SoftCrypt

SHAZRINA SAMSUDIN
WEK 000251

Sarjana Muda Sains Komputer
Fakulti Sains Komputer & Teknologi Maklumat
Universiti Malaya
Kuala Lumpur
Sesi 2002/2003

ABSTRAK

SoftCrypt adalah merupakan suatu perisian tunggal yang berdasarkan antaramuka pengguna bergrafik. Ianya dibangunkan dengan objektif untuk menawarkan perlindungan keselamatan maklumat secara umum yang merangkumi pelbagai kategori pengguna komputer. Pengguna bagi perisian ini boleh melakukan enkripsi dan dekripsi dengan menggunakan algoritma DES dan/atau RSA ke atas sebarang maklumat teks yang mereka kehendaki serta menggunakan tandatangan digital yang menggunakan algoritma DSA bagi mengesahkan identiti pengguna tersebut.

Bagi tujuan pembangunan perisian ini, model air terjun dengan prototaip telah digunakan dan bahasa pengaturcaraan Java telah dipilih bagi menghasilkan sistem SoftCrypt. Perisian – perisian lain yang digunakan semasa membangunkan sistem ini ialah Symantec Visual Café 4.0 Expert Edition, Java 2 SDK Versi 1.4.1 dan Notepad.

Sistem ini akan mengandungi lima modul utama iaitu modul *encryption,* modul *decryption,* modul *sign*, modul *verify* dan modul *key manager.* Modul *key manager* akan menguruskan segala yang berkaitan dengan mengimport dan mengeksport kunci – kunci awam pengguna.

SETULUS PENGHARGAAN

Syukur alhamdulillah saya panjatkan ke hadrat Ilahi kerana dengan izin dan limpah kurniaNya saya dapat menyiapkan laporan Projek Ilmiah Tahap Akhir ini. Pada kesempatan ini, ingin saya rakamkan sekalung penghargaan kepada semua pihak yang telah memberi sokongan dan inspirasi di dalam perlaksanaan projek ini.

Di sini juga, saya ingin mengucapkan ribuan terima kasih kepada Puan Azwina Yusof selaku penyelia projek yang sudi mengorbankan masa dan memberi tunjuk ajar kepada saya sepanjang tempoh pembangunan projek ini. Tanpa bimbingan serta sikap bertimbang rasa beliau adalah sukar bagi saya untuk meyiapkan projek seperti yang dikehendaki. Tidak ketinggalan, terima kasih kepada Puan Norazlina Khamis seraya moderator projek yang turut menyumbangkan pandangan serta komen bagi meningkatkan kualiti projek yang dibangunkan.

Untaian kasih yang tidak terhingga untuk kedua ibubapa saya yang tidak jemu memberi dorongan dan mendoakan kejayaan saya. Buat teman – teman sealiran yang banyak memberi cadangan serta pendapat, terima kasih diucapkan. Semoga tuhan sahaja yang dapat membalas jasa kalian.

Akhir kata, saya mengucapkan terima kasih sekali lagi dan setinggi – tinggi penghargaan kepada semua yang terlibat secara langsung mahupun tidak langsung ketika membangunkan projek ini.

SENARAI ISI KANDUNGAN

SENARAI RAJAH

PENGENALAN

1.1 Pengenalan

Seiring dengan kepesatan pembangunan teknologi maklumat dan kecanggihan perkakasan komputer, serangan ke atas keselamatan komputer seringkali berlaku sejak kebelakangan ini. Secara tidak langsung, ini menimbulkan keraguan terhadap keselamatan data yang disimpan di dalam komputer. Ianya juga menyebabkan masalah bersabit keselamatan data menjadi semakin rumit dan sukar untuk dielakkan. Masalah ini bertambah kompleks apabila ianya melibatkan isu penyimpanan maklumat sulit dan sensitif serta proses pemindahan data atau pertukaran maklumat penting menggunakan rangkaian yang kurang selamat seperti Internet.

Oleh yang demikian, semakin ramai pengguna komputer yang mula memberi perhatian terhadap keselamatan data dan maklumat yang disimpan di dalam komputer mereka serta perkhidmatan *e - mail* yang mereka gunakan di dalam aktiviti seharian. Maklumat tersebut perlu dilindungi agar ianya tidak boleh dicapai oleh pengguna yang tidak sah serta menghalang komputer peribadi atau rangkaian persendirian daripada terdedah kepada ancaman – ancaman keselamatan seperti pintasan ke atas komunikasi data, modifikasi data yang tidak sah, penyamaran sebagai pengguna yang sah dan pemalsuan maklumat.

Terdapat pelbagai kaedah kawalan keselamatan yang boleh digunakan bagi memastikan data yang disimpan di dalam komputer selamat. Antaranya ialah penggunaan katalaluan dan kad pintar (*'smart card'*), pemberian hak capaian data kepada pengguna yang sah sahaja, kawalan keselamatan fizikal seperti penggunaan

kunci dan kamera pengawasan bagi memastikan bilik komputer selamat daripada dicerobohi oleh orang luar dan sebagainya. Namun demikian, kaedah – kaedah di atas masih lagi mempunyai kelemahan dan kekurangannya yang tersendiri. Selain daripada itu, ciri – ciri keselamatan yang ditawarkannya adalah tidak memadai bagi menghadapi cabaran serangan keselamatan komputer pada masa kini jika dibandingkan dengan kaedah kawalan keselamatan menggunakan kriptografi yang dapat menjamin keselamatan data dengan sepenuhnya dan secara menyeluruh.

Kriptografi boleh dikatakan sebagai salah satu kaedah yang paling efektif untuk menjamin keselamatan maklumat sensitif yang disimpan di dalam komputer atau memastikan penghantaran maklumat menggunakan Internet adalah selamat. Secara umumnya, kriptografi merujuk kepada teknik – teknik untuk menulis mesej rahsia di mana penerima yang sah sahaja yang boleh menyahkodkan mesej tersebut dan memahaminya. Cara kriptografi berfungsi adalah sama seperti pintu yang telah berkunci di mana ia memerlukan kunci yang spesifik untuk membukanya. Kebiasaannya, kriptografi melibatkan proses enkripsi (*'encryption'*) iaitu penyembunyian maklumat menggunakan kod rahsia dan proses dekripsi ('decryption') iaitu mendapatkan semula maklumat yang telah disembunyikan melalui proses enkripsi. Kriptografi moden boleh dibahagikan kepada dua teknik umum iaitu Kriptografi Kunci Rahsia (*'Secret Key Cryptography'*) dan Kriptografi Kunci Awam (*'Public Key* Cryptography').

Penggunaan enkripsi dan dekripsi menyediakan perlindungan kepada pengguna secara umumnya melalui tiga cara iaitu:

- Mengenalpasti dan mengesahkan identiti pengguna agar hanya pengguna yang sah sahaja yang boleh melakukan transaksi yang melibatkan akaunnya sendiri. Ianya boleh dilakukan dengan menggunakan tandatangan digital iaitu tandatangan elektronik yang menggunakan infrastruktur kunci awam untuk mengesahkan identiti penghantar mesej ataupun identiti orang yang telah menandatangani sesuatu dokumen elektronik.
- Melindungi dan menjamin keselamatan maklumat sulit pengguna.
- Memastikan maklumat tidak diubahsuai semasa melakukan penghantaran maklumat melalui Internet.

Oleh itu, projek ini dilaksanakan dengan tujuan untuk membangunkan satu sistem yang boleh melakukan enkripsi dan dekripsi bagi melindungi dan menjamin keselamatan data secara umum. Sistem itu akan dinamakan sebagai SoftCrypt.

