LEMBAR PENGESAHAN SKRIPSI

"APLIKASI DESKTOP ENKRIPSI DEKRIPSI VIDEO MENGGUNAKAN ALGORITMA KRIPTOGRAFI SEED 128"

OLEH:

FRIDA APRILIA FARID

D42113523

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 22 Januari 2019.
Diterima dan disahkan sebagai salah satu syarat memperoleh gelar Sarjana Teknik (ST.)
pada Program Studi S1 Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Gowa, 22 Januari 2019

Disetujui oleh:

Pembimbing I,	Pembimbing II,
Dr.Ir. Ingrid Nurtanio, M.T.	Dr.Eng. Zulkifli Tahir,S.T., M.Sc.
NIP. 19610813 198811 2 001	NIP. 19840403 201012 1 004

Diterima dan disahkan oleh:
Ketua Departemen Teknik Informatika

Dr. Amil Ahmad Ilham, S.T., M.IT.
NIP. 19731010 199802 1 001

APLIKASI DESKTOP ENKRIPSI DEKRIPSI VIDEO MENGGUNAKAN ALGORITMA KRIPTOGRAFI SEED 128

TUGAS AKHIR

Disusun dalam rangka memenuhi salah satu persyaratan

Untuk menyelesaikan program Strata-1 Prodi Informatika Jurusan Elektro

Universitas Hasanuddin

Makassar

Disusun Oleh:

FRIDA APRILIA FARID

D42113523

DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2019

ABSTRAK

Konten digital seperti data video seharusnya diproteksi tidak hanya ketika dikirimkan, tetapi juga ketika konten digital tersebut sampai kepada pemakainya. Oleh karena itu, dibutuhkan suatu mekanisme untuk mengatasi masalah keamanan data video. Salah satunya adalah dengan menggunakan teknologi kriptografi. Data video dapat dienkripsi sebelum didistribusikan atau sekedar diamankan didalam komputer. Ketika file akan dipulihkan kembali maka dilakukan dekripsi video. Pada penelitian ini digunakan algoritma kriptografi SEED 128 untuk membangun sistem enkripsi dan dekripsi data video. SEED adalah kunci simetris blok cipher 128-bit yang dikembangkan oleh *KISA (Korea Information Security Agency)* dan tim ahli sejak tahun 1998. Sistem aplikasi desktop dibangun menggunakan *Java Programming Language* dengan mengimplementasikan algoritma kriptografi SEED 128 telah berhasil melakukan enkripsi dan dekripsi untuk video. Hasil pengujian yang dilakukan untuk format MP4, MKV, AVI, FLV, WMV berhasil di enkripsi dan dekripsi. Pada video yang dilakukan proses enkripsi, terjadi perubahan ukuran karena proses algoritma yang mengubah blok bit data sehingga menjadi acak. Pada proses dekripsi blok bit data dipulihkan dan kembali seperti ukuran video awal. Durasi waktu yang diperlukan untuk memproses data video ketika enkripsi dan dekripsi tergantung pada ukuran Byte. Semakin besar ukuran video maka waktu yang diperlukan juga akan semakin lama, dan begitupun sebaliknya.

Kata Kunci : *Video, Kriptografi, Algoritma SEED 128.*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Alhamdulillahi Katsiiron, puji syukur penulis panjatkan kepada Allah SWT. yang telah melimpahkan rahmat, hidayah, kekuatan dan segala pertolongan-Nya sehingga skripsi ini dapat diselesaikan. Sholawat serta salam semoga selalu dihaturkan kepada Rasulullah Muhammad SAW, beserta keluarga dan sahabat beliau. Saya persembahkan skripsi ini untuk Ibu dan Ayah saya, semoga kalian bahagia dan bangga.

Skripsi yang berjudul "*Aplikasi Desktop Enkripsi dan Dekripsi Video Menggunakan Algoritma Kriptografi SEED 128*" yang disusun untuk memenuhi salah satu syarat guna mencapai gelar kesarjanaan Strata Satu (S1) ini tidak lepas dari bantuan serta dukungan dari berbagai pihak, pada kesempatan ini penulis mengucapkan terima kasih dan sekaligus memohon maaf atas segala kekurangan maupun kesalahan ucapan dan sikap selama ini.

Ucapan terima kasih ini disampaikan kepada :

1. Ibu dan Ayah, serta saudara/i kandung saya (Muh.Yusri Farid, Nuraini Farid, Fitriani Farid, Diana Farid, Irda Farid, dan Salma Farid) yang selalu sabar serta menjadi kekuatan untuk penulis dengan mendoakan dan memberikan dukungan, baik secara moril maupun materil, serta keluarga esar saya yang tidak bisa disebutkan satu per satu.
   u Dr.Ir.Ingrid Nurtanio,M.T., selaku dosen pembimbing I yang telah eluangkan waktu untuk memberikan bimbingan dan arahan selama

penelitian hingga penyusunan skripsi ini dan atas ilmu yang diberikan selama masa studi di Teknik Informatika, Fakultas Teknik Universitas Hasanuddin.

3. Bapak Dr.Eng.Zulkifli Tahir, S.T. M.Sc., selaku dosen pembimbing II yang telah meluangkan waktu untuk memberikan bimbingan dan arahan selama penelitian hingga penyusunan skripsi ini dan atas ilmu yang diberikan selama masa studi di Teknik Informatika, Fakultas Teknik Universitas Hasanuddin.
4. Bapak Dr.Amil Ahmad Ilham, S.T.,M.I.T., sebagai dosen penguji yang telah meluangkan waktu memberikan masukan dan kritik serta saran untuk perbaikan skripsi ini.
5. Bapak Dr.Eng.Muhammad Niswar, S.T.,M.I.T., sebagai dosen penguji yang telah meluangkan waktu memberikan masukan dan kritik serta saran untuk perbaikan skripsi ini.
6. Bapak A. Ais Prayogi Alimuddin, S.T.,M.Eng., sebagai dosen penguji yang telah meluangkan waktu memberikan masukan dan kritik serta saran untuk perbaikan skripsi ini.
7. Semua jajaran dosen pengajar Jurusan Elektro dan Program Studi Informatika atas ilmu dan arahan yang diberikan selama masa studi di Fakultas Teknik, Universitas Hasanuddin.
8. Bapak Robert dan Bapak Sainuddin sebagai staff akademik Teknik formatika yang telah memberikan banyak bantuan dibidang administrasi.

9. Group Angels (Ria Rifayanty Ruslan, Sry Rahayu Halwain, Aidha Arfani, A.Meldayasari, Maureen Magdalena Sitorus, Mu'tasimah, A.Ainun Khaerunnisyah Qodrat, Fijrah Aprilla S. Asri) yang selama ini telah mensupport, membantu, menghibur serta menyemangati penulis selama penelitian dan penyusunan skripsi.
10. AMPLIF13R saudara/i seangkatan Jurusan Elektro 2013 yang banyak memberikan support bantuan dan semangat untuk penulis.
11. Teman-teman seperjuangan Prodi Teknik Informatika yang banyak memberikan support bantuan dan semangat untuk penulis.
12. Rahmat Hidayat Slamet sebagai senior terbaik yang selalu mambantu penulis untuk membangun sistem.
13. Semua pihak yang tidak dapat saya sebutkan satu-persatu.

Ibarat tak ada gading yang tak retak dan bahwasanya segala kesempurnaan hanya milik Allah SWT. Rabb Seluruh Alam. Penulis menyadari penulisan skripsi ini masih banyak kekurangan, oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun serta kedepannya dapat dikembangkan. Semoga skripsi ini dapat membawa manfaat bagi penulis, para pembaca dan peneliti selanjutnya.

Makassar, 11 November 2018

Penyusun

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

I.1. Latar Belakang

Saat ini jumlah pengguna internet di Indonesia mengalami kenaikan setiap tahunnya. Hal ini dikuatkan oleh hasil riset lembaga riset internasional Emarketer yang memproyeksi jumlah pengguna internet di Indonesia sekitar 102,8 juta jiwa di tahun 2016 dan akan mengalami peningkatan sekitar 123 juta jiwa di tahun 2018. Dari segi infrastruktur, pembangunan infrastruktur di Indonesia mengalami perkembangan yang signifikan, sehingga masyarakat Indonesia tidak hanya bisa merasakan jaringan 3G, namun sudah bisa merasakan internet super cepat di jaringan 4G. Semakin handalnya berbagai komponen teknologi membuat semua orang menjadi mudah mengakses berbagai konten yang ada di internet. Berbagai macam web, blog, serta platform menyediakan berbagai macam konten multimedia, artikel, dan berita-berita seluruh dunia. Salah satu konten yang paling sering di akses adalah video. Video semakin jamak ditonton lewat perangkat mobile seperti smartphone (Yusuf, 2018).

Marak-maraknya video dibuat dan di upload di blog dan platform video di internet. Perkembangan bisnis konten digital telah membawa peluang bagi kejahatan klasik di bidang teknologi informasi, ancaman yang timbul terkait keamanan video adalah pembajakan, privasi, dan pelanggaran HAKI (Hak yaan Intelektual). Konten-konten yang seharusnya menjadi properti legal rodusen dan secara legal dimiliki oleh orang yang telah membelinya, bisa

dengan mudah disalahgunakan oleh pihak-pihak yang tidak bertanggungjawab. Banyak video yang seharusnya ditujukan hanya kepada kelompok orang tertentu dapat pula dilihat oleh sekelompok lain yang bukan haknya. Sebagai contoh para provider Video on Demand yang membeli lisensi untuk menyebarluaskan video. Tetapi jika file video tidak diamankan, maka orang akan bebas medownload dan memperbanyak untuk kepentingan lain. Hal ini tentu saja sangat merugikan. Konten digital seperti video seharusnya diproteksi tidak hanya ketika dikirimkan, tetapi juga ketika konten digital tersebut sampai kepada pemakainya. Oleh karena itu, dibutuhkan suatu mekanisme untuk mengatasi masalah keamanan data video. Salah satunya adalah dengan menggunakan teknologi kriptografi, data video dilakukan enkripsi sebelum didistribusikan atau sekedar diamankan didalam komputer, ketika file akan dipulihkan kembali maka dilakukan dekripsi video. Dengan demikian saya mengangkat judul *"Aplikasi Desktop Enkripsi Dekripsi Video Menggunakan Algoritma Kriptografi SEED 128"*. Kriptografi adalah ilmu penulisan rahasia dengan tujuan menyembunyikan makna dari pesan (Christof & Pelzl, 2009). Algoritma kriptografi ada beberapa jenis, dan salah satunya adalah algoritma kunci simetris.

