dapat diketahui.

Secara matematis, proses enkripsi merupakan pengoperasian fungsi E (enkripsi) menggunakan e (kunci enkripsi) pada M (plaintext) sehingga dihasilkan C (ciphertext), notasinya :

Ee(M) – C

Sedangkan untuk proses dekripsi, merupakan pengoperasian fungsi D (dekripsi) menggunakan d (kunci dekripsi) pada C (ciphertext) sehingga dihasilkan M (plaintext), notasinya :

Dd(C) = M

Sehingga dari dua hubungan diatas berlaku :

Dd(Ee(M)) = M

Berikut ini adalah penjelasan algoritma kriptografi yang digunakan pada e-KTP.

1. Elliptic Curve Digital Signature

Eliliptic Curve Digital Signature Algorithm (ECDSA) adalah kurva elliptic analog dari Digital signature Algorithm (DSA). Tidak seperti permasalahan logaritma diskrit dan permasalahan faktorisasi integer, algoritma subexponensial diketahui untuk permasalahan kurva logarima diskrit. Untuk alasan ini, strength-per-key-bit adalah subtansial terbesar dalam sebuah algoritma yang menggunakan kurva elliptic.

Skematik ECDSA

Skematik ECDSA adalah part penghitung digital untuk menulis penandaan. Sebuah penandaan digital adalah sebuah bilangan bebas pada beberapa rahasia yang hanya diketahui oleh penandaan dan tambahannya, pada sesuatu dari pesan yang ditandai. Penandaan ini akan menjadi pembuktian.

Paparan ini dipersoalkan pada penandaan digital Asymetrik dengan sebuah tambahan. “Asymetrik” berarti bahwa kesatuan memiliki sepasang kunci dari kunci pribadi dan panghubung kunci umum. Kesatuan melindungi dari kunci pribadi yang digunakan untuk penandaan pesan, dan membuat tiruan yang tepat dari kunci umum untuk kesatuan yang lain yang menggunakan ini untuk menetapkan penandaan. “Tambahan” berarti sebuah cryptogrhaphic memiliki fungsi yang digunakan untuk menimbulkan sebuah intisari pesan dari suatu pesan, dan menandakan transformasi digunakan pada intisari pesan lebih dari pesan itu sendiri.

Skematik penandaan digital paenggunaan hari ini dapat diklasifikasikan berdasarkan pada permasalahan matematika yang menyediakan dasar untuk keamanan.

a. Integer Faktorisasi (IF). Yang dasar keamanan mereka pada intrektabiliti dari permasalahan faktorisasi integer.

b. Diskrit logaritma (DL) yang dasar keamanannya dititik beratkan pada permasalahan logaritma dalam sebuah daerah terbatas.

c. Skema Kurva Elliptik. Yang dasar keamanan dititik beratkan pada permasalahan curva elliptic logaritma diskrit.

Parameter-parameter domain ECDSA

Parameter-parameter domain untuk ECDSA terdiri dari sebuah pilihan yang sesuai dengan kurva eliptik E yang didefinisikan melebihi bidang tak hingga Fq dari karakteristik p, dan sebuah titik dasar (base point) G E(Fq). Parameter-parameter domain basa saja bersama-sama dengan sebuah grup , atau khusus untuk seorang pemakai tunggal. Untuk menyimpulkan, parameter-parameter domain terdiri dari :

a. Sebuah bidang ukuran q, dimana salah satu q = p , sebuah prime ganjil, atau q = 2m ;

b. Sebuah indikasi FR (field representation) dari representasi yang digunakan untuk elemen dari Fq ;

c. (opsional) sebuah string bit seedE dari ukuran minimal 160 bit , jika kurva eliptik dihasilkan sesuai dengan metode yang dijelasksan pada sect.5.2;

d. dua elemen bidang xg dan yg dalam Fq yang mendefinisikan sebuah titik tak hingg G = (xg , yg) dari orde primadalam E(Fq);

e. orde n dari titik G, dengan n >2160 dan n > 4 q ; dan

f. kofaktor h = #E(Fq)/n.