1.2 Pernyataan Masalah

Setiap perisian yang dibangunkan mempunyai tujuannya yang tersendiri dan kebiasaannya tujuan utama sesuatu perisian itu dibangunkan adalah untuk memodelkan penyelesaian bagi masalah – masalah dunia sebenar dan meningkatkan kecekapan pemprosesan sesuatu kerja. Namun, terdapat juga perisian yang dibangunkan dengan tujuan untuk menjadikannya sebagai suatu alternatif kepada penyelesaian – penyelesaian yang telah sedia ada.

Walaupun kini, telah banyak perisian enkripsi yang dihasilkan, timbul pula beberapa masalah lain yang berkaitan dengan pembelian dan pengagihan perisian – perisian tersebut. Di dalam konteks laporan ini, masalah yang dikemukakan adalah berkenaan dengan masalah yang dihadapi oleh pengguna yang berada di Malaysia. Antara masalah tersebut ialah:

- Kebanyakan daripada perisian – perisian enkripsi yang terdapat di pasaran perlu dibeli oleh pengguna dan biasanya urusniaga pembelian perisian tersebut dilakukan di dalam matawang Dollar Amerika maka ini menjadikan kos bagi pembelian sesuatu perisian itu agak mahal bagi pembeli – pembeli yang berada di luar Amerika.
- Sesetengah syarikat – syarikat perisian ada menawarkan perkhidmatan memuaturun perisian enkripsi keluaran syarikat mereka secara percuma melalui penggunaan Internet. Akan tetapi, kerajaan Amerika Syarikat telah mengenakan undang – undang yang ketat berkaitan isu penjualan dan pengeksportan produk – produk kriptografi yang menggunakan algoritma berjenis kuat (*'strong cryptography algorithm'*) ke luar dari Amerika Syarikat

dan ini menyebabkan perkhidmatan tersebut hanya boleh digunakan oleh mereka yang tinggal di Amerika Syarikat sahaja.

- Terdapat juga beberapa pembangun – pembangun perisian persendirian yang telah menulis program – program enkripsi yang boleh dimuaturun melalui Internet secara percuma oleh sesiapa sahaja tanpa mengira di mana mereka berada. Namun demikian, kebanyakan program – program tersebut menggunakan algoritma kriptografi lemah (*'weak cryptography algorithm'*) dan ada kemungkinan sesetengah program itu mengandungi ralat – ralat yang tidak diketahui oleh pembangun di mana ralat tersebut mungkin boleh menjadi suatu ancaman terhadap keselamatan komputer yang menggunakannya. Semuanya ini menyebabkan kebolehpercayaan program - program tersebut berkurangan.

Berdasarkan pernyataan – pernyataan masalah di atas, tujuan utama SoftCrypt dibangunkan adalah seperti berikut:

- Sebagai salah satu alternatif kepada program – program enkripsi yang sedia ada yang boleh diedarkan secara percuma kepada sesiapa sahaja dan SoftCrypt akan menggunakan algoritma kriptografi berjenis kuat.
- Sebagai sampel untuk tujuan pembelajaran bagi pelajar – pelajar yang berminat di dalam bidang kriptografi di mana mereka boleh mencuba sendiri melakukan enkripsi dan dekripsi ke atas data. Pendekatan secara *'hands-on learning'* ini lebih memudahkan para pelajar memahami konsep – konsep berkaitan kriptografi.

- Boleh dijadikan sebagai panduan kepada pembangun perisian yang lain yang ingin membangunkan perisian enkripsi yang baru agar dapat menghasilkan perisian yang lebih baik.
- Boleh digunakan untuk kegunaan peribadi pengguna bagi mengenkripkan maklumat – maklumat yang diingini tanpa mengira maklumat tersebut adalah maklumat sulit ataupun tidak.

1.3 Objektif

Dalam membangunkan SoftCrypt ini, beberapa objektif telah digariskan. Objektif ini akan dijadikan sebagai panduan untuk menghasilkan sistem seperti yang dikehendaki. Antara objektif – objektif yang ingin dicapai adalah seperti berikut:

1) Merekabentuk dan membangunkan aplikasi keselamatan tunggal (*'stand-alone application'*) yang menyediakan perkhidmatan seperti enkripsi, dekripsi serta tandatangan digital dengan menggunakan algoritma kriptografi berjenis kuat iaitu DES, RSA dan/atau DSA.

2) Menghasilkan sistem yang mesra pengguna di mana ia merangkumi aspek – aspek seperti antaramuka yang menarik, sistem yang mudah digunakan serta senang dipelajari bagi semua tahap pengguna.

3) Merekabentuk dan menghasilkan sistem yang membolehkan pengguna melindungi keselamatan data mereka bagi tujuan penyimpanan ataupun semasa penghantaran data.

4) Membangunkan sistem yang boleh digunakan untuk kegunaan peribadi mahupun untuk urusan rasmi dan ianya boleh digunakan oleh sesiapa sahaja termasuklah golongan pelajar.

1.4 Skop

Sebagaimana di dalam pembangunan sistem – sistem yang lain, SoftCrypt juga mempunyai skop yang tersendiri. Skop bagi sistem ini boleh dibahagikan kepada beberapa bahagian iaitu pengguna sasaran, bahasa penghantar dan juga kekangan sistem.

1.4.1 Pengguna Sasaran

Secara umumnya, SoftCrypt sesuai digunakan oleh:

- Pengguna yang mempunyai pengetahuan yang sederhana mengenai bidang komputer.
- Pengguna yang inginkan perisian yang mudah digunakan dan senang dipelajari, kandungan perisian yang ringkas dan tidak membebani pengguna dengan maklumat – maklumat yang terlalu teknikal atau terperinci.
- Pengguna yang inginkan jaminan terhadap keselamatan data yang disimpan di dalam komputer mereka ataupun perlindungan keselamatan data semasa melakukan penghantarannya melalui Internet.
- Pengguna yang ingin menggunakan perkhidmatan tandatangan digital yang mudah tanpa melibatkan penggunaan sijil (*'certificate'*) ataupun pihak pengantara.

1.4.2 Bahasa Penghantar

Bahasa penghantar merupakan salah satu elemen yang penting di dalam pembangunan sesuatu perisian. Bagi tujuan penulisan laporan projek, Bahasa Melayu akan digunakan sebagai bahasa penghantar. Manakala Bahasa Inggeris pula akan digunakan sebagai bahasa penghantar di dalam SoftCrypt serta di dalam penulisan manual pengguna. Bahasa Inggeris dipilih sebagai bahasa penghantar kerana secara umumnya, Bahasa Inggeris adalah merupakan bahasa pertuturan antarabangsa yang difahami dan digunakan di seluruh dunia dan ini dapat memberikan nilai global kepada SoftCrypt dan seterusnya membolehkannya digunakan oleh sesiapa sahaja yang berada di serata dunia.

1.4.3 Kekangan Sistem

- Membuat enkripsi dan dekripsi bagi fail berjenis teks sahaja dengan menggunakan algoritma RSA dan/atau DES. Jenis fail yang boleh disokong oleh sistem ini boleh ditambah pada masa akan datang.
- Menyediakan perkhidmatan tandatangan digital serta pengesahannya yang akan menggunakan algoritma DSA. Akan tetapi, tandatangan digital itu tidak akan dihubungkan dengan pihak *'Certificate Authority (CA)'*.
- Proses penghantaran kunci perlu dilakukan secara manual oleh pengguna kerana sistem ini tidak mempunyai penyambungan kepada Internet atau sebarang *server* dan keselamatan dan kerahsiaan kunci adalah tanggungjawab pengguna untuk memastikannya.