Pada penelitian ini digunakan algoritma kriptografi SEED 128 untuk membangun sistem enkripsi dan dekripsi data video. SEED adalah kunci simetris blok cipher 128-bit yang dikembangkan oleh KISA (Korea mation Security Agency) dan tim ahli sejak tahun 1998. SEED adalah

standar asosiasi industri nasional (TTAS KO-12,0004, 1999). SEED telah diadopsi untuk sebagian besar sistem keamanan di Korea (KISA, 2005).

I.2. Rumusan Masalah

Berdasarkan latar belakang diatas, adapun rumusan masalah untuk penelitian ini adalah bagaimana membuat sistem enkripsi dan dekripsi dengan mengimplimentasikan algoritma SEED 128 untuk mengamankan video.

I.3. Batasan Masalah

Hal-hal yang dibatasi pada penelitian ini agar tidak memperluas pembahasan, yaitu:

1. Algoritma kriptografi yang digunakan adalah SEED 128.
2. Bahasa pemrograman yang dipakai adalah Java dengan IDE Netbeans
3. Jenis data yang digunakan adalah video dengan format MP4, MKV, FLV, AVI, dan WMV.
4. Pendistribusian dan pembobolan kunci tidak dibahas.
5. Performansi yang diukur adalah waktu proses enkripsi dan dekripsi dan kualitas data video dari segi ukuran.

I.4. Tujuan Penelitian

Tujuan dari peneilitian ini adalah membuat sistem enkripsi dan dekripsi untuk mengamankan konten video menggunakan algoritma kriptografi SEED 128.

I.5. Manfaat Penelitian

Manfaat penelitian ini adalah menghasilkan sistem aplikasi desktop yang bisa digunakan untuk mengamankan dan melindungi data video yang secara legal milik produsen agar tidak disalahgunakan oleh pihak yang tidak berhak.

I.6. Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini adalah sebagai berikut :

1. BAB I PENDAHULUAN

   Bab ini menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi penelitian, serta sistematika penulisan tugas akhir.

2. BAB II TINJAUAN PUSTAKA

   Bab ini menjelaskan tentang teori-teori dasar dan penelitian-penelitian terkait dari berbagai literatur yang menjadi referensi dalam pengerjaan proyek tugas akhir ini.

3. BAB III ANALISIS DAN PERANCANGAN SISTEM

   Bab ini merincikan metodologi penelitian, analisis serta rancangan dari sistem aplikasi pada proyek tugas akhir ini.

4. BAB IV HASIL DAN PEMBAHASAN

   Berisi tentang implementasi sistem dan pengujian sistem hasil penelitian.

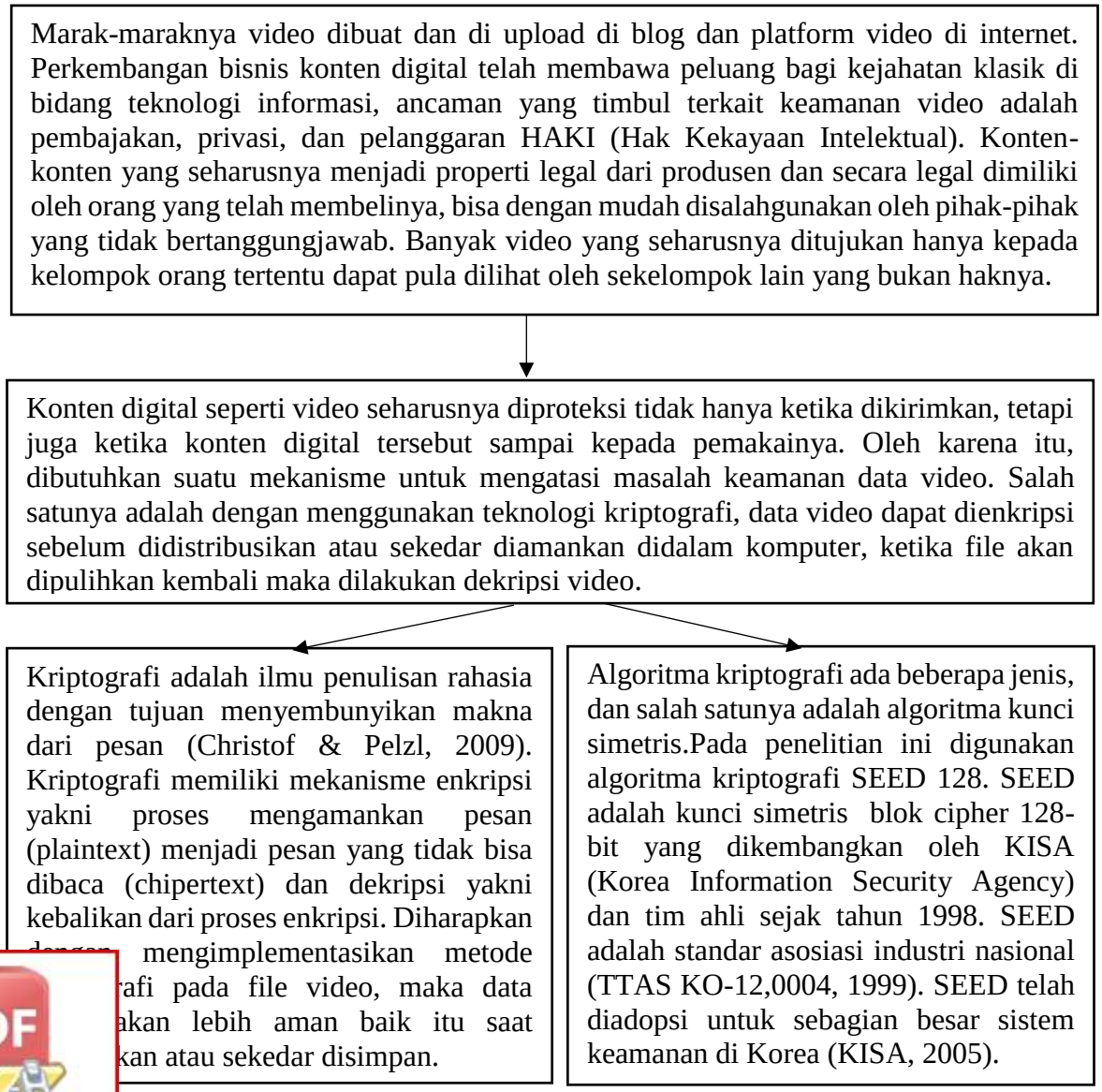
5. BAB V PENUTUP

   Bab ini berisi kesimpulan akhir serta saran pengembangan dari pengerjaan proyek tugas akhir ini untuk penelitian di masa yang akan datang.

BAB II

TINJAUAN PUSTAKA

Pada bab ini berisi teori-teori yang mendasari obyek yang diteliti, diperoleh dari menghimpun data-data atau sumber-sumber yang berhubungan dengan topik penelitian tentang video, kriptografi, dan algoritma SEED 128. Gambar 2.1 memperlihatkan kerangka pikir pada penelitian ini.



Gambar 2.1. Kerangka Pikir Penelitian

Penelitian ini berfokus pada pengimplementasian algoritma kriptografi SEED 128 untuk enkripsi dan dekripsi file video, dimana akan dilihat durasi pemrosesan data pada saat pengujian sistem dalam melakukan enkripsi dan dekripsi. Jika sistem bisa melakukan enkripsi dan dekripsi dengan kunci yang sama maka sistem telah dikatakan berhasil dijalankan.

II.1. Video

Video adalah informasi yang berisi gambar dan suara serta memiliki ciri khas gambar bergerak dengan kecepatan tertentu atau frame per second. Parameter video menentukan kualitas video,terdapat 3 parameter video, yakni :

1. Frame per Second (FPS), adalah banyaknya frame yang dimainkan tiap detik. Nilai FPS adalah 20 hingga 30 fps.
2. Bitrate, adalah nilai pengukuran dari bit yang dikirimkan per waktu tertentu.
3. Resolution, adalah ukuran gambar yang ditampilkan pada layar.