Proses ECDSA

Dalam protokol ECDSA, pihak yang akan melakukan tanda tangandigital, mempunyai parameter domain kurva eliptik berupa D={q,FR,a,b,G,n,h} dan pasangan kunci kunci rahasia dA dan kunci public QA. Kemudian pihak yang akan melakukan verifikasi terhadap tanda tangan, memiliki salinan dokumen D yang otentik dan kunci publik QA.

Generasi kunci (Key Generation)

a. Memilih sebuah bilangan bulat random dA, yang nilainyadiantara [1,n-1]

b. Menghitung QA = dA • G = (x1,y1)

c. Kunci rahasia = dA, dan kunci publik = QA.

Penandaan (Signing)

a. Memilih sebuah bilangan bulat random k, yang nilainya diantara [1,n-1].

b. Menghitung QA = k • G = (x1,y1) dan r = x1 modn, jika r= 0, maka kembali ke langkah 1.

c. Kunci rahasia = dA, dan kunci publik = QA.

d. Menghitung k-1 mod n

e. Menghitung e = Hash(m)

f. Menghitung s = k-1 {e+dA • r} mod nAsd

g. tanda tangan Author untuk message (m) adalah (r,s)

Verifikasi (Verifying)

a. Memverifikasi bahwa r dan s adalah bilangan bulat yang antara [1,n-1]
b. Menghitung e = Hash (m)
c. Menghitung w = s-1 mod n
d. Menghitung u1 = ew mod n dan u2 = rw mod n
e. Menghitung u1 • G + u2 • QA = (x1,y1)
f. Menghitung v = x1 mod n
g. Menerima tanda tangan jika dan hanya jika v = r

2. RSA

RSA di bidang kriptografi adalah sebuah algoritma pada enkripsi public key. RSA merupakan algoritma pertama yang cocok untuk digital signature seperti halnya ekripsi, dan salah satu yang paling maju dalam bidang kriptografi public key. RSA masih digunakan secara luas dalam protokol electronic commerce, dan dipercaya dalam mengamnkan dengan menggunakan kunci yang cukup panjang.

Cara kerja

Pembangkitan Kunci

Semisal Alice ingin Bob mengirimkan kepadanya sebuah pesan pribadi (private message) melalui media transmisi yang tidak aman (insecure). Alice melakukan langkah-langkah berikut untuk membuat pasangan kunci public key dan private key:

- Pilih dua bilangan prima p ≠ q secara acak dan terpisah untuk tiap-tiap p dan q. Hitung N = p q. N hasil perkalian dari p dikalikan dengan q.
- Hitung φ = (p-1)(q-1).
- Pilih bilangan bulat (integer) antara satu dan φ (1 < e < φ) yang juga merupakan coprime dari φ.
- Hitung d hingga d e ≡ 1 (mod φ).

- bilangan prima dapat diuji probabilitasnya menggunakan Fermat's little theorem- a^(n-1) mod n = 1 jika n adalah bilangan prima, diuji dengan beberapa nilai a menghasilkan kemungkinan yang tinggi bahwa n ialah bilangan prima. Carmichael numbers (angka-angka Carmichael) dapat melalui pengujian dari seluruh a, tetapi hal ini sangatlah langka.
- langkah 3 dan 4 dapat dihasilkan dengan algoritma extended Euclidean; lihat juga aritmetika modular.
- langkah 4 dapat dihasilkan dengan menemukan integer x sehingga d = (x(p-1)(q-1) + 1)/e menghasilkan bilangan bulat, kemudian menggunakan nilai dari d (mod (p-1)(q-1));
- langkah 2 PKCS#1 v2.1 menggunakan &lamda; = lcm(p-1, q-1) selain daripada φ = (p-1)(q-1)).

Pada public key terdiri atas:

- N, modulus yang digunakan.
- e, eksponen publik (sering juga disebut eksponen enkripsi).

Pada private key terdiri atas:

- N, modulus yang digunakan, digunakan pula pada public key.
- d, eksponen pribadi (sering juga disebut eksponen dekripsi), yang harus dijaga kerahasiaannya.