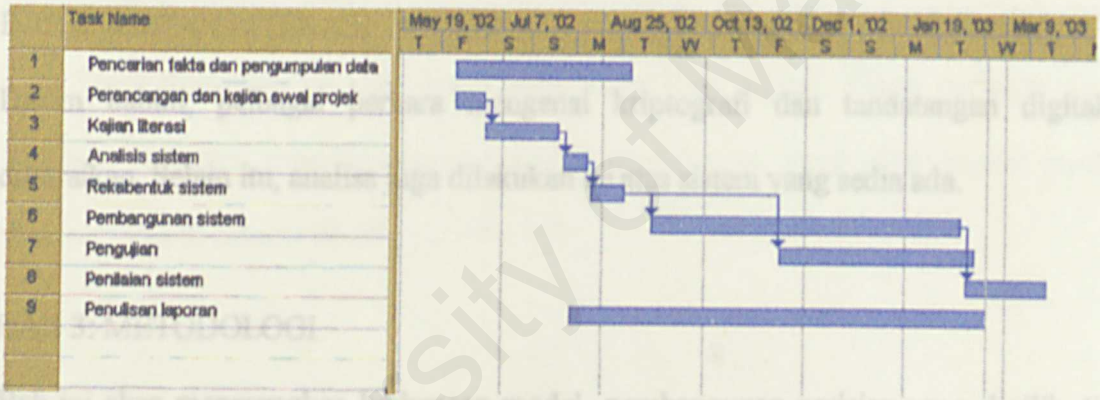
- Setiap pengguna sistem hanya boleh mempunyai satu tandatangan digital berserta tiga pasang kunci sahaja di mana setiap pasang kunci akan mengandungi satu kunci rahsia, satu kunci awam serta satu kunci peribadi
- Pengguna tidak boleh menggunakan kunci yang ditakrifkan sendiri kerana semua kunci akan dijanakan secara automatik oleh sistem SoftCrypt ini.

1.5 Hasil yang Dijangkakan

Dijangkakan di akhir tempoh pembangunan projek ini, satu versi lengkap SoftCrypt dapat dihasilkan. Di mana ianya boleh berfungsi untuk melakukan enkripsi dan dekripsi dengan menggunakan algoritma RSA dan/atau DES serta penggunaan tandatangan digital yang menggunakan algoritma DSA. Selain itu, diharapkan projek ini dapat menghasilkan sistem SoftCrypt yang mesra pengguna.

1.6 Penskedulan Tugas

Perancangan adalah merupakan perkara terpenting di dalam proses pembangunan sesuatu perisian. Perancangan yang rapi serta teliti dari segi pembahagian masa adalah perlu bagi memastikan perisian yang dibangunkan itu dapat disiapkan di dalam tempoh masa yang telah ditetapkan. Carta Gantt digunakan bagi menjadualkan aktiviti – aktiviti yang harus dilakukan pada masa yang ditetapkan di sepanjang tempoh proses pembangunan SoftCrypt seperti yang digambarkan di dalam Rajah 1.1



Rajah 1.1 : Carta Gantt

1.7 Organisasi Tesis

Laporan ini terbahagi kepada 8 bab dan ringkasan bagi setiap bab adalah seperti berikut:

BAB 1: PENGENALAN

Bab ini menerangkan tentang pengenalan kepada keselamatan data dan dunia kriptografi serta sistem SoftCrypt. Ia merangkumi objektif sistem, skop, tujuan sistem dibangunkan dan skedul penbangunannya.

BAB 2: KAJIAN LITERASI

Dalam babini, pelbagai perkara mengenai kriptografi dan tandatangan digital dihuraikan. Selain itu, analisa juga dilakukan ke atas sistem yang sedia ada.

BAB 3: METODOLOGI

Bab ini akan menerangkan berkenaan model pembangunan perisian yang dipilih di dalam membangunkan sistem SoftCrypt ini.

BAB 4: ANALISA SISTEM

Bab ini sangat penting dalam menerangkan konsep utama yang digunakan serta menjadi tunjang utama di dalam menganalisa keperluan sistem.

BAB 5: REKABENTUK SISTEM

Bab ini membincangkan tentang fasa rekabentuk sistem SoftCrypt serta rekabentuk antaramuka yang terdapat di dalam sistem tersebut.

BAB 6: PEMBANGUNAN SISTEM

Bab ini serba sedikit akan membincangkan tentang fasa pembangunan sistem yang melibatkan persekitaran yang digunakan serta hal – hal yang berkaitan dengan pengaturcaraan yang dilakukan semasa membangunkan sistem ini.

BAB 7: PENGUJIAN

Bab ini pula akan membincangkan tentang jenis – jenis pengujian yang dijalankan. Pengujian adalah perlu bagi menghasilkan satu sistem yang kukuh dan bebas ralat.

BAB 8: PENILAIAN SISTEM

Di dalam bab ini akan dibincangkan tentang kelebihan, kekurangan serta peningkatan yang boleh dilakukan ke atas sistem pada masa akan datang. Selain daripada itu, masalah – masalah yang dihadapi serta jalan penyelesaian yang telah diambil turut dibincangkan di dalam bab ini.

KAJIAN LITERASI

2.1 Pengenalan

Kajian literasi adalah merupakan suatu kajian untuk mengumpulkan maklumat berkenaan sistem SoftCrypt yang akan dibangunkan. Pengumpulan maklumat dibuat dengan menyeluruh agar maklumat yang diperolehi dapat dijadikan sebagai garis panduan semasa merangka proses pembangunan sistem tersebut.

Selain itu, kajian serta penilaian juga dilakukan ke atas sistem yang telah dibangunkan di mana sistem yang mempunyai konsep yang lebih kurang sama atau relevan dengan sistem yang hendak dibangunkan. Ianya dilakukan bagi mengenalpasti kelemahan dan kelebihan sistem – sistem yang sedia ada.

2.2 Kriptografi

Penggunaan kriptografi telah bermula sejak terciptanya tulisan dan penggunaannya telah berkembang luas disebabkan oleh keinginan untuk menyimpan rahsia.

2.2.1 Sejarah Ringkas

Sekitar 400 tahun SM, orang Spartan telah menggunakan enkripsi untuk menghantar mesej – mesej ketenteraan. Pembawa mesej akan memakai sejenis tali pinggang khas yang dikenali sebagai *'Scytale'* bagi menyimpan mesej yang perlu dibawa. Mesej tersebut telah ditulis di dalam bentuk himpunan set – set simbol yang tidak bermakna. Penerima perlu menggunakan sejenis silinder khas untuk memaparkan mesej yang asal

Julius Caesar pula telah memperkenalkan '*Caesar Cipher*' yang merupakan salah satu kaedah enkripsi yang menggunakan teknik penggantian huruf. Beliau telah menggantikan huruf demi huruf di dalam teks biasa, tiga kedudukan ke kanan turutan huruf. Ini menyebabkan huruf A menjadi huruf D, huruf B menjadi huruf E, huruf J menjadi huruf M dan seterusnya seperti di dalam Rajah 2.1.

Rajah 2.1: Ceasar Cipher

Di China dan India juga terdapat pelbagai bentuk enkripsi yang telah digunakan. Manakala orang Arab adalah orang yang pertama menemui beberapa kaedah *'cryptanalysis'* iaitu proses mendapatkan semula teks asal daripada teks yang telah dienkripkan tanpa mengetahui kunci yang telah digunakan dan kaedah - kaedah tersebut telah direkodkan ke dalam buku.

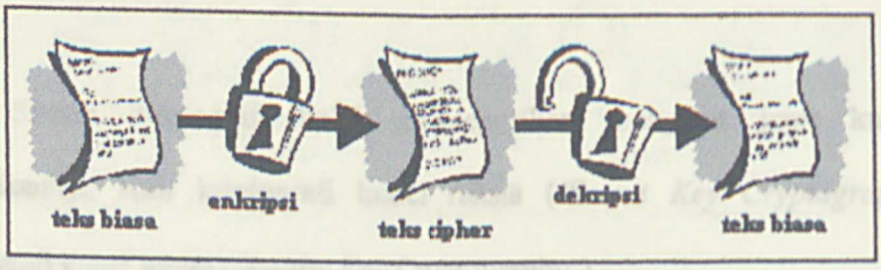
2.2.2 Kriptografi

Perkataan kriptografi berasal daripada perkataan Greek iaitu *'kryptos'* yang bermakna rahsia atau tersembunyi dan *'graphos'* yang bermakna penulisan. Dengan ini, kriptografi bolehlah dikaitkan dengan bidang penulisan rahsia.