Video pada dasarnya tersusun atas serangkaian frame dimana rangkaian frame tersebut ditampilkan pada layar dengan kecepatan tertentu tergantung pada frame rate yang diberikan biasanya ditandai dalam frame per second. Jika frame rate tinggi, maka mata manusia tidak dapat menangkap gambar au frame melainkan menangkapnya sebagai rangkaian yang ntinu/berlanjut (video). Video merupakan suatu teknologi yang berfungsi

untuk menangkap, merekam, memproses, ataupun mentransmisikan dan menata ulang gambar bergerak yang biasa disebut gambar-gambar mati yang dibaca berurutan dalam suatu waktu dan dalam kecepatan tertentu. Video terbagi atas dua jenis, yaitu:

- Video Analog, Analog video tersusun dari gelombang bersambung yang bervariasi, dengan kata lain nilai sinyal akan memiliki angka yang beragam tetapi terbatas pada batas maksimum dan minimum yang diijinkan. Video Analog merupakan produk dari industri pertelevisian dan oleh sebab itu dijadikan sebagai standard televisi video analaog (kebanyakan masih digunakan untuk penyiaran televisi) masih merupakan platform yang paling banyak dipasang untuk mengirim dan melihat video. Dalam sistem analog, sinyal video dari kamera dikirim ke dalam video VCR (*Video Cassette Recording*), yang akan direkam dalam video tape magnetik.
- Video Digital, digital video ditransmisikan hanya berupa titik presisi yang dipilih pada interval dalam kurva. Tipe sinyal digital yang dapat dipakai oleh komputer kita adalah tipe binary. Data binary diwakili dengan angka 1 dan 0, angka 1 mewakili nilai maksimum dan angka 0 mewakili nilai minimum. Video Digital adalah produk dari industri komputer dan oleh sebab itu dijadikan standard data digital. Arsitektur video digital tersusun atas sebuah format dalam mengkode dan memainkan kembali file video engan komputer, dan sebuah software media player yang dapat digunakan ntuk membuka format file video tersebut. Arsitektur video digital yang

dapat digunakan adalah Apple QuickTime, Microsoft Windows Media Format, dan Real Network RealMedia. Format file video yang berhubungan adalah QuickTime movie (.mov), Audio Video Interleaved (.avi), dan RealMedia (.rm). Beberapa software media player dapat mengenali dan memainkan lebih dari satu format file video.

II.2. Format Video

Format video melibatkan dua konsep teknologi yang berbeda : containers (kadang-kadang disebut pembungkus) dan codec (kependekan dari coder / decoder).

Container mengambarkan struktur file: di mana berbagai potongan disimpan, bagaimana mereka disisipkan, dan codec mana yang digunakan oleh potongan mana. Ini dapat menentukan codec audio serta video. Ini digunakan untuk mengemas video & komponennya (audio / metadata) dan diidentifikasi (biasanya) oleh ekstensi file seperti .AVI, .MP4 atau .MOV.

Codec (singkatan dari "coder / decoder") adalah cara pengkodean audio atau video ke dalam aliran byte. Ini adalah metode yang digunakan untuk menyandikan video dan merupakan penentu utama kualitas.

Dimisalkan Container sebagai file itu sendiri dan codec sebagai isinya. Yang penting untuk disadari adalah bahwa format container yang baik dapat menampung banyak codec. Misalnya container .MOV dapat menampung ir semua jenis data codec. Hal yang sama berlaku untuk file .MP4 dan n .AVI dapat menampung beragam codec sebagai isinya. Kontainer tidak

menentukan kualitas atau fitur video itu sendiri, itu tergantung pada codec. Cara yang tepat untuk menggambarkan video adalah dengan menunjukkan keduanya: File .MOV yang berisi data H.264 dan file .AVI yang berisi data DivX.

- AVI (Audio Video Interleave), format ini termasuk format video yang tidak dikompresi. AVI adalah format standar file video untuk Microsoft Windows yang juga merupakan format video tertua karena diperkenalkan sejak Windows 3.1. Video yang menggunakan format ini akan menghasilkan ukuran file yang sangat besar karena resolusi yang dipakai sesuai dengan resolusi asli dari sumber videonya, yaitu kaset video. Format ini merupakan salah satu format yang berkualitas tinggi karena mampu menghasilkan pergerakan 15 frame perdetik dalam resolusi maksimal dengan kualitas suara mencapai 11,025 Hz.

  File AVI dapat berisi data audio dan video dalam container file yang memungkinkan pemutaran audio dengan video secara sinkron. AVI adalah turunan dari Resource Interchange File Format (RIFF), yang membagi data file menjadi "potongan". Setiap "potongan" diidentifikasi oleh sebuah tag. File AVI mengambil bentuk potongan tunggal dalam file berformat RIFF, yang kemudian dibagi lagi menjadi dua "potongan" wajib dan satu "potongan" opsional. Sub-potongan pertama diidentifikasi oleh tag "hdrl". Sub-potongan ini adalah header file dan berisi metadata tentang video, berti lebar, tinggi, dan frame rate. Sub-potongan kedua diidentifikasi oleh "movi". Potongan ini berisi data audio / visual aktual yang membentuk

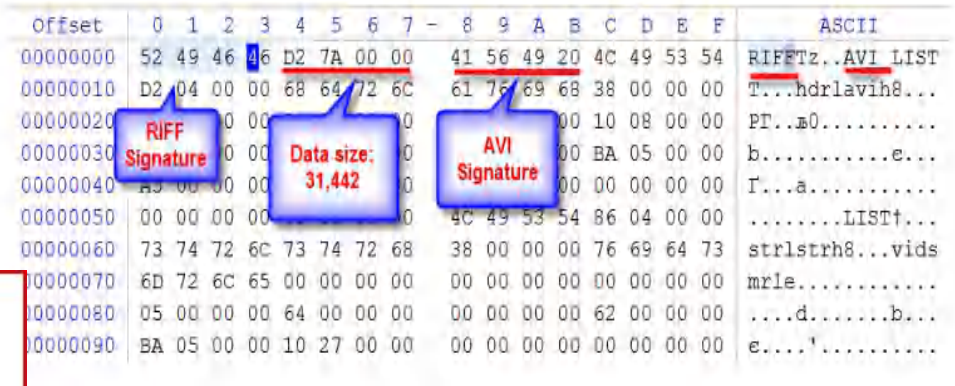
film AVI. Opsional ketiga ORANGE Sub-potongan diidentifikasi oleh tag "idx1" yang mengindeks offset dari potongan data dalam file.

Setiap container RIFF, termasuk informasi audio-video AVI harus memiliki signature (tag) RIFF (hex: 52 49 46 46) di awal file.

File RIFF disusun dalam segmen data (potongan). Setiap segmen diawali dengan header 12 byte: signature 4 byte (RIFF), ukuran data 4 byte (urutan little-endian, byte rendah pertama) dan 4 byte tipe RIFF: signature AVI [spasi]. Ukuran potongan adalah ukuran data plus 8 byte. Meringkas ukuran untuk semua potongan yang ditemukan kemudian menghitung ukuran file total.

Contoh : sample.avi berdurasi 4 detik dengan ukuran 31.744 bytes.

Saat memeriksa data file sample.avi menggunakan Hex Viewer, seperti Active @ Disk Editor, yang termasuk dalam paket Active @ File Recovery, kita dapat melihatnya dimulai dengan signature RIFF (hex: 52, 49, 46, 46). Pada offset 8 terdapat signature Audio Video Interleave RIFF Type AVI [spasi] (hex: 41, 56, 49, 20). Berikut gambar 2.2 informasi potongan data biner file avi.



Gambar 2.2. Informasi potongan data biner file avi.

Pada offset 4 ada ukuran data: 31.442 (hex: D2, 7A, 00, 00) dalam urutan little-endian (byte rendah pertama). Menambahkan panjang header ke ukuran data, dapat dihitung total ukuran file AVI: 31.442 + 8 = 31.450 byte. Ukuran file aktual adalah 31.744 byte, namun mulai dari offset 31.450 dalam file hanya ada nol, yang berarti bahwa ukuran file hanya meningkat untuk disejajarkan pada batas 1kb (kelipatan 1.024 byte).

- Moving Picture Experts Group (MPEG), format ini merupakan standar untuk hasil kompresi file digital video audio. MPEG menghasilkan kualitas gambar yang tinggi tapi tidak membutuhkan kapasitas file besar. Kompresi file MPEG terkadang menghilangkan sejumlah frame perpindahan sehingga proses transisinya sering tidak enak dipandang. Resolusi video yang berformat MPEG mendukung resolusi setengah layar dan satu layar tergantung jenis MPEG-nya. Format ini memiliki beragam standar antara lain MPEG-1, MPEG-2 dan MPEG-4. Versi MPEG-1 hanya mampu menghasilkan kualitas video dibawah video VCR. Sedangkan untuk MPEG-2 mempunyai kecepatan 60 frame per detik sehingga mampu mengompresi video berdurasi 2 jam dalam beberapa gigabytes saja. Format MPEG-4 dikeluarkan pada saat tahun 1998 dan biasa digunakan dalam aplikasi internet, ponsel dan televisi. Dengan kapasitasnya yang kecil, MPEG-4 dapat menunjang transmisi via jaringan ber-bandwith kecil. Pada umumnya, kamera video digital sudah mampu menghasilkan output berupa MPEG. rmat file MP4 dirancang untuk menampung media informasi presentasi PEG-4 secara fleksibel, format extensible yang memfasilitasi pertukaran,

manajemen, editing, dan presentasi media. Presentasi ini mungkin 'lokal' ke sistem yang berisi presentasi, atau mungkin melalui jaringan atau mekanisme pengiriman arus lainnya. Format filenya ini dirancang untuk bebas dari protokol pengiriman apapun saat dukungan yang efisien memungkinkan untuk pengiriman secara umum. Desainnya didasarkan pada format QuickTime dari Apple Computer Inc. Format file MP4 terdiri dari object-oriented struktur yang disebut 'atom'. Tag yang unik dan panjang mengidentifikasi setiap atom. Kebanyakan atom menggambarkan hierarki metadata memberikan informasi seperti titik indeks, jangka waktu, dan pointer ke data media. Kumpulan atom ini terkandung dalam sebuah atom yang disebut 'atom film'. Data media itu sendiri terletak di tempat lain, bisa di file MP4, terkandung dalam satu atau lebih 'mdat' atau data media atom, atau berada di luar file MP4 dan direferensikan via URL's. Format file MP4 adalah format streamable, sebagai lawan dari format streaming. Artinya, format file tidak mendefinisikan sebuah protokol on-the-wire, dan tidak pernah benar-benar dialirkan melalui media transmisi. Sebagai gantinya, metadata dalam file diketahui sebagai 'petunjuk trek' memberikan instruksi, memberi tahu server aplikasi bagaimana menyampaikan data media melalui protokol pengiriman tertentu. Ada beberapa petunjuk trek untuk satu presentasi, menggambarkan bagaimana menyampaikan berbagai macam pengiriman protokol. Dengan cara ini, format file memudahkan streaming apa pernah dialirkan secara langsung. Metadata dalam file, dikombinasikan ngan penyimpanan data media yang fleksibel, memungkinkan format MP4

mendukung streaming, editing, pemutaran lokal, dan pertukaran konten, sehingga memenuhi persyaratan format MPEG-4 Intermediate (Park, Kim, & Yoon).