Biasanya, berbeda dari bentuk private key (termasuk parameter CRT):

- p dan q, bilangan prima dari pembangkitan kunci.
- d mod (p-1) dan d mod (q-1) (dikenal sebagai dmp1 dan dmq1).
- (1/q) mod p (dikenal sebagai iqmp).

Bentuk ini membuat proses dekripsi lebih cepat dan signing menggunakan Chinese Remainder Theorem (CRT). Dalam bentuk ini, seluruh bagian dari private key harus dijaga kerahasiaannya.

Alice mengirimkan public key kepada Bob, dan tetap merahasiakan private key yang digunakan. p dan q sangat sensitif dikarenakan merupakan faktorial dari N, dan membuat perhitungan dari d menghasilkan e. Jika p dan q tidak disimpan dalam bentuk CRT dari private key, maka p dan q telah terhapus bersama nilai-nilai lain dari proses pembangkitan kunci.

Proses enkripsi pesan

Misalkan Bob ingin mengirim pesan m ke Alice. Bob mengubah m menjadi angka n < N, menggunakan protokol yang sebelumnya telah disepakati dan dikenal sebagai padding scheme.

Maka Bob memiliki n dan mengetahui N dan e, yang telah diumumkan oleh Alice. Bob kemudian menghitung ciphertext c yang terkait pada n:

$$c = n^e \mod N$$

Perhitungan tersebut dapat diselesaikan dengan cepat menggunakan metode exponentiation by squaring. Bob kemudian mengirimkan c kepada Alice.

Proses dekripsi pesan

Alice menerima c dari Bob, dan mengetahui private key yang digunakan oleh Alice sendiri. Alice kemudian memulihkan n dari c dengan langkah-langkah berikut:

$$n = c^d \mod N$$

Perhitungan di atas akan menghasilkan n, dengan begitu Alice dapat mengembalikan pesan semula m. Prosedur dekripsi bekerja karena

$$c^d \equiv (n^e)^d \equiv n^{ed} \pmod N.$$

.Kemudian, dikarenakan ed ≡ 1 (mod p-1) dan ed ≡ 1 (mod q-1), hasil dari Fermat's little theorem.

$$n^{ed} \equiv n \pmod p$$

dan

$$n^{ed} \equiv n \pmod q$$

Dikarenakan p dan q merupakan bilangan prima yang berbeda, mengaplikasikan Chinese remainder theorem akan menghasilkan dua macam kongruen

$$n^{ed} \equiv n \pmod{pq}$$

serta

$$c^d \equiv n \pmod N.$$

Contoh proses

Berikut ini merupakan contoh dari enkripsi RSA dan dekripsinya. Parameter yang digunakan disini berupa bilangan kecil.

Kita membuat

p = 61 — bilangan prima pertama (harus dijaga kerahasiannya atau dihapus secara hati-hati)

q = 53 — bilangan prima kedua (harus dijaga kerahasiannya atau dihapus secara hati-hati)

N = pq = 3233 — modulus (diberikan kepada publik)

e = 17— eksponen publik (diberikan kepada publik)

d = 2753 — eksponen pribadi (dijaga kerahasiannya)

Public key yang digunakan adalah (e,N). Private key yang digunakan adalah d. Fungsi pada enkripsi ialah:

encrypt(n) = ne mod N = n17 mod 3233

dimana n adalah plaintext Fungsi dekripsi ialah:

decrypt(c) = cd mod N = c2753 mod 3233

dimana c adalah ciphertext

Untuk melakukan enkripsi plaintext bernilai "123", perhitungan yang dilakukan

encrypt(123) = 12317 mod 3233 = 855

Untuk melakukan dekripsi ciphertext bernilai "855" perhitungan yang dilakukan

decrypt(855) = 8552753 mod 3233 = 123

Kedua perhitungan di atas diselesaikan secara effisien menggunakan square-and-multiply algorithm pada modular exponentiation.