Kriptografi sebenarnya ialah suatu cabang sains penggunaan algoritma matematik untuk melakukan enkripsi dan dekripsi ke atas data iaitu proses menukarkan mesej asal kepada mesej yang baru dan sukar difahami dengan menggunakan kunci yang spesifik. Mesej yang baru itu boleh ditukarkan semula ke mesej asal melalui kaedah yang sama.

Kriptografi melibatkan proses enkripsi dan dekripsi ke atas data. Enkripsi atau penyulitan ialah suatu proses untuk menukarkan data, maklumat atau mesej asal yang dikenali sebagai teks biasa (*'plaintext'*) kepada maklumat yang telah dikodkan yang dipanggil sebagai teks *cipher* (*'ciphertext'*). Dekripsi atau penyahsulitan pula adalah proses sebaliknya iaitu menukarkan semula mesej di dalam bentuk teks *cipher* kepada mesej asalnya iaitu di dalam bentuk teks biasa. Jenis dekripsi yang dilakukan

adalah mengikut jenis enkripsi yang digunakan ke atas mesej tersebut. Rajah 2.2 menunjukkan proses enkripsi dan dekripsi yang mudah.



Rajah 2.2: Enkripsi dan Dekripsi

Algoritma kriptografi adalah teknik atau peraturan yang digunakan di dalam proses enkripsi dan dekripsi. Algoritma kriptografi berjenis kuat selalunya dibentuk daripada fungsi – fungsi matematik agar data yang telah dienkrip sukar dipecahkan. Kebiasaannya, algoritma kriptografi akan digunakan bersama kunci di dalam proses enkripsi bagi menghasilkan teks *cipher* yang spesifik.

Menurut ISO/IEC10116 (1997), kunci telah didefinasikan sebagai suatu turutan simbol yang mengawal operasi penukaran mesej di dalam proses kriptografi. Bilangan kunci yang mungkin perlulah besar agar proses dekripsi sukar dilakukan tanpa menggunakan kunci yang sebenar. Saiz kunci diukur di dalam bit. Pertambahan satu bit akan menggandakan bilangan kunci yang mungkin dan bilangan ini akan meningkat secara gandaan. Oleh yang demikian, semakin besar saiz kunci maka semakin sukar untuk memecahkan teks *cipher* kerana bilangan kunci yang mungkin semakin banyak.

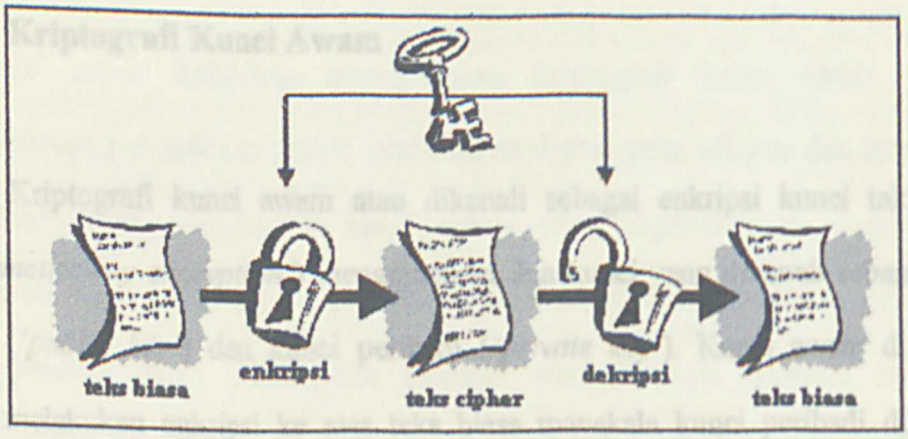
Kesimpulannya, keselamatan data yang telah dienkrip bergantung sepenuhnya kepada kerahsiaan dan saiz kunci yang digunakan serta kekuatan algoritma kriptografi yang menggunakan kunci tersebut.

Sistem kriptografi boleh dikategorikan mengikut jenis kunci yang digunakannya iaitu kriptografi kunci rahsia (*'Secret Key Cryptography'*) dan kriptografi kunci awam (*'Public Key* Cryptography').

2.2.3 Kriptografi Kunci Rahsia

Kriptografi kunci rahsia atau turut dikenali sebagai enkripsi kunci simetri (*'symmetric-key encryption'*) atau enkripsi kunci peribadi (*'private-key encryption'*) hanya menggunakan satu jenis kunci sahaja iaitu kunci rahsia (*'secret key'*) untuk melakukan enkripsi dan dekripsi ke atas mesej. Antara contoh – contoh algoritma yang menggunakan kriptografi kunci rahsia ialah DES, *Triple*-DES, IDEA, RC4 dan sebagainya.

Di dalam kriptografi kunci rahsia, penghantar akan menggunakan kunci rahsia untuk mengenkripkan teks biasa kepada teks *cipher* dan menghantar teks *cipher* tersebut kepada penerima. Penerima kemudiannya akan menggunakan kunci rahsia yang sama untuk mengdekripkan semula mesej tersebut kepada teks biasa. Rajah 2.3 menunjukkan kriptografi kunci rahsia secara umum.



Rajah 2.3: Kriptografi Kunci Rahsia

Maka jelas bahawa, kunci rahsia itu perlu diketahui oleh penghantar dan penerima mesej dan kedua – duanya perlu bersetuju untuk menggunakan algoritma enkripsi yang sama dan menjaga kerahsiaan kunci rahsia yang telah dipersetujui bersama.

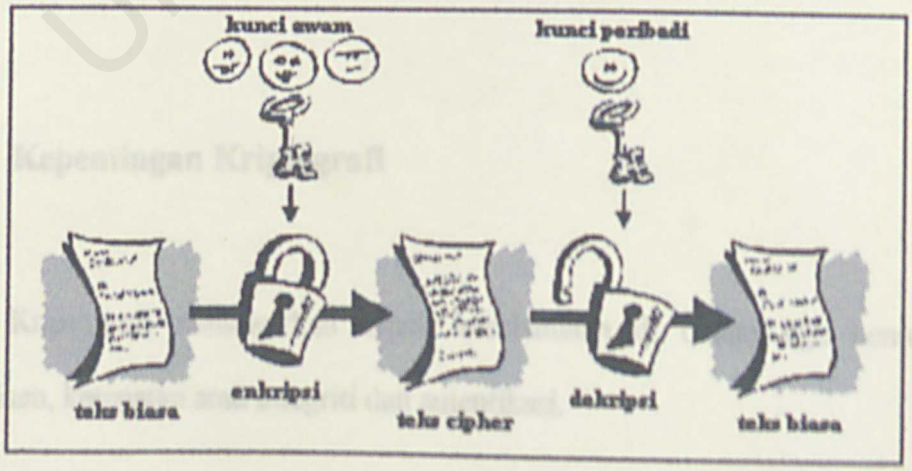
Terdapat dua jenis algoritma enkripsi bagi kriptografi kunci rahsia iaitu algoritma enkripsi yang beroperasi ke atas teks biasa satu bit pada satu masa yang dirujuk sebagai *'stream cipher'* dan algoritma enkripsi yang beroperasi ke atas teks biasa mengikut blok – blok yang mengandungi sekumpulan bit dan ianya dikenali sebagai *'block cipher'*.

Oleh kerana kriptografi kunci rahsia hanya bergantung kepada satu kunci sahaja untuk melakukan enkripsi dan dekripsi, maka masalah utama yang dihadapinya adalah berkenaan pengagihan kunci rahsia dengan selamat. Walaubagaimanapun, perlaksanaan kriptografi kunci rahsia adalah sangat cepat berbanding perlaksanaan kriptografi kunci awam.