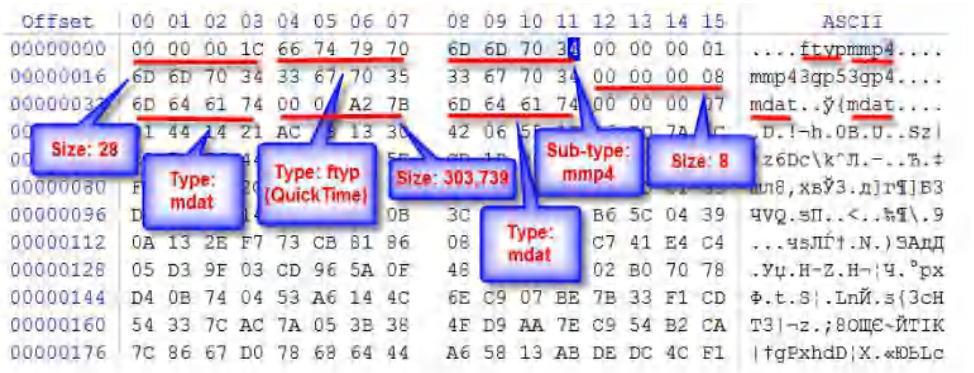
MPEG-4 Bagian 14 atau MP4 adalah format multimedia digital yang paling umum digunakan untuk menyimpan video dan audio, tetapi juga dapat digunakan untuk menyimpan data lain seperti subtitle dan gambar foto. MP4 memungkinkan streaming melalui Internet. Satu-satunya ekstensi nama file resmi untuk file MPEG-4 Bagian 14 adalah .mp4, tetapi banyak yang memiliki ekstensi lain, paling umum .m4a dan .m4p. Spesifikasi format file MPEG-4 didasarkan pada spesifikasi format QuickTime.

File MP4 terdiri dari potongan berurutan. Setiap potong memiliki 8 byte header: potongan berukuran 4-byte (big-endian, byte tinggi pertama) dan tipe potongan 4-byte - salah satu dari tanda tangan yang telah ditentukan: "ftyp", "mdat", "moov", "pnot" , "udta", "uuid", "moof", "free", "skip", "jP2", "wide", "load", "ctab", "imap", "matt", "kmat", " klip "," crgn "," sync "," chap "," tmcd "," scpt "," ssrc "," PICT ".

Potongan pertama harus dari tipe "ftype" dan memiliki sub-tipe di offset 8. MP4 ditentukan oleh sub-tipe yang harus merupakan salah satu dari nilai: "avc1", "iso2", "isom", "isom", "mmp4", "mp41 "," mp42 "," mp71 "," msnv "," ndas "," ndsc "," ndsh "," ndsm "," ndsp "," ndss "," ndxc "," ndxh "," ndxh "," ndxm ", "ndxp", "ndxs". Iterasi potongan sampai jenis yang tidak ketahui terdeteksi menyusun file MP4.

ntoh : video sample.mp4 berdurasi 31 detik dengan ukuran 309.977 bytes.

Saat memeriksa data biner file sample.mp4 menggunakan Hex Viewer, seperti Active @ Disk Editor, kita dapat melihatnya dimulai dengan ftyp tanda tangan (hex: 66 74 79 70) pada offset 4, yang menentukan Jenis File Container QuickTime. Berikut gambar 2.3 informasi potongan data biner file mp4.



Gambar 2.3. Informasi potongan data biner file mp4.

Sub-tipe file adalah mmp4 (hex: 6D 6D 70 34) yang mengarah ke tipe file MP4. Ukuran blok pertama adalah 28 (hex: 00 00 00 1C, big-endian, byte tinggi pertama), ukuran terletak di offset 0.

Pada offset 28 (hex: 1C) terletak potongan kedua, yang memiliki ukuran 8 dan tipe mdat (hex: 6D 64 61 74).

Potongan berikutnya terletak pada offset 28 + 8 = 36 (hex: 24) dan memiliki ukuran 303.739 (hex: 00 04 A2 7B) dan ketik mdat (hex: 6D 64 61 74) pada offset 40 (hex: 28).

Potongan berikutnya terletak di offset 36 + 303.739 = 303.775 dan memiliki ukuran 6.202 (hex: 00 00 18 3A) dan ketik moov (hex: 6D 6F 6F 76) di offset 3.779.

Ini adalah potongan terakhir, jadi total ukuran file adalah 303.775 + 6.202 = 309.977 byte.

- MOV, termasuk dalam format video terkompresi. MOV dibuat oleh Apple Computer dan dijalankan pada platform Macintosh tetapi sekarang dapat dijalankan pada Windows dengan meng-installcodec Quicktime. MOV termasuk video yang ditujukan untuk online video, website yang berbasis multimedia dan CD-ROM. Format ini bisa langsung dihasilkan oleh ponsel berkamera yang memiliki fitur membuat video. Salah satu keunggulan format MOV adalah mampu mendukung video interaktif, yaitu Virtual Reality (VR).
- 3GP (3GPP Format file), adalah sebuah multimedia container format yang ditetapkan oleh Third Generation Partnership Project untuk 3G UMTS jasa multimedia. Yang digunakan di 3G ponsel, tetapi juga dapat dimainkan pada beberapa 2G dan 4G. Ukurannya lebih kecil dari pada AVI dan MPEG.
- FLV (Flash Video), adalah sebuah wadah format file yang digunakan untuk mengirimkan video melalui internet mengunakan Adobe Flash Player. Awal diproduksi oleh Macromedia versi 6-10. Konten video flash juga mungkin tertanam di dalam SWF file. Ada dua format file video yang berbeda didefinisikan oleh Adobe System dan didukung dalam Adobe Flash Player. Audio dan Video FLV data dalam di encode dalam cara yang sama ketika mereka berada dalam file SWF. Yang terakhir format file F4V didasarkan da basis ISO format file media dan didukung dimulai dengan Flash Player

9 Update 3. Format FLV memiliki ukuran yang lebih kecil dari AVI dan MOV, tetapi lebih besar dari format SWF dan MPEG.

- SWF (Shockwave Flash), berdiri untuk "Small WebFormat" kemudian berubah menjadi "Shockwave Flash" oleh Macromedia, kemudian kembali berubah kembali ke SmallWebFormat ketika perusahaan memilih untuk terbuka repositori untuk multimedia dan terutama untuk vector graphics, berasal dari Future Wave Software dan telah datang di bawah kendali Adobe. Dimaksudkan untuk menjadi cukup kecil untuk dipublikasikan di Web, SWF file dapat berisi animasi atau applet dari berbagai tingkat interaktif dan fungsi. Format SWF memiliki ukuran sedang, kira - kira setengah ukuran AVI.
- WMV (Windows Media Video), adalah bagian dari sistem Windows Media buatan Microsoft adalah sebuah codec untuk mengencode film dan mentransformasikan slide show yang berisi format bitmap kedalam video terkompres. WMV sebenarnya adalah versi proprietary dari MPEG-4. Video Stream sering dikombinasikan dengan Audio Stream dalam format WMA, dengan video WMV yang dikemas kedalam container AVI atau ASF.
- VOB (Video Object), adalah sebuah format kontainer di DVD-Video media. VOB dapat berisi video, audio, subtitle dan menu isi multiplexing bersama sama ke dalam bentuk sungai. VOB didasarkan pada aliran program MPEG format , tetapi dengan keterbatasan dan beberapa spesifikasi tambahan.
- KV (Matroska Video), merupakan standard format kontainer Multimedia ng bersifat terbuka, merupakan format file yang dapat menyimpan banyak

(tidak terbatas) jumlah video, audio, gambar, track subtitle hanya dalam sebuah atau satu file. Sebenarnya mempunyai konsep yang mirip dengan format AVI, MP4,3GP,FLV atau ASF, tetapi Matroska merupakan format dengan spesifikasi yang terbuka sepenuhnya (open source). MKV merupakan tipe file Matroska untuk video dengan subtitle dan audio (Arifin, 2015).

II.3. Kriptografi

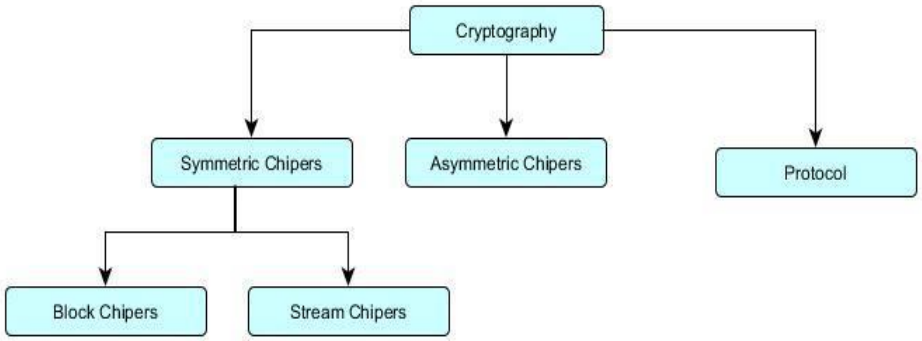
Kriptografi adalah ilmu penulisan rahasia dengan tujuan menyembunyikan arti sebuah pesan (Christof & Pelzl, 2009).