Padding schemes

Padding Scheme harus dibangun secara hati-hati sehingga tidak ada nilai dari m yang menyebabkan masalah keamanan. Sebagai contoh, jika kita ambil contoh sederhana dari penampilan ASCII dari m dan menggabungkan bit-bit secara bersama-sama akan menghasilkan n, kemudian pessan yang berisi ASCII tunggal karakter NUL (nilai numeris 0) akan menghasilkan n= 0, yang akan menghasilkan ciphertext 0 apapun itu nilai dari e dan N yang digunakan. Sama halnya dengan karakter ASCII tunggal SOH (nilai numeris 1) akan selalu menghasilkan chiphertext 1. Pada kenyataannya, untuk sistem yang menggunakan nilai e yang kecil, seperti 3, seluruh karakter tunggal ASCII pada pesan akan disandikan menggunakan skema yang tidak aman, dikarenakan nilai terbesar n adalah nilai 255, dan 2553 menghasilkan nilai yang lebih kecil dari modulus yang sewajarnya, maka proses dekripsi akan menjadi masalah sederhana untuk mengambil pola dasar dari ciphertext tanpa perlu menggunakan modulus N. Sebagai konsekuensinya, standar seperti PKCS didesain dengan sangat hati-hati sehingga membuat pesan asal-asalan dapat terenkripsi secara aman. Dan juga berdasar pada bagian Kecepatan, akan dijelaskan kenapa m hampir bukanlah pesan itu sendiri tetapi lebih pada message key yang dipilh secara acak.

Pengesahan pesan

RSA dapat juga digunakan untuk mengesahkan sebuah pesan. Misalkan Alice ingin mengirim pesan kepada Bob. Alice membuat sebuah hash value dari pesan tersebut, di pangkatkan dengan bilangan d dibagi N (seperti halnya pada deskripsi pesan), dan melampirkannya sebagai "tanda tangan" pada pesan tersebut. Saat Bob menerima pesan yang telah "ditandatangani", Bob memangkatkan "tanda tangan" tersebut dengan bilangan e dibagi N (seperti halnya pada enkripsi pesan), dan membandingkannya dengan nilai hasil dari hash value dengan hash value pada pesan tersebut. Jika kedua cocok, maka Bob dapat mengetahui bahwa pemilik dari pesan tersebut adalah Alice, dan pesan pun tidak pernah diubah sepanjang pengiriman.

Harap dicatat bahwa padding scheme merupakan hal yang esensial untuk mengamankan pengesahan pesan seperti halnya pada enkripsi pesan, oleh karena itu kunci yang sama tidak digunakan pada proses enkripsi dan pengesahan.

3. Hash Function

Hash function atau fungsi hash adalah suatu cara menciptakan “fingerprint” dari berbagai data masukan. Hash function akan mengganti atau mentranspose-kan data tersebut untuk menciptakan fingerprint, yang biasa disebut hash value. Hash value biasanya digambarkan sebagai suatu string pendek yang terdiri atas huruf dan angka yang terlihat random (data biner yang ditulis dalam notasi heksadesimal). Suatu hash function adalah sebuah fungsi matematika, yang mengambil sebuah panjang variabel string input, yang disebut pre-image dan mengkonversikannya ke sebuah string output dengan panjang yang tetap dan biasanya lebih kecil, yang disebut message digest5.

Hash function digunakan untuk melakukan fingerprint pada pre-image, yaitu menghasilkan sebuah nilai yang dapat menandai (mewakili) pre-image sesungguhnya. Fungsi hash satu arah (one-way hash function) adalah hash function yang bekerja satu arah, yaitu suatu hash function yang dengan mudah dapat menghitung hash value dari pre-image, tetapi sangat sukar untuk menghitung pre-image dari hash value. Sebuah fungsi hash satu arah, H(M), beroperasi pada suatu pre-image pesan M dengan panjang sembarang, dan mengembalikan nilai hash h yang memiliki panjang tetap. Dalam notasi matematika fungsi hash satu arah dapat ditulis sebagai:

h = H(M), dengan h memiliki panjang b

Ada banyak fungsi yang mampu menerima input dengan panjang sembarang dan menghasilkan output dengan panjang tetap, tetapi fungsi hash satu arah memiliki karakteristik tambahan yang membuatnya satu arah :

Diberikan M, mudah menghitung h.