2.2.4 Kriptografi Kunci Awam

Kriptografi kunci awam atau dikenali sebagai enkripsi kunci tak simetri (*'asymmetric-key encryption'*) menggunakan dua kunci yang dikenali sebagai kunci awam (*'public key'*) dan kunci peribadi (*'private key'*). Kunci awam digunakan untuk melakukan enkripsi ke atas teks biasa manakala kunci peribadi digunakan untuk melakukan dekripsi ke atas teks *cipher*. Kedua – dua kunci tersebut mempunyai perkaitan secara matematik dan saling melengkapi antara satu sama lain. Antara contoh – contoh algoritma yang menggunakan kriptografi kunci awam ialah Elgamal, RSA, Diffie-Hellman, DSA dan sebagainya.

Di dalam kriptografi kunci awam, penghantar iaitu sesiapa sahaja boleh mengenkripkan mesej dengan menggunakan kunci awam penerima tanpa mengetahui kunci peribadi mereka. Hanya penerima yang mempunyai kunci peribadi yang sah sahaja yang boleh mengdekripkan mesej tersebut. Dengan itu, semua komunikasi hanya melibatkan kunci awam sahaja, tiada kunci peribadi yang perlu dikongsi bersama di antara penghantar dengan penerima. Rajah 2.4 menunjukkan kriptografi kunci awam secara umum.



Rajah 2.4: Kriptografi Kunci Awam

Di antara kelebihan menggunakan kriptografi kunci awam ialah ia membenarkan penggunaan sistem tandatangan digital yang efisyen dan ianya boleh digunakan untuk menghantar kunci rahsia yang digunakan di dalam sistem kriptografi kunci rahsia kepada penerima yang diingini dengan selamat. Penggunaan kunci peribadi yang tidak perlu dihantar atau dikongsi bersama orang lain meningkatkan keselamatan data yang telah dienkripkan. Selain itu, pasangan kunci awam dan kunci peribadi boleh digunakan di dalam jangka masa yang lama tanpa perlu ditukar selalu kecuali apabila kerahsiaan kunci peribadi telah terjejas.

Walaupun demikian, penggunaan kriptografi kunci awam juga mempunyai kekurangannya yang tersendiri. Antaranya ialah perlaksanaan kriptografi kunci awam adalah sangat perlahan kerana ia melibatkan pengiraan dan penggunaan operasi – operasi matematik dan algoritma pengiraannya adalah lebih kompleks berbanding algoritma yang digunakan di dalam kriptografi kunci rahsia. Kekuatan kriptografi kunci awam bergantung kepada kesukaran dan kekompleksan operasi – operasi matematik. Jika pada masa akan datang, penyelesaian bagi masalah – masalah matematik tersebut telah dijumpai, maka keselamatan dan kekuatan kriptografi kunci awam akan terjejas.

2.2.5 Kepentingan Kriptografi

Kriptografi menawarkan aspek keselamatan di dalam tiga bentuk iaitu kerahsiaan, ketepatan atau integriti dan autentikasi.

- Kerahsiaan dan kesulitan data terjamin melalui proses enkripsi kerana ia dapat memastikan tiada siapa yang boleh membaca mesej yang telah dienkripkan melainkan penerima yang sah.
- Ketepatan dan integriti data dapat dipastikan melalui proses enkripsi. Sesetengah jenis algoritma enkripsi dapat melindungi data daripada dipalsukan atau diubah. Algoritma tersebut dapat mengesan sebarang perubahan kecil yang dilakukan ke atas data yang telah dienkripkan.
- Autentikasi pula adalah berkenaan proses mengesahkan identiti seseorang bagi mempastikan pengguna tersebut adalah seperti yang didakwanya dan mesej yang dihantar adalah sah. Teknik autentikasi yang semakin popular digunakan ialah tandatangan digital.

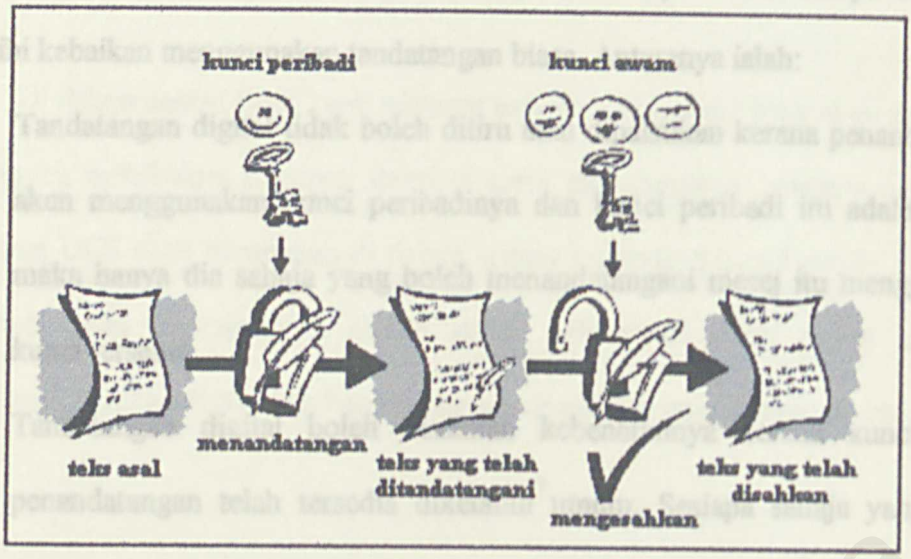
Selain daripada itu, kriptografi juga dapat menyediakan mekanisme untuk membuktikan memang benar penghantar itu telah menghantar mesej tersebut dan ianya dapat menghalang pihak yang tidak berhak daripada mencuri maklumat semasa melakukan proses penghantaran data.

2.3 Tandatangan Digital

Tandatangan digital adalah nilai yang terhasil daripada pengiraan menggunakan kunci peribadi. Ia menunjukkan seseorang yang mempunyai kunci peribadi telah mengesahkan bahawa kandungan mesej tersebut adalah betul dan sah. Tandatangan digital akan menggunakan algoritma kriptografi kunci awam.

Konsep tandatangan digital adalah hampir sama seperti tandatangan biasa kecuali tandatangan biasa mudah dipalsukan, manakala tandatangan digital adalah mustahil untuk dipalsukan dan ditiru. Ini kerana tandatangan digital digunakan bersama untuk membuktikan kandungan maklumat yang telah ditandatangani serta identiti penandatangannya.

Secara amnya, tandatangan digital dapat dihasilkan dengan melakukan enkripsi menggunakan kunci peribadi penghantar ke atas mesej yang ingin dihantar. Jika mesej itu dapat didekripkan dengan betul menggunakan kunci awam penghantar maka sahlah mesej tersebut berasal daripada penghantar berkenaan. Jika terdapat sebarang perubahan di dalam tandatangan digital atau di dalam mesej yang dihantar, maka tandatangan itu tidak dapat lagi digunakan untuk mengesahkan kebenaran mesej tersebut atau identiti penghantarnya. Rajah 2.5 menunjukkan konsep tandatangan digital secara ringkas.



Rajah 2.5: Tandatangan Digital

Langkah – langkah untuk menghasilkan tandatangan digital:

1) Penghantar mengenkripkan mesej menggunakan kunci peribadinya bagi menandatangani mesej tersebut.
2) Kemudian, penghantar akan mengenkripkan mesej tersebut menggunakan kunci awam penerima bagi merahsiakannya daripada diketahui oleh orang lain yang bukan penerima yang sah.