Berikut ini merupakan beberapa aspek penting dalam kriptografi:

1. Authentication, penerima informasi atau pesan dapat memastikan keaslian pesan yang diterima. apakah informasi atau pesan tersebut benar-benar berasal dari pengirim yang sebenarnya.
2. Integrity, dapat memastikan keaslian informasi atau pesan yang dikirim melalui jaringan serta memastikan bahwa informasi atau pesan tersebut tidak dimodifikasi oleh orang lain
3. Non-Repudation, pengirimlah benar-benar yang telah mengirim informasi atau pesan tersebut.
4. Authority, informasi yang telah dikirim melalui jaringan tidak dapat dimodifikasi oleh orang yang tidak memiliki hak untuk mengakses informasi tersebut.

5. Confidentiality, menjaga kerahasiaan informasi dari pengguna yang tidak memiliki hal akses.
6. Privacy, hal-hal yang bersifat pribadi
7. Availability, merupakan ketersediaan informasi yang dibutuhkan.

Kriptografi sendiri terbagi menjadi cabang-cabang yang penting, sebagai berikut :



Gambar 2.4. Cabang Kriptografi

Pada dasarnya kriptografi dapat dibagi menjadi tiga cabang penting, yaitu:

1. Algoritma kunci simetri (Symmetric Chipers)

Algoritma-algoritma yang berada pada kategori ini mempunyai kunci untuk enkripsi dan dekripsi yang sama.

2. Algoritma kunci nirsimetri (Asymmetric-key Chipers)

Algoritma-algoritma dalam kategori ini mempunyai kunci untuk enkripsi (kunci publik) yang berbeda dengan kunci untuk dekripsi (kunci rahasia). Kunci public at umum/tidak rahasia.

3. Protocol Kriptografi

Secara kasar, protokol crypto berurusan dengan aplikasi algoritma kriptografi. Algoritma simetris dan asimetris dapat dilihat sebagai blok bangunan dengan aplikasi seperti komunikasi Internet aman yang dapat direalisasikan. Skema Lapisan Keamanan Transportasi (TLS), yang digunakan dalam browser everyWeb, adalah contoh dari protokol cryptographic (Christof & Pelzl, 2009).

Algoritma kriptografi (cipher) yang beroperasi dalam mode bit dapat dikelompokkan menjadi dua kategori:

1. Cipher Aliran (Stream Cipher)

Algoritma kriptografi beroperasi pada plainteks/cipherteks dalam bentuk bit tunggal, yang dalam hal ini rangkaian bit dienkripsikan/didekripsikan bit per bit.

2. Cipher Blok (Block Cipher)

Algoritma kriptografi beroperasi pada plainteks/cipherteks dalam bentuk blok bit, yang dalam hal ini rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya. Misalnya panjang blok adalah 64 bit, maka itu berarti algoritma enkripsi memperlakukan 8 karakter setiap kali penyandian (1 karakter = 8 bit dalam pengkodean ASCII).

Baik cipher aliran maupun cipher blok, keduanya termasuk ke dalam goritma kriptografi simetri (Munir, 2004).

II.3.1. Chiper Blok

Pada cipher blok, rangkaian bit-bit plainteks dibagi menjadi blok-blok bit dengan panjang sama, biasanya 64 bit (tapi adakalanya lebih). Enkripsi dilakukan terhadap blok bit plainteks menggunakan bit-bit kunci (yang ukurannya sama dengan ukuran blok plainteks). Algoritma enkripsi menghasilkan blok cipherteks yang berukuran sama dengan blok plainteks. Dekripsi dilakkan dengan cara yang serupa seperti enkripsi.

Misalkan blok plainteks *(P)* yang berukuran *m* bit dinyatakan sebagai vector

$$P = (p1, p2, ..., pm)$$

yang dalam hal ini p_i adalah 0 atau 1 untuk i = 1, 2, …, *m*, dan blok cipherteks *(C)* adalah

$$C = (c1, c2, ..., cm)$$

yang dalam hal ini c_i adalah 0 atau 1 untuk i = 1, 2, …, m.

Bila plainteks dibagi menjadi n buah blok, barisan blok-blok plainteks dinyatakan sebagai

$$(P1, P2, \ldots, Pn)$$

Untuk setiap blok plainteks Pi, bit-bit penyusunnya dapat dinyatakan sebagai vektor

$$Pi = (p_{i1}, p_{i2}, \ldots, p_{im})$$

Enkripsi dan dekripsi dengan kunci K dinyatakan berturut-turut dengan persamaan

$E_K(P) = C$…… untuk enkripsi

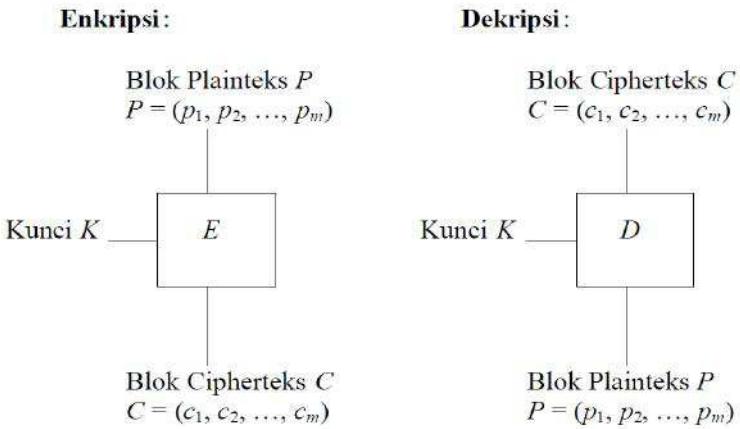
dan

$D_K(C) = P$…….. untuk dekripsi

Fungsi E haruslah fungsi yang berkoresponden satu-ke-satu, sehingga

$E^{-1} = D$

Skema enkripsi dan dekripsi dengan cipher blok digambarkan pada Gambar 2.5. Fungsi E dan D dispesifikasikan oleh kriptografer.



Gambar 2.5. Skema enkripsi dan dekripsi pada chiper blok

II.3.2. Teknik Kriptografi Klasik digunakan pada Chiper Blok

Algoritma blok cipher menggabungkan beberapa teknik kriptografi klasik dalam proses enkripsi. Dengan kata lain, cipher blok dapat diacu sebagai super-enkripsi. Teknik kriptografi klasik yang digunakan adalah :

1. Substitusi.

   Teknik ini mengganti satu atau sekumpulan bit pada blok plainteks tanpa mengubah urutannya. Secara matematis, teknik substitusi ini ditulis sebagai

$$c_i = E(p_i),\ i = 1, 2, \ldots \text{ (urutan bit)}$$

yang dalam hal ini ci adalah bit cipherteks, pi adalah bit plainteks, dan f adalah fungsi substitusi. Dalam praktek, E dinyatakan sebagai fungsi matematis atau dapat merupakan tabel susbtitusi (S-box).

2. Transposisi atau permutasi

Teknik ini memindahkan posisi bit pada blok plainteks berdasarkan aturan tertentu. Secara matematis, teknik transposisi ini ditulis sebagai

$$C = PM$$

yang dalam hal ini C adalah blok cipherteks, P adalah blok plainteks, dan M adalah fungsi transposisi. Dalam praktek, M dinyatakan sebagai tabel atau matriks permutasi.

Selain kedua teknik di atas, cipher blok juga menggunakan dua teknik tambahan sebagai berikut:

3. Ekspansi

Teknik ini memperbanyak jumlah bit pada blok plainteks berdasarkan aturan tertentu, misalnya dari 32 bit menjadi 48 bit.

4. Kompresi

Teknik ini kebalikan dari ekspansi, di mana jumlah bit pada blok plainteks diciutkan berdasarkan aturan tertentu.

II.3.3. Mode Operasi Chiper Blok

Plainteks dibagi menjadi beberapa blok dengan panjang tetap. Beberapa mode operasi dapat diterapkan untuk melakukan enkripsi terhadap keseluruhan blok plainteks. Empat mode operasi yang lazim diterapkan pada sistem blok cipher adalah:

1. Electronic Code Book (ECB)

   Pada mode ini, setiap blok plainteks Pi dienkripsi secara individual dan independen menjadi blok cipherteks Ci. Secara matematis, enkripsi dengan mode ECB dinyatakan sebagai

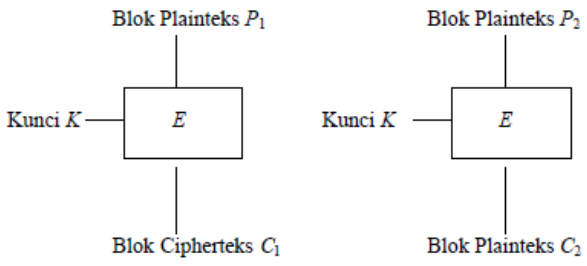
   $$Ci = EK(Pi)$$

   dan dekripsi sebagai

   $$Pi = DK(Ci)$$

   yang dalam hal ini, Pi dan Ci masing-masing blok plainteks dan cipherteks ke-i.

   Gambar 2.6 memperlihatkan enkripsi dua buah blok plainteks, P1 dan P2 dengan mode ECB, yang dalam hal ini E menyatakan fungsi enkripsi yang melakukan enkripsi terhadap blok plainteks dengan menggunakan kunci K.



Gambar 2.6. Skema Enkripsi dan Dekripsi dengan mode ECB

Contoh : Misalkan plainteks (dalam biner) adalah

10100010001110101001

Bagi plainteks menjadi blok-blok yang berukuran 4 bit:

1010 0010 0011 1010 1001

atau dalam notasi HEX adalah A23A9.

Misalkan kunci (K) yang digunakan adalah (panjangnya juga 4 bit) 1011

atau dalam notasi HEX adalah B.