Diberikan h, sulit menghitung M agar H(M) = h.

Diberikan M, sulit menemukan pesan lain, M', agar H(M) = H(M').

Dalam dunia nyata, fungsi hash satu arah dikembangkan berdasarkan ide sebuah fungsi kompresi. Fungsi satu arah ini menghasilkan nilai hash berukuran n bila diberikan input berukuran b. Input untuk fungsi kompresi adalah suatu blok pesan dan hasil blok teks sebelumnya. Sehingga hash suatu blok M, adalah

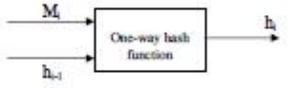
hi = f(Mi,hi-1)

dengan

hi = hash value saat ini.

Mi = blok pesan saat ini.

hi-1 = hash value blok teks sebelumnya.



Gbr. Fungsi hash satu arah

Fungsi hash sangat berguna untuk menjaga integritas sebuah data. Sudah banyak algoritma hash function yang diciptakan, namun hash function yang umum digunakan saat ini adalah MD5 dan SHA (Secure Hash Algorithm). Algoritma hash function yang baik adalah yang menghasilkan sedikit hash collision.

4. 3DES

Konsep Triple DES sebenarnya sama dengan DES, namun terdapat beberapa pengembangan, yaitu:

Bentuk umum TDES (mode EEE):

Enkripsi: C = EK3(EK2(EK1 (P)))

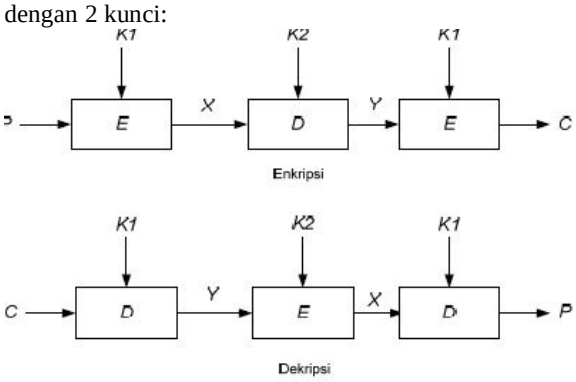
Dekripsi: P = DK1(DK2 (DK3 (C)))

Untuk menyederhanakan TDES, maka langkah di tengah diganti dengan D (mode EDE).

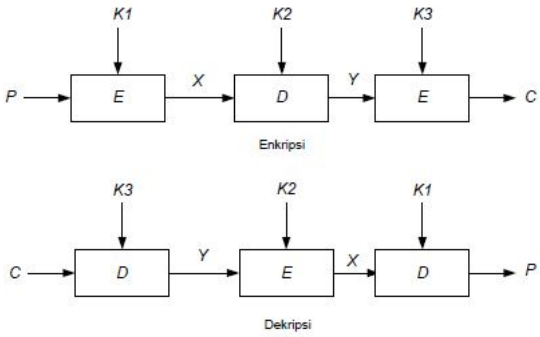
Ada dua versi TDES dengan mode EDE:

- Menggunakan 2 kunci
- Menggunakan 3 kunci

Berikut merupakan skema Triple DES dengan 2 kunci:



Dan di bawah ini skema Triple DES dengan 3 kunci:



5. AES (Advanced Encrypyion Standard)

Dalam proses enkripsi input, diperlukanlah empat macam operasi yang dilakukan berulang-ulang dalam beberapa putaran dan menggunakan kunci cipher. Jumlah putaran yang digunakan algoritma ini ada tiga macam seperti pada tabel di bawah ini.

Tipe	Panjang Kunci	Panjang Blok Input	Jumlah Putaran
AES-128	128 bit	128 bit	10
AES-192	192 bit	128 bit	12
AES-256	256 bit	128 bit	14

Tabel. Jumlah Putaran Pengoperasian AES

Operasi

Ada empat macam operasi yang dilakukan setiap putaran.