Langkah – langkah untuk mengesahkan tandatangan digital:

1) Penerima mendekripkan mesej yang diterima menggunakan kunci peribadinya bagi mengesahkan tandatangan tersebut dan memastikan mesej yang diterima tidak mengalami sebarang modifikasi.
2) Kemudiannya, penerima akan mendekripkan mesej tersebut menggunakan kunci awam penghantar bagi mempastikan dan mengesahkan penghantar tersebut yang telah menghantar mesej itu.

Penggunaan tandatangan digital mempunyai banyak kebaikannya dan ianya melebihi kebaikan menggunakan tandatangan biasa. Antaranya ialah:

- Tandatangan digital tidak boleh ditiru atau dipalsukan kerana penandatangan akan menggunakan kunci peribadinya dan kunci peribadi itu adalah rahsia maka hanya dia sahaja yang boleh menandatangani mesej itu menggunakan kunci tersebut.
- Tandatangan digital boleh disahkan kebenarannya kerana kunci awam penandatangan telah tersedia diketahui umum. Sesiapa sahaja yang boleh mencapai mesej dan tandatangan itu, boleh mengesahkan bahawa mesej tersebut telah ditandatangani oleh penandatangan itu dan mempastikan mesej serta tandatangan tersebut tidak diubahsuai.
- Penggunaan tunggal tandatangan digital kerana ianya unik mengikut mesej – mesej yang menggunakannya.
- Selepas penandatangan menandatangani mesej tersebut, dan mesej bersama tandatangan itu dihantar kepada penerima yang diingini, penandatangan tidak boleh menyangkal atau menafikan bahawa dia tidak menandatangani mesej tersebut kecuali dapat dibuktikan kunci peribadi penandatangan telah dicuri.
- Mesej yang mempunyai tandatangan digital dianggap telah dimeterai secara digital dan jika ianya diubah, tandatangan itu tidak lagi sah untuk digunakan.

2.4 Algoritma

Di dalam sistem SoftCrypt, terdapat beberapa algoritma yang akan digunakan bagi tujuan melakukan enkripsi, dekripsi serta menghasilkan tandatangan digital. Algoritma DES akan digunakan di dalam sistem kriptografi kunci rahsia, algoritma *RSA* pula akan digunakan di dalam sistem kriptografi kunci awam manakala algoritma DSA akan digunakan bagi tujuan menghasilkan tandatangan digital.

2.4.1 DES

DES atau nama penuhnya *'Data Encryption Standard'* telah dibangunkan di sekitar tahun 1970an oleh kumpulan penyelidik IBM untuk dijadikan sebagai satu piawai di dalam bidang komunikasi dan perlindungan data. DES adalah sejenis *'block cipher'* yang menggunakan blok 64-bit bersama kunci yang bersaiz 56-bit. Oleh kerana DES merupakan suatu sistem kriptografi kunci rahsia, maka algoritma dan kunci yang digunakan untuk enkripsi dan dekripsi adalah sama iaitu ia mengandungi *'initial permutation'* dan *'final permutation'* yang telah dilaksanakan sebanyak 16 kali pusingan. Ini bermakna algoritma utama DES akan diulangi sebanyak 16 kali bagi menghasilkan teks *cipher* yang dikehendaki.

'Initial permutation' akan dilaksanakan ke atas blok 64-bit yang telah dienkripkan. Kemudian, pengiraan dilakukan mengikut kunci yang digunakan dan akhir sekali, *'final permutation'* dilakukan ke atas blok tersebut. *'Initial permutation'* dan *'final permutation'* memerlukan penggunaan blok yang bersaiz 64-bit dan ia

akan menukarkan posisi setiap bit di dalam blok itu mengikut susunan yang telah ditetapkan.

Kunci bagi DES mengandungi 64 digit binari yang mana 56 bit daripadanya akan dijanakan secara rawak dan digunakan terus di dalam algoritmanya manakala 8 bit lagi yang tidak digunakan oleh algoritma itu akan digunakan untuk mengesan ralat dan bit – bit tersebut dikenali sebagai bit pariti. Bit – bit pariti akan dikeluarkan semasa algoritma dan kunci 56-bit digunakan bagi mencipta 16 sub-kunci berbeza yang bersaiz 48-bit. Satu sub-kunci akan dihasilkan bagi setiap pusingan yang dibuat.

Jika kunci 64-bit yang lengkap digunakan dan 56-bit pembolehubah telah dipilih secara rawak maka tiada kaedah lain selain daripada kaedah serangan paksaan (*'brute-force attack'*) yang boleh digunakan untuk mendapatkan kunci yang sebenar. Ianya boleh dilakukan dengan mencuba kesemua 2^{56} kunci yang berkemungkinan.

2.4.2 RSA

RSA adalah singkatan nama yang mewakili tiga orang penciptanya iaitu Ron Rivest, Adi Shamir dan Len Adleman. RSA telah dicipta pada tahun 1977 di MIT. Ciri keselamatan yang ditawarkan oleh RSA bergantung kepada kesukaran memfaktor nombor yang besar kepada nombor perdananya. RSA adalah merupakan suatu sistem kriptografi kunci awam maka ia akan menggunakan dua kunci iaitu kunci awam dan kunci peribadi.

Kunci bagi algoritma RSA akan dijanakan dengan menggunakan pengiraan matematik yang melibatkan pemilihan 2 nombor perdana yang besar (p dan q) dan kedua – dua nombor tersebut akan didarabkan bagi mendapatkan suatu jumlah, n. Nombor – nombor tersebut kemudiannya akan digunakan di dalam satu algoritma matematik bagi menentukan kunci awam dan kunci peribadi RSA. Kelemahan utama RSA adalah tempoh pemprosesannya yang sangat perlahan jika dibandingkan dengan sistem kriptografi kunci rahsia.

2.4.3 DSA

DSA atau *'Digital Signature Algorithm'* yang telah dikeluarkan oleh *'National Institute of Standards and Technology's'* (NIST) adalah merupakan sistem kriptografi kunci awam yang hanya digunakan untuk tujuan tandatangan digital sahaja. Saiz bagi kunci DSA yang selalu digunakan adalah di dalam lingkungan 512-bit sehingga 1024-bit.

Cara algoritma DSA digunakan ialah:

1) Penghantar mesej akan mengenkripkan mesej yang ingin dihantar dengan menggunakan kunci peribadi DSA kepunyaannya dan menghantar mesej tersebut kepada penerima yang diingini.
2) Penerima mesej itu pula akan mendekripkan mesej itu menggunakan kunci awam penghantar mesej.
3) Jika penerima mendapati mesej tersebut adalah silap dan tidak dapat difahami maka terdapat kemungkinan mesej tersebut bukan dihantar oleh penghantar mesej yang asal atau mesej itu telah diubah semasa proses penghantarannya.

2.5 Bahasa Pengaturcaraan

Di dalam membangunkan sistem SoftCrypt ini, terdapat dua pilihan bahasa pengaturcaraan yang boleh digunakan iaitu Java dan Microsoft Visual C++.

2.5.1 Java

Java telah dibangunkan oleh Sun Microsystems di sekitar tahun 1995. Antara komponen – komponen Java yang boleh digunakan bagi tujuan kriptografi ialah *'Java Cryptography Architecture'* (JCA) dan *'Java Cryptography Extension'* (JCE).

JCA akan menerangkan rekabentuk kelas – kelas kriptografi secara keseluruhannya yang merangkumi konsep kriptografi serta algoritmanya. JCA telah dibina bagi mengasingkan konsep kriptografi daripada proses implementasinya. Konsep kriptografi tersebut akan digunakan oleh kelas – kelas yang terdapat di dalam pakej *java.security* dan pakej *javax.crypto*. Proses implementasi kriptografi tersebut akan dibekalkan oleh pembekal kriptografi (*'cryptographic providers'*).