Misalkan fungsi enkripsi *E* yang sederhana (tetapi lemah) adalah dengan meng XOR-kan blok plainteks *Pi* dengan *K*, kemudian geser secara *wrapping* bit-bit dari *Pi* ⊕ *K* satu posisi ke kiri. Proses enkripsi untuk setiap blok digambarkan sebagai berikut:

```
                1010 0010 0011 1010 1001
                1011 1011 1011 1011 1011 ⊕
               ---------------------------
Hasil XOR:      0001 1001 1000 0001 0010
Geser 1 bit ke kiri:  0010 0011 0001 0010 0100
Dalam notasi HEX:      2    3    1    2    4
```

Jadi, hasil enkripsi plainteks

10100010001110101001 (A23A9 dalam notasi HEX)

adalah

00100011000100100100 (23124 dalam notasi HEX).

blok plainteks yang sama selalu dienkripsi menjadi blok cipherteks yang sama (atau identik). Pada contoh di atas, blok 1010 muncul dua kali dan selalu dienkripsi menjadi 0010. Contoh yang lebih nyata misalkan pesan

KUTU BUKU DI LEMARIKU

dibagi menjadi blok-blok yang terdiri dua huruf (dengan menghilangkan semua spasi) menjadi

KU TU BU KU DI LE MA RI KU

maka blok yang menyatakan “KU” akan dienkripsi menjadi blok cipherteks (dua huruf) yang sama.

Kata “code book” di dalam ECB muncul dari fakta bahwa karena blok plainteks yang sama selalu dienkripsi menjadi blok cipherteks yang sama, maka secara teoritis dimungkinkan membuat buku kode plainteks dan cipherteks yang berkoresponden.

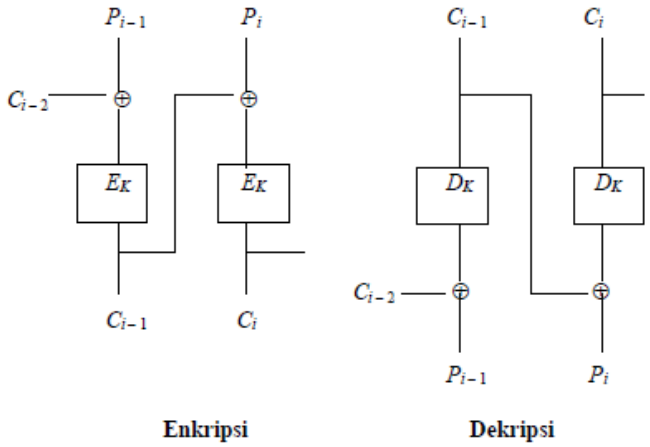
Namun, semakin besar ukuran blok, semakin besar pula ukuran buku kodenya. Misalkan jika blok berukuran 64 bit, maka buku kode terdiri dari $2^{64} – 1$ buah kode (entry), yang berarti terlalu besar untuk disimpan. Lagipula, setiap kunci mempunyai buku kode yang berbeda.

- Padding

  Ada kemungkinan panjang plainteks tidak habis dibagi dengan panjang ukuran blok yang ditetapkan (misalnya 64 bit atau lainnya). Hal ini mengakibatkan blok terakhir berukuran lebih pendek daripada blok-blok lainnya. Satu cara untuk mengatasi hal ini adalah dengan padding, yaitu menambahkan blok terakhir dengan pola bit yang teratur agar panjangnya sama dengan ukuran blok yang ditetapkan. Misalnya ditambahkan bit 0 semua, atau bit 1 semua, atau bit 0 dan bit 1 berselang-seling. Misalkan ukuran blok adalah 64 bit (8 byte) dan blok terakhir terdiri dari 24 bit (3 byte). Tambahkan blok terakhir dengan 40 bit (5 byte) agar menjadi 64 bit, misalnya dengan menambahkan 4 buah byte 0 dan satu buah byte angka 5. Setelah dekripsi, hapus 5 byte terakhir dari blok dekripsi terakhir.

2. Cipher Block Chaining (CBC)

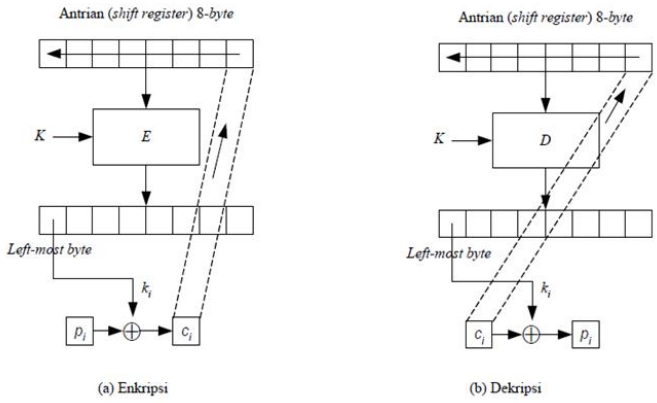
Mode ini menerapkan mekanisme umpan-balik (feedback) pada sebuah blok, yang dalam hal ini hasil enkripsi blok sebelumnya diumpan-balikkan ke dalam enkripsi blok yang current. Caranya, blok plainteks yang current di-XOR-kan terlebih dahulu dengan blok cipherteks hasil enkripsi sebelumnya, selanjutnya hasil peng-XOR-an ini masuk ke dalam fungsi enkripsi. Dengan mode CBC, setiap blok cipherteks bergantung tidak hanya pada blok plainteksnya tetapi juga pada seluruh blok plainteks sebelumnya. Dekripsi dilakukan dengan memasukkan blok cipherteks yang current ke fungsi dekripsi, kemudian meng-XOR kan hasilnya dengan blok cipherteks sebelumnya. Dalam hal ini, blok cipherteks sebelumnya berfungsi sebagai umpan-maju (feedforward) pada akhir proses dekripsi.



Gambar 2.7. Skema enkripsi dan dekripsi dengan mode CBC

3. Cipher Feedback (CFB)

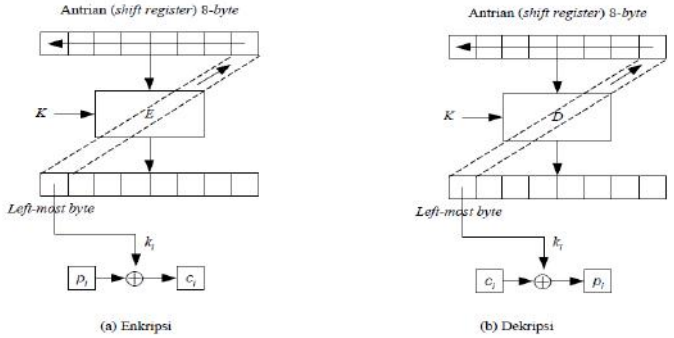
Mengatasi kelemahan pada mode CBC jika diterapkan pada komunikasi data (ukuran blok yang belum lengkap) Data dienkripsikan dalam unit yang lebih kecil daripada ukuran blok. Unit yang dienkripsikan dapat berupa bit per bit (jadi seperti cipheraliran), 2 bit, 3-bit, dan seterusnya. Bila unit yang dienkripsikan satu karakter setiap kalinya, maka mode CFB-nya disebut CFB8-bit.CFB n-bit mengenkripsi plainteks sebanyak n bit setiap kalinya, n≤m (m = ukuran blok). Dengan kata lain, CFB mengenkripsikan cipherblok seperti pada cipheraliran. Mode CFB membutuhkan sebuah antrian (queue) yang berukuran sama dengan ukuran blok masukan. Tinjau mode CFB8-bit yang bekerja pada blok berukuran 64-bit (setara dengan 8 byte) pada gambar 2.8.



Gambar 2.8. Skema enkripsi dan dekripsi dengan mode CFB

4. Output Feedback (OFB)

   Mode OFB mirip dengan mode CFB, kecuali n bit dari hasil enkripsi terhadap antrian disalin menjadi elemen posisi paling kanan diantrian. Dekripsi dilakukan sebagai kebalikan dari proses enkripsi. Gambar 2.9 adalah mode OFB 8-bit yang bekerja pada blok berukuran 64 bit (setara dengan 8 byte).



Gambar 2.9. Skema enkripsi dan dekripsi dengan mode OFB.

II.4. Algoritma Kriptografi SEED 128

SEED adalah blok kunci simetri 128-bit yang telah dikembangkan oleh KISA (Korea Information Security Agency) dan tim ahli sejak tahun 1998. SEED adalah standar asosiasi industri nasional (TTAS KO-12.0004, 1999). SEED telah diadopsi ke sebagian besar sistem keamanan di Korea. SEED dirancang untuk memanfaatkan S-box dan permutasi yang seimbang dengan teknologi komputasi saat ini. SEED memiliki struktur Feistel dengan 16 putaran, dan kuat terhadap kriptanalisis diferensial dan kriptanalisis linear ang seimbang dengan trade-off keamanan / efisiensi.

Dekripsi dan enkripsi lengkap dari algoritma SEED, yang merupakan chiper dengan blok data 128-bit dan kunci rahasia 128-bit. Fitur abstrak SEED diuraikan sebagai berikut:

- Struktur Feistel dengan 16 putaran;
- Ukuran blok data input-output 128 bit;
- Panjang kunci 128 bit;
- Dua S-box 8x8;
- Operasi campuran dari exclusive OR (XOR) dan modular addition.

Sedangkan notasi yang digunakan yaitu ;

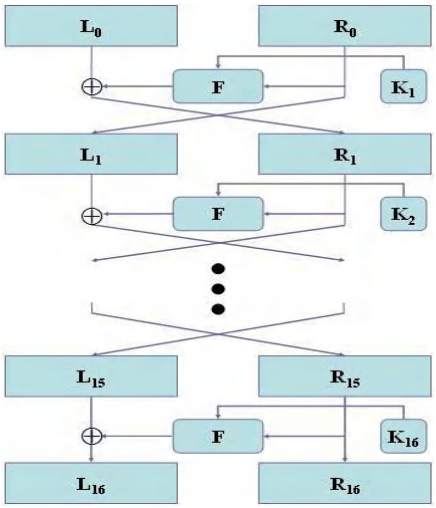
- & : bitwise AND
- + : penambahan dalam modular 2^{32}
- - : pengurangan dalam modular 2^{32}
- ⊕ : bitwise exclusive OR
- << n : rotasi melingkar kiri dengan n bit
- >> n : rotasi melingkar kanan dengan n bit
- ‖ : concatenation

II.4.1 Struktur SEED 128

Algoritma SEED 128 menggunakan struktur jaringan Feistel. Jaringan Feistel ak dipakai pada algoritma kriptografi seperti DES, LOKI, GOST, FEAL, fer, Blowfish, dan lain-lain karena model ini bersifat reversible untuk es enkripsi dan dekripsi. Sifat reversible ini membuat kita tidak perlu

membuat algoritma baru untuk mendekripsi cipherteks menjadi plainteks. Sifat reversible tidak bergantung pada *Fungsi F* sehingga *Fungsi F* dapat dibuat serumit mungkin.