Transformasi Subtitusi Byte

Dalam operasi ini, setiap byte yang akan dienkripsi disubtitusikan dengan nilai byte lain dengan menggunakan S-box. S-box dibuat dari multiplicative inverse dari angka yang diberikan dalam Rijndael's finite field yang kemudian ditransformasikan dengan affine transformation menggunakan S-box. S-box dibuat dari multiplicative inverse dari angka yang diberikan dalam Rijndael's finite field yang kemudian ditransformasikan dengan affine transformation

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Gambar : Affine Transformation

Hasilnya kemudian di-xor dengan 9910 atau 0x6316 atau 11000112. Operasi matriks dengan xor ini ekuivalen dengan persamaan:

b'i=bi b(i+4)mod8 b(i+5)mod8 b(i+6)mod 8 b(i+7)mod8 ci

dengan b', b, dan c adalah array 8 bit dan nilai c adalah 01100011.

Proses tersebut menghasilkan masing-masing

nilai dari elemen tabel S-box yang hasilnya sebagai berikut.

Tabel 2. S-box

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Tabel. *S-box*

Seperti yang telah diketahui sebelumnya, AES merupakan algoritma simetri, yang berarti table subtitusi yang dibutuhkan untuk mengenkripsi berbeda dengan untuk mendekripsi. Untuk acuan tersebut, digunakanlah tabel S-box inversi sebagai berikut.

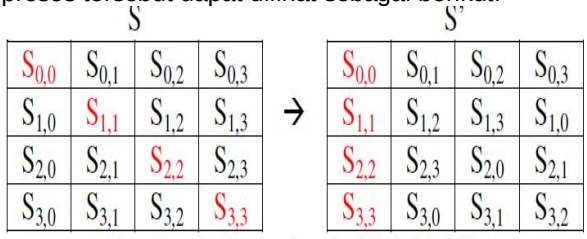
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Tabel . *S-box* Inversi

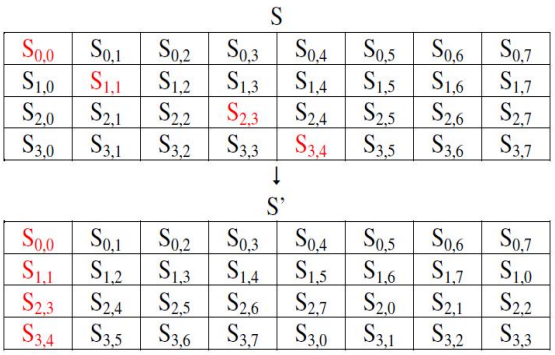
Sebagai contoh, input yang akan dienkripsikan adalah
95 95 08 19
4f 6b 5c 6e
c8 89 80 26
fc 75 4e 6c
Dengan menggunakan *S-box*, hasil dari operasi ini adalah
2a 2a 30 d4
84 7f 4a 9f
e8 a7 cd f7
b0 9d 2f 50
Jika hasil tersebut ingin dikembalikan ke nilai semula sebelum operasi, nilai-nilainya dapat disubtitusikan dengan menggunakan tabel *S-box* inversi. Operasi transformasi subtitusi *byte* pada proses enkripsi dan dekripsi tidak dilakukan pada putaran pertama. Operasi ini hanya dilakukan pada putaran kedua hingga terakhir.

Transformasi Pergeseran Baris

Pada operasi ini, *byte-byte* pada setiap baris digeser secara memutar dengan pergeseran yang berbeda dari tiap-tiap baris. Setiap baris digeser dengan aturan tertentu untuk jenis panjang blok yang berbeda. Baris pertama blok untuk semua jenis panjang blok (128, 196, dan 256 bit) tidak digeser. Baris kedua untuk semua jenis panjang blok digeser 1 ke kiri. Pergeseran baris ketiga dan keempat untuk panjang blok 128 dan 196 bit berbeda dengan 256 bit. Pada panjang blok 128 dan 196 bit, baris ketiga digeser ke kiri sebanyak dua kali dan baris keempat digeser ke kiri sebanyak tiga kali. Pada panjang blok 256 bit, baris ketiga digeser ke kiri sebanyak tiga kali dan baris keempat digeser ke kiri sebanyak empat kali [5]. Untuk lebih jelasnya, proses tersebut dapat dilihat sebagai berikut.