Oleh kerana kerajaan Amerika Syarikat telah mengenakan sekatan ke atas pengeksportan algoritma – algoritma kriptografi yang tertentu, ini menyebabkan Sun telah membahagikan kelas – kelas kriptografi kepada dua kumpulan. Kumpulan pertama dimasukkan ke dalam pakej *java.security.** yang merupakan sebahagian daripada JDK 1.2. Kelas – kelas kriptografi di dalam kumpulan ini boleh dieksport ke luar Amerika Syarikat tanpa dikenakan sebarang sekatan. Kumpulan kedua iaitu

komponen JCE 1.2 adalah untuk digunakan di dalam Amerika Syarikat dan Kanada sahaja.

JCE adalah lanjutan kepada JCA dan ianya merupakan suatu set pakej – pakej yang menyediakan rangka kerja dan implementasi bagi enkripsi, penjanaan kunci dan persetujuan kunci serta perlaksanaan algoritma *'Message Authentication Code'* (MAC). JCE juga turut menyokong enkripsi berjenis simetri, enkripsi tak simetri, *block cipher'* dan *'stream cipher'*. Pada masa kini, komponen JCE 1.2.1 telahpun diintegrasikan ke dalam *'Java 2 SDK v 1.4'*, di mana ianya boleh digunakan di luar Amerika Syarikat tanpa sebarang sekatan.

Di dalam Java, pembekal kriptografi yang sedia ada dikenali sebagai SUN dan SunJCE. Namun demikian, Java juga membenarkan penggunanya menambah dan mentakrifkan sendiri pembekal kriptografi yang akan membekalkan algoritma yang ingin digunakan bagi kelas – kelas konsep kriptografi yang lain. Oleh yang demikian, pakej perpustakaan kriptografi seperti pakej Bouncy Castle Crypto ataupun Cryptix JCE yang disediakan oleh Cryptix boleh digunakan bagi menampung algoritma - algoritma kriptografi yang tidak terdapat di dalam Java.

2.5.2 Microsoft Visual C++

Microsoft Visual C++ (VC++) adalah salah satu produk keluaran Microsoft yang merupakan lanjutan kepada bahasa pengaturcaraan C++. Selain daripada Microsoft CryptoAPI, model kriptografi rangka kerja .NET (*'.NET Framework*

Cryptography Model') turut menyediakan kelas – kelas bagi mengimplementasikan fungsi kriptografi di dalam VC++.

Kelas asas yang digunakan untuk menyimpan kelas – kelas kriptografi yang lain dikenali sebagai *System.Security.Cryptography.* Kelas asas ini menyediakan perkhidmatan kriptografi yang biasa termasuklah mengkodkan dan menyahkodkan data, *hashing*, penjanaan nombor rawak serta autentikasi mesej.

Operasi – operasi kriptografi pula akan dilaksanakan oleh modul yang berasingan yang dikenali sebagai '*Cryptographic Service Providers'* (CSPs). Setiap CSP menyediakan implementasi yang berlainan bagi lapisan *Cryptography API.* Sesetengahnya menyediakan algoritma kriptografi yang kuat manakala ada juga sesetengah CSP yang boleh berkomunikasi dengan pengguna secara langsung contohnya semasa proses penggunaan tandatangan digital.

2.6 Analisis Sistem Sedia Ada

Di dalam bahagian ini, dua perisian telah dipilih untuk dianalisis iaitu Cypherus dan Stealth Encryptor. Tujuan analisis ini dilakukan adalah untuk:

- Mendapatkan pandangan umum dan membandingkan fungsi – fungsi yang terdapat di dalam perisian yang telah dibangunkan oleh pembangun sistem yang lain.
- Mendapatkan idea bagaimana untuk membangunkan SoftCrypt.
- Menambah dan memperbaiki fungsi – fungsi yang terdapat pada SoftCrypt.

2.6.1 Cypherus

Cypherus adalah merupakan suatu perisian yang menggunakan algoritma *Blowfish* yang mempunyai kunci bersaiz di dalam lingkungan 128-bit hingga 448-bit. Cypherus ini akan melakukan enkripsi dan dekripsi yang melibatkan koleksi - koleksi fail ('archive') di mana satu senarai fail boleh dienkripkan menjadi satu koleksi fail. Rajah 2.6 menunjukkan antaramuka bagi Cypherus yang memaparkan fail – fail yang telah dienkripkan ke dalam satu koleksi yang dinamakan sebagai *ttry.cry*.

Rajah 2.6: Cypherus

Selain daripada itu, Cypherus juga turut menyediakan beberapa perkhidmatan yang lain iaitu:

- *Autoencryptaur* yang melakukan enkripsi dan dekripsi ke atas sekumpulan fail yang memerlukan keselamatannya dijaga sepanjang masa walaupun fail – fail itu selalu dibuka dan diedit.
- *Text Encryptaur* yang berfungsi seperti *Notepad* bagi membolehkan pengguna menaip dan mengenkripkan teks tersebut.
- *Make EXE* adalah fungsi untuk menjadikan fail yang telah dienkripkan sebagai fail yang boleh dilarikan oleh pengguna yang mempunyai katalaluan yang betul.

Antara kelebihan yang terdapat pada Cypherus ialah:

- Mempunyai pengurus kunci yang akan menguruskan katalaluan – katalaluan pengguna dan menyimpan kunci - kunci penguna.
- Pengguna mempunyai pilihan samada untuk menggunakan kunci yang telah dijanakan oleh sistem, menakrifkan kunci sendiri atau menggunakan kunci yang sedia ada yang disimpan oleh pengurus kunci.
- Mempunyai antaramuka yang menarik dan boleh ditukar mengikut citarasa pengguna.
- Mempunyai *'wizard'* yang boleh membantu pengguna melakukan enkripsi dan dekripsi.
- Membenarkan penghantaran *e – mail* yang telah dienkripkan.
- Perisian yang menyokong bilangan pengguna yang ramai dan setiap pengguna boleh mempunyai katalaluan lebih daripada satu.

Walaubagaimanapun, Cypherus juga mempunyai kekurangannya iaitu:

- Perisian ini perlulah dibeli dengan harga $49.95 USD iaitu bersamaan dengan RM189.90.
- Mempunyai menu New dan Open tetapi ianya hanya boleh digunakan untuk membuka *'archive'* yang berjenis *.cry dan memaparkan senarai nama fail yang telah dienkripkan sahaja.
- Tidak mempunyai *menu bar* yang biasa digunakan menyebabkan pengguna yang kurang mahir agak terganggu dan terkeliru semasa menggunakan perisian ini.
- Hanya menyediakan satu jenis algoritma sahaja bagi tujuan enkripsi dan dekripsi iaitu algoritma *Blowfish*.

2.6.2 Stealth Encryptor

Stealth Encryptor adalah perisian pengguna tunggal yang boleh digunakan untuk melakukan enkripsi dan dekripsi ke atas fail dengan menggunakan 4 pilihan kaedah iaitu:

- Menggunakan algoritma *Blowfish* bersama kunci 128-bit.
- Menggunakan algoritma DES bersama kunci 64-bit.
- Menggunakan kaedah '*Media Stealth*' bersama kunci 40-bit.
- Menggunakan kaedah '*Fast Encryption*' bersama kunci 64-bit.

Rajah 2.7 menunjukkan antaramuka bagi perisian Stealth Encryptor.

Rajah 2.7: Stealth Encryptor

Antara kelebihan yang terdapat pada Stealth Encryptor ialah:

- Membuat salinan fail sebelum ianya dienkripkan.
- Mempunyai mekanisme penguncian skrin bagi perisian ini apabila pengguna hendak meninggalkan sekejap komputer.
- Fail yang dienkrip boleh disembunyikan di dalam bentuk imej.
- Pengguna boleh menaip sendiri kunci yang ingin digunakan atau meminta perisian menjanakan kunci tersebut.
- Menyokong operasi *'drag and drop'* semasa melakukan enkripsi dan dekripsi.