Input 128 bit dibagi menjadi blok-blok 64 bit dan blok kanan 64 bit adalah input untuk fungsi putaran F dengan sebuah subkey 64 bit dihasilkan dari penjadwalan key. Struktur dari SEED di tunjukkan pada gambar 2.10.



Gambar 2.10. Struktur dari SEED

II.4.2. Fungsi putaran F

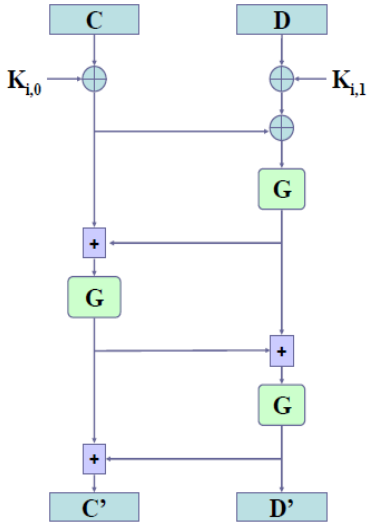
Blok input 64-bit dari fungsi putaran dibagi menjadi dua blok 32-bit (C, D) dan dibungkus dengan 4 fase: fase pencampuran dua blok subkey 32-bit ($K_{i,0}$; $K_{i,1}$) dan 3 lapisan fungsi G dengan tambahan untuk pencampuran dua blok 32-bit. Outputnya adalah C', D' dari fungsi F dengan dua blok input 32-bit C, D dan dua [illegible]bkunci $K_{i,0}$, $K_{i,1}$ adalah sebagai berikut:

$$C' = G[G[G\{(C \oplus K_{i,0}) \oplus (D \oplus K_{i,1})\} + (C \oplus K_{i,0})] + G\{(C \oplus K_{i,0}) \oplus (D \oplus K_{i,1})\}]$$

$$+ G[G\{(C \oplus K_{i,0}) \oplus (D \oplus K_{i,1})\} + (C \oplus K_{i,0})]$$

$$D' = G[G[G\{(C \oplus K_{i,0}) \oplus (D \oplus K_{i,1})\} + (C \oplus K_{i,0})] + G\{(C \oplus K_{i,0}) \oplus (D \oplus K_{i,1})\}]$$

Struktur fungsi putaran F ditunjukkan pada gambar berikut :



Gambar 2.11. Fungsi putaran F.

II.4.3. Fungsi G

Fungsi G memiliki dua lapisan: sebuah lapisan dari dua 8×8 S-box dan sebuah lapisan dari blok permutasi dari enam belas 8-bit sub-blok. Lapisan pertama dua S-box dihasilkan dari fungsi Boolean x^{247} dan x^{251}. Lapisan kedua adalah satu set permutasi di setiap S-box.

Output Z_0, Z_1, Z_2, Z_3 dari fungsi G dengan empat input 8-bit X_0, X_1, X_2, X_3 adalah sebagai berikut:

$$Z_0 = (S_1(X_0) \& m_0) \oplus (S_2(X_1) \& m_1) \oplus (S_1(X_2) \& m_2) \oplus (S_2(X_3) \& m_3)$$

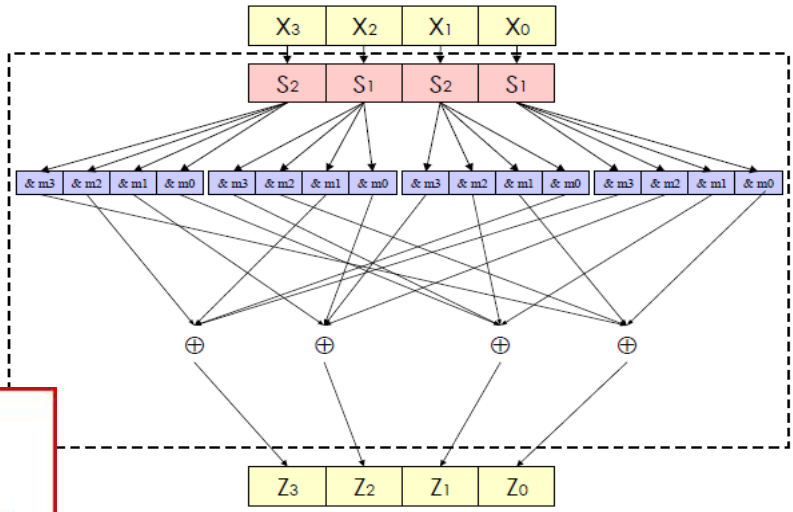
$$Z_1 = (S_1(X_0) \& m_1) \oplus (S_2(X_1) \& m_2) \oplus (S_1(X_2) \& m_3) \oplus (S_2(X_3) \& m_0)$$

$$Z_2 = (S_1(X_0) \& m_2) \oplus (S_2(X_1) \& m_3) \oplus (S_1(X_2) \& m_0) \oplus (S_2(X_3) \& m_1)$$

$$Z_3 = (S_1(X_0) \& m_3) \oplus (S_2(X_1) \& m_0) \oplus (S_1(X_2) \& m_1) \oplus (S_2(X_3) \& m_2)$$

dimana, m_0 = 0xfc, m_1 = 0xf3, m_2 =0xcf dan m_3 = 0x3f.

Struktur fungsi G ditunjukkan pada gambar 2.12.



Gambar 2.12. Fungsi G.

II.4.4. Desain S-Box

Dua S-box S_1, S_2 adalah bagian dari G dan didefinisikan sebagai berikut:

$$S_i : Z_2^8 \longrightarrow Z_2^8,\ S_i(x) = A \bullet x \oplus b_i$$

dimana, n_1=247, n_2=251, b_1=169, b_2=56 dan

$$A^{(1)} = \begin{pmatrix} 1&0&0&0&1&0&1&0 \\ 1&1&1&1&1&1&1&0 \\ 1&0&0&0&0&1&0&1 \\ 0&1&0&0&0&0&1&0 \\ 0&1&0&0&0&1&0&1 \\ 0&0&1&0&0&0&0&1 \\ 1&0&0&0&1&0&0&0 \\ 0&0&0&1&0&1&0&0 \end{pmatrix} ; A^{(2)} = \begin{pmatrix} 0&1&0&0&0&1&0&1 \\ 1&0&0&0&0&1&0&1 \\ 1&1&1&1&1&1&1&0 \\ 0&0&1&0&0&0&0&1 \\ 1&0&0&0&1&0&1&0 \\ 1&0&0&0&1&0&0&0 \\ 0&1&0&0&0&0&1&0 \\ 0&0&0&1&0&1&0&0 \end{pmatrix}$$

Perhatikan bahwa A (i) • $x^n{}_i \oplus$ b adalah transformasi affine dari x^{ni}. Untuk x sembarang dalam Z_2^8, x dapat dinyatakan sebagai bentuk vektor biner x = (x_7, ..., x_0) (yaitu, x = $x_7 2^7 + x_6 2^6 + ... + x_1 2 + x_0$).Digunakan polinomial primitif p (x) = $x^8 + x^6 + x^5 + x + 1$ untuk mewakili x^{ni} dalam Z_2^8.

Untuk meningkatkan efisiensi fungsi G, kami mendefinisikan extended S-boxes (SS-boxes, 4Kbyte)

$$SS_3(X) = S_2(X) \,\&\, m_2 \,||\, S_2(X) \,\&\, m_1 \,||\, S_2(X) \,\&\, m_0 \,||\, S_2(X) \,\&\, m_3$$

$$SS_2(X) = S_1(X) \,\&\, m_1 \,||\, S_1(X) \,\&\, m_0 \,||\, S_1(X) \,\&\, m_3 \,||\, S_1(X) \,\&\, m_2$$

$$(X) = S_2(X) \,\&\, m_0 \,||\, S_2(X) \,\&\, m_3 \,||\, S_2(X) \,\&\, m_2 \,||\, S_2(X) \,\&\, m_1$$

$$SS_0(X) = S_1(X)\,\&\,m_3 \;||\; S_1(X)\,\&\,m_2 \;||\; S_1(X)\,\&\,m_1 \;||\; S_1(X)\,\&\,m_0$$

dan mengimplementasikan fungsi G seperti persamaan di bawah. (Z adalah output 32bit dari fungsi G seperti Z = Z_3 || Z_2 || Z_1 || Z_0)

$$Z = SS_3(X_3) \oplus SS_2(X_2) \oplus SS_1(X_1) \oplus SS_0(X_0)$$

Berikut ini adalah tabel dua S-box S1, S2.