Gambar : Operasi pada Blok 128 bit



Gambar : Operasi pada Blok 256 bit

Sebagai contoh, hasil operasi ini terhadap input yang
nilainya adalah output dari hasil operasi subtitusi b*yte* sebelumnya adalah sebagai berikut
2a 2a 30 d4
7f 4a 9f 84
cd f7 e8 a7
50 b0 9d 2f

Transformasi Percampuran Kolom

Transformasi ini mengoperasikan blok pada masing masing kolomnya. Setiap kolom diperlakukan sebagai *four-term polynomial* dengan cara Galois Field (GF) (2_8) dan dimodulokan dengan x tetap $a(x)$ [3], yaitu
$a(x) = \{03\}x_3 + \{01\}x_2 + \{01\}x + \{02\}$

Hal ini dapat dituliskan sebagai perkalian matriks sebagai berikut.

$$s'(x) = a(x) \otimes s(x)$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

dengan *c* adalah letak kolom, sehingga hasilnya

$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$
$$s'_{0,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$
$$s'_{0,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$
$$s'_{0,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c})$$

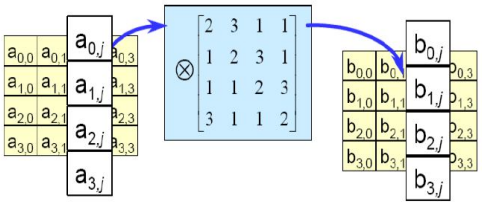
Jika hasil perkalian memiliki lebih dari 8 bit, bit yang lebih tidak begitu saja dibuang. Hasil tersebut di dengan 100011011_2 [5]. Sebagai contoh, perkalian 11001010 dengan 11 dengan GF(2^8) sebagai berikut.

11001010
11
---------- *
11001010
11001010
---------- xor
101011110
100011011
---------- xor
1000101

Nilai 1000101 merupakan hasil dari perkalian tersebut. Misalnya, jika dalam transfomasi ini input yang dipakai adalah hasil dari operasi pergeseran baris
sebelumnya, hasil yang diperoleh adalah sebagai berikut.

48 cd af ac
c8 0c ab 1a
24 5e d8 74
6c b8 06 fa

Transformasi ini dapat diilustrasikan sebagai berikut



Gambar : Ilustrasi Transformasi Percampuran kolom)

Operasi transformasi ini tidak digunakan dalam putaran terakhir, baik untuk enkripsi maupun dekripsi.

Transformasi Penambahan Kunci

Dalam operasi transformasi ini, digunakanlah upakunci untuk masing-masing putaran yang berasal dari kunci utama dengan menggunakan jadwal kunci

Rijndael (*Rijndael's key schedule)* ukuran upakunci tersebut sama dengan ukuran blok yang akan diproses. Upakunci tersebut kemudian di-xor dengan blok input sehingga diperoleh hasilnya. Sebagai contoh, jika inputnya adalah

a3 c5 08 08
78 a4 ff d3
00 ff 36 36
28 5f 01 02

dan diperoleh upakunci

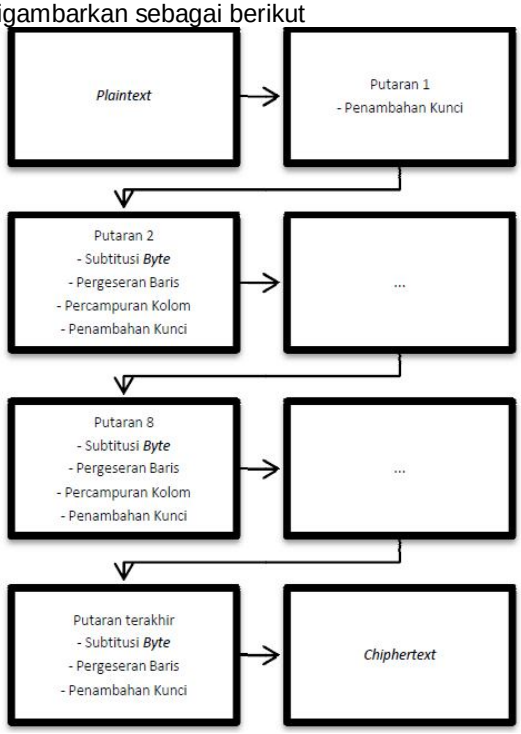
36 8a c0 f4
ed cf 76 a6
08 a3 b6 78
31 31 27 6e