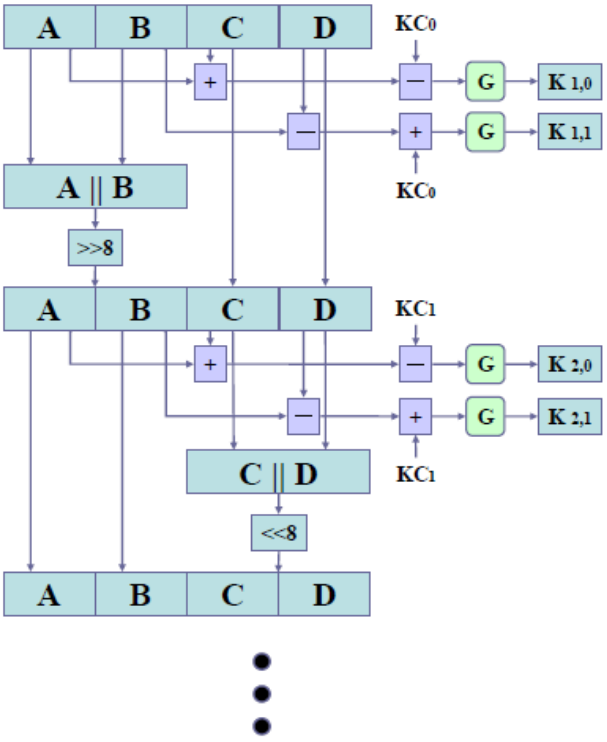
Tabel 2. 1. S-Box S1

i	S1(i)	i	S1(i)	i	S1(i)	i	S1(i)	i	S1(i)	i	S1(i)	i	S1(i)	i	S1(i)
0	169	1	133	2	214	3	211	4	84	5	29	6	172	7	37
8	93	9	67	10	24	11	30	12	81	13	252	14	202	15	99
16	40	17	68	18	32	19	157	20	224	21	226	22	200	23	23
24	165	25	143	26	3	27	123	28	187	29	19	30	210	31	238
32	112	33	140	34	63	35	168	36	50	37	221	38	246	39	116
40	236	41	149	42	11	43	87	44	92	45	91	46	189	47	1
48	36	49	28	50	115	51	152	52	16	53	204	54	242	55	217
56	44	57	231	58	114	59	131	60	155	61	209	62	134	63	201
64	96	65	80	66	163	67	235	68	13	69	182	70	158	71	79
72	183	73	90	74	198	75	120	76	166	77	18	78	175	79	213
80	97	81	195	82	180	83	65	84	82	85	125	86	141	87	8
88	31	89	153	90	0	91	25	92	4	93	83	94	247	95	225
96	253	97	118	98	47	99	39	100	176	101	139	102	14	103	171
104	162	105	110	106	147	107	77	108	105	109	124	110	9	111	10
112	191	113	239	114	243	115	197	116	135	117	20	118	254	119	100
120	222	121	46	122	75	123	26	124	6	125	33	126	107	127	102
128	2	129	245	130	146	131	138	132	12	133	179	134	126	135	208
136	122	137	71	138	150	139	229	140	38	141	128	142	173	143	223
144	161	145	48	146	55	147	174	148	54	149	21	150	34	151	56
152	244	153	167	154	69	155	76	156	129	157	233	158	132	159	151
160	53	161	203	162	206	163	60	164	113	165	17	166	199	167	137
168	117	169	251	170	218	171	248	172	148	173	89	174	130	175	196
176	255	177	73	178	57	179	103	180	192	181	207	182	215	183	184
184	15	185	142	186	66	187	35	188	145	189	108	190	219	191	164
192	52	193	241	194	72	195	194	196	111	197	61	198	45	199	64
200	190	201	62	202	188	203	193	204	170	205	186	206	78	207	85
208	59	209	220	210	104	211	127	212	156	213	216	214	74	215	86
216	119	217	160	218	237	219	70	220	181	221	43	222	101	223	250
[illegible]27		225	185	226	177	227	159	228	94	229	249	230	230	231	178
[illegible]9		233	234	234	109	235	95	236	228	237	240	238	205	239	136
[illegible]2		241	58	242	88	243	212	244	98	245	41	246	7	247	51
[illegible]32		249	27	250	5	251	121	252	144	253	106	254	42	255	154

Tabel *2.2. S-Box S2*

i	S2(i)	i	S2(i)	i	S2(i)	i	S2(i)	i	S2(i)	i	S2(i)	i	S2(i)	i	S2(i)
0	56	1	232	2	45	3	166	4	207	5	222	6	179	7	184
8	175	9	96	10	85	11	199	12	68	13	111	14	107	15	91
16	195	17	98	18	51	19	181	20	41	21	160	22	226	23	167
24	211	25	145	26	17	27	6	28	28	29	188	30	54	31	75
32	239	33	136	34	108	35	168	36	23	37	196	38	22	39	244
40	194	41	69	42	225	43	214	44	63	45	61	46	142	47	152
48	40	49	78	50	246	51	62	52	165	53	249	54	13	55	223
56	216	57	43	58	102	59	122	60	39	61	47	62	241	63	114
64	66	65	212	66	65	67	192	68	115	69	103	70	172	71	139
72	247	73	173	74	128	75	31	76	202	77	44	78	170	79	52
80	210	81	11	82	238	83	233	84	93	85	148	86	24	87	248
88	87	89	174	90	8	91	197	92	19	93	205	94	134	95	185
96	255	97	125	98	193	99	49	100	245	101	138	102	106	103	177
104	209	105	32	106	215	107	2	108	34	109	4	110	104	111	113
112	7	113	219	114	157	115	153	116	97	117	190	118	230	119	89
120	221	121	81	122	144	123	220	124	154	125	163	126	171	127	208
128	129	129	15	130	71	131	26	132	227	133	236	134	141	135	191
136	150	137	123	138	92	139	162	140	161	141	99	142	35	143	77
144	200	145	158	146	156	147	58	148	12	149	46	150	186	151	110
152	159	153	90	154	242	155	146	156	243	157	73	158	120	159	204
160	21	161	251	162	112	163	117	164	127	165	53	166	16	167	3
168	100	169	109	170	198	171	116	172	213	173	180	174	234	175	9
176	118	177	25	178	254	179	64	180	18	181	224	182	189	183	5
184	250	185	1	186	240	187	42	188	94	189	169	190	86	191	67
192	133	193	20	194	137	195	155	196	176	197	229	198	72	199	121
200	151	201	252	202	30	203	130	204	33	205	140	206	27	207	95
208	119	209	84	210	178	211	29	212	37	213	79	214	0	215	70
216	237	217	88	218	82	219	235	220	126	221	218	222	201	223	253
224	48	225	149	226	101	227	60	228	182	229	228	230	187	231	124
232	14	233	80	234	57	235	38	236	50	237	132	238	105	239	147
240	55	241	231	242	36	243	164	244	203	245	83	246	10	247	135
248	217	249	76	250	131	251	143	252	206	253	59	254	74	255	183

II.4.5. Penjadwalan Kunci



Gambar 2.13. Penjadwalan Kunci

Penjadwalan kunci menghasilkan masing-masing subkunci putaran. Ini menggunakan fungsi G, penjumlahan, pengurangan, dan rotasi melingkar (kiri / kanan). Kunci input 128-bit dibagi menjadi empat blok 32-bit (A, B, C, D) dan dua subkunci 32-bit dari putaran 1, $K_{1,0}$ dan $K_{1,1}$, dihasilkan sebagai berikut:

$K_{1,0} = G(A + C - KC_0)$, $K_{1,1} = G(B - D + KC_0)$.

Dua sub kunci 32-bit dari putaran ke-2, $K_{2,0}$ dan $K_{2,1}$ dihasilkan dari kunci input 8-bit dengan rotasi kanan dari 64-bit pertama (A || B) sebagai berikut:

$A||B \leftarrow (A||B)>>8$.

$K_{2,0} = G(A + C – KC_1)$, $K_{2,1} = G(B + KC_1 – D)$.

Dua subkunci dari putaran ke-3, $K_{3,0}$ dan $K_{3,1}$ dihasilkan dari 8-bit rotasi kiri dari 64-bit terakhir (C || D) sebagai berikut:

$C||D \leftarrow (C||D)<<8$.

$K_{3,0} = G(A + C – KC_2)$, $K_{3,1} = G(B – D + KC_2)$.

Sisa dari subkunci dihasilkan secara iteratif. Pseudo kode untuk penjadwalan kunci adalah sebagai berikut :

```
for (i=1; i<=16; i++) {
    Ki,0 ← G(A+C-KCi-1);
    Ki,1 ← G(B-D+KCi-1);
   if (i%2==1) A||B ←(A||B) >>8
   else C||D ←(C||D) <<8
}
```

dimana, setiap konstan KCi dihasilkan dari bagian nomor golden rasio $\frac{\sqrt{5}-1}{2}$.

Tabel 2.3. Konstan KC_i

Konstan dalam bentuk heksadesimal (KCi)			
i	Value	i	Value
	0x9e3779b9	8	0x3779b99e
	0x3c6ef373	9	0x6ef3733c
	0x78dde6e6	10	0xdde6e678

3	0xf1bbcdcc	11	0xbbcdccf1
4	0xe3779b99	12	0x779b99e3
5	0xc6ef3733	13	0xef3733c6
6	0x8dde6e67	14	0xde6e678d
7	0x1bbcdccf	15	0xbcdccf1b

II.4.6. Adopsi Algoritma

SEED telah digunakan secara luas di Korea untuk layanan rahasia seperti perdagangan elektronik dan layanan keuangan, dll. SEED adalah standar asosiasi industri Korea (TTAS.KO-12.0004, 1999).

Hingga September 2003, kode sumbernya di C telah didistribusikan ke 600 perusahaan termasuk akademisi dan lembaga penelitian oleh KISA melalui e-mail. Selain itu, ada beberapa perusahaan internasional dalam sejumlah bisnis, seperti BULL-Korea, RSA Security, IBM-Korea, Entrust-Korea, dll. KISA telah mengupload dokumen terkait di berandanya dan http: // www. kisa.or.kr/seed/.

Penggunaan SEED telah mencakup aplikasi layanan keamanan seperti, e-Commerce, e-mail, penerima khusus dengan Broadcasting, layanan keuangan, penyimpanan data, pengumpulan tol elektronik, VPN, Manajemen Hak Digital, dll.