Maka, hasilnya adalah

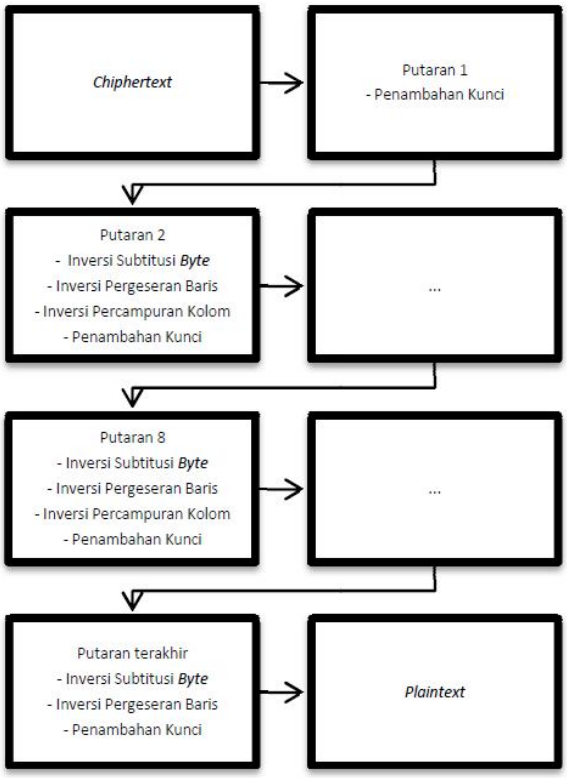
a6 34 1a 00
24 dd f1 0e
62 a8 73 cf
48 b9 5d 61

Putaran

Seperti yang telah diketahui sebelumnya pada Tabel *Affine Transformation* jumlah putaran pengoperasian blok input untuk setiap macam panjang blok berbeda-beda. Akan tetapi jumlah putaran untuk proses enkripsi dan dekripsi tetap sama. Proses enkripsi dan dekripsi dapat digambarkan sebagai berikut



Gambar: Diagram Proses Enkripsi



Gambar : Diagram Proses Dekripsi

V. KESIMPULAN

Dengan banyaknya algoritma kriptografi yang digunakan untuk mengamankan data pada e-KTP, maka dapat dikatakan bahwa e-KTP ini memiliki tingkat keamanan yang baik.

REFERENCES

[1] http://www.e-ktp.com/2011/06/hello-world/. Diakses tanggal 16 Desember 2012

[2] http://jdih.bpk.go.id/wp-content/uploads/2012/03/permen_no.6_th_20111.pdf. Diakses 16 Desember 2012.

[3] http://robby.c.staff.gunadarma.ac.id/Downloads/files/4565/KRIPTOGRAFI.doc. Diakses 16 Desember 2012

[4] http://www.library.upnvj.ac.id/pdf/s1teknikinformatika08/204511076/skripsi.pdf. Diakses 16 Desember 2012

[5] http://www.rsasecurity.com/rsalabs/node.asp?id=2125. Diakses 16 Desember 2012

[6] http://budi.insan.co.id/courses/security/2006-2007/Report-Roswan-Latuconsina.pdf. Diakses 16 Desember 2012

[7] http://www.quadibloc.com/crypto/co040401.html. Diakses 16 Desember 2012

[8] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf diakses 16 Desember 2012

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 18 Deseember 2012

Muhammad Zen
13511060