Kekurangan yang terdapat pada Stealth Encryptor pula ialah:

- Perisian ini perlu dibeli dengan harga $44.95 USD iaitu bersamaan dengan RM170.80.
- Pengguna perlu mengingati atau menyalin kunci yang telah dijanakan oleh perisian ini
- Perisian ini hanya boleh digunakan oleh seorang pengguna dan ianya menyokong penggunaan satu katalaluan sahaja.
- Agak sukar bagi pengguna untuk bertukar dari perisian ini ke perisian *Windows* yang lain kerana antaramuka perisian ini memenuhi keseluruhan ruang skrin komputer.

METODOLOGI

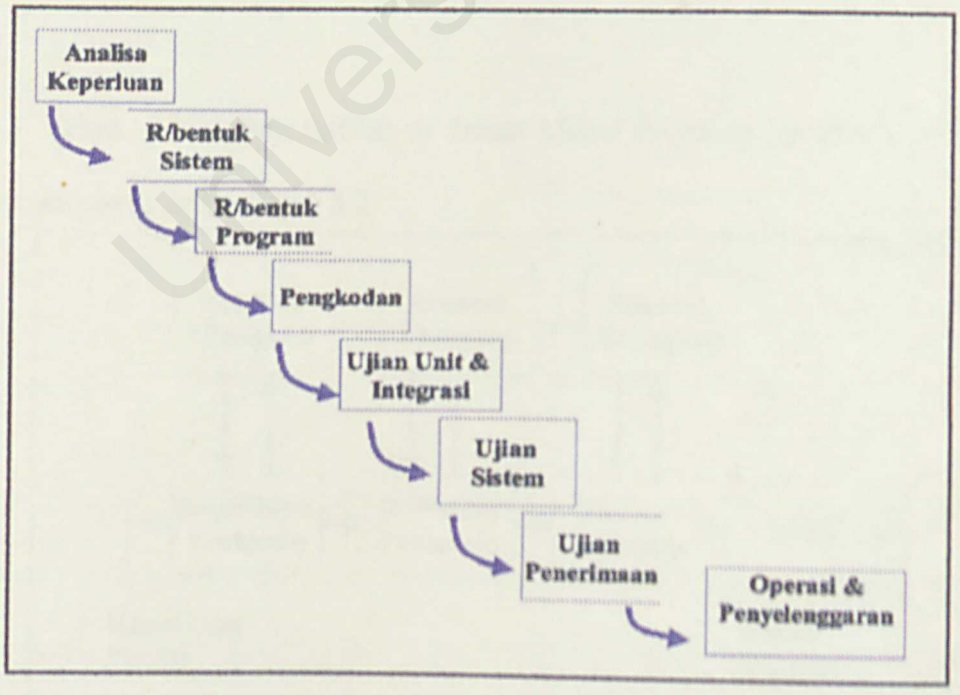
3.1 Pengenalan

Metodologi melibatkan kajian dan perlaksanaan cara dan prosedur dalam membentuk sesuatu sistem. Dalam kajian ini, beberapa metodologi akan dikaji dan metodologi yang bersesuaian akan digunakan untuk membangunkan sistem SoftCrypt. Metodologi yang bersesuaian diperlukan untuk mengambarkan dengan jelas setiap fasa pembangunan sebelum ia dimulakan dan ianya akan dijadikan sebagai panduan sepanjang tempoh pembangunan SoftCrypt ini.

3.2 Model Air Terjun

Model Air Terjun telah diperkenalkan buat pertama kalinya oleh Royce pada tahun 1970. Model ini merupakan satu proses pembangunan yang telah ditakrifkan dengan jelas dan lengkap di mana setiap fasa perlu diselesaikan perlaksanaannya sebelum beralih ke fasa seterusnya. Model ini sangat mudah digunakan dan ianya boleh digunakan jika keperluan sistem telah difahami dan ditakrifkan dengan jelas dan lengkap.

Model Air Terjun telah membahagikan proses perisian kepada beberapa fasa seperti yang ditunjukkan di dalam Rajah 3.1. Antara fasa – fasa yang terdapat di dalam Model Air Terjun ini ialah fasa Analisa Keperluan, Rekabentuk Sistem, Rekabentuk Program, Pengkodan, Ujian Unit dan Integrasi Ujian Sistem Ujian Penerimaan dan akhir sekali ialah fasa Operasi dan Penyelenggaraan.



Rajah 3.1: Model Air Terjun

Antara kelebihan – kelebihan menggunakan Model Air Terjun ini ialah:

- Memudahkan pembangun mengesan kesilapan dan keabaian sistem.
- Memudahkan sasaran proses jangkamasa pembangunan sistem.
- Memudahkan pembangun mengasingkan satu fasa pembangunan dengan fasa pembangunan yang lain.
- Memudahkan kerja – kerja mendokumentasikan sistem.

Kelemahan – kelemahan yang terdapat pada Model Air Terjun pula:

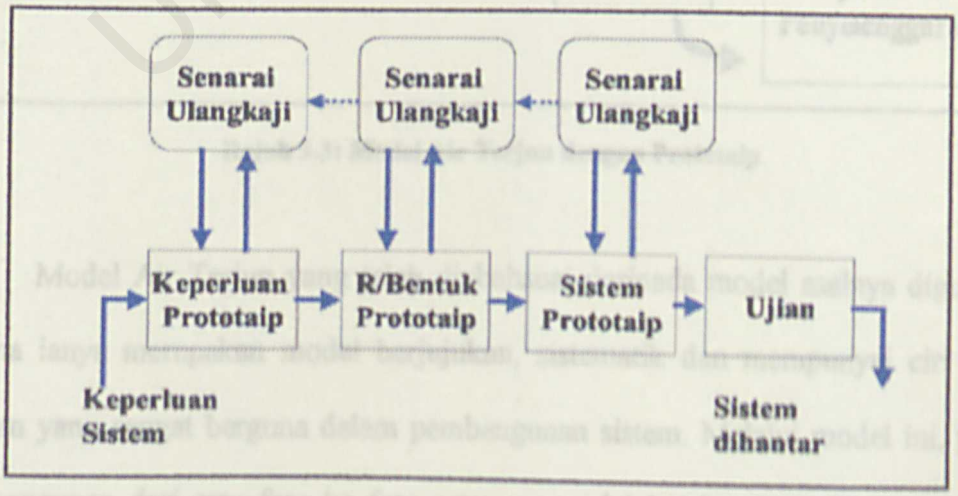
- Tidak menggambarkan cara kod dihasilkan.
- Tidak menyediakan panduan untuk mengendalikan sebarang perubahan yang berlaku pada produk dan aktiviti.
- Pengubahsuaian yang dibuat semasa pembangunan memasuki fasa lain akan menyebabkan perulangan fasa yang sebelumnya berlaku.
- Fasa – fasa pembangunan seringkali dilangkau dan menyebabkan proses pembangunan menjadi tidak teratur.

3.3 Model Prototaip

Model Prototaip adalah satu model pembangunan perisian yang membenarkan pembangun sistem membentuk keseluruhan atau sebahagian daripada sistem yang hendak dibangunkan itu dengan cepat bagi memahami isu – isu yang timbul supaya pembangun sistem dan pengguna mempunyai tahap pemahaman yang sama mengenai sistem tersebut. Ia digunakan untuk mengesahkan keperluan sistem adalah lengkap dan bersesuaian. Terdapat dua jenis pendekatan bagi Model Prototaip ini iaitu:

- Pemprototaipan *'Throwaway'*: objektif pemprototaipan jenis ini ialah untuk memahami keperluan pengguna dan membina sistem mengikut keperluan yang dinyatakan.
- Pemprototaipan Penilaian: objektif pemprototaipan jenis ini ialah untuk bekerjasama dengan pengguna bagi menerokai keperluan mereka dan seterusnya menghasilkan sistem akhir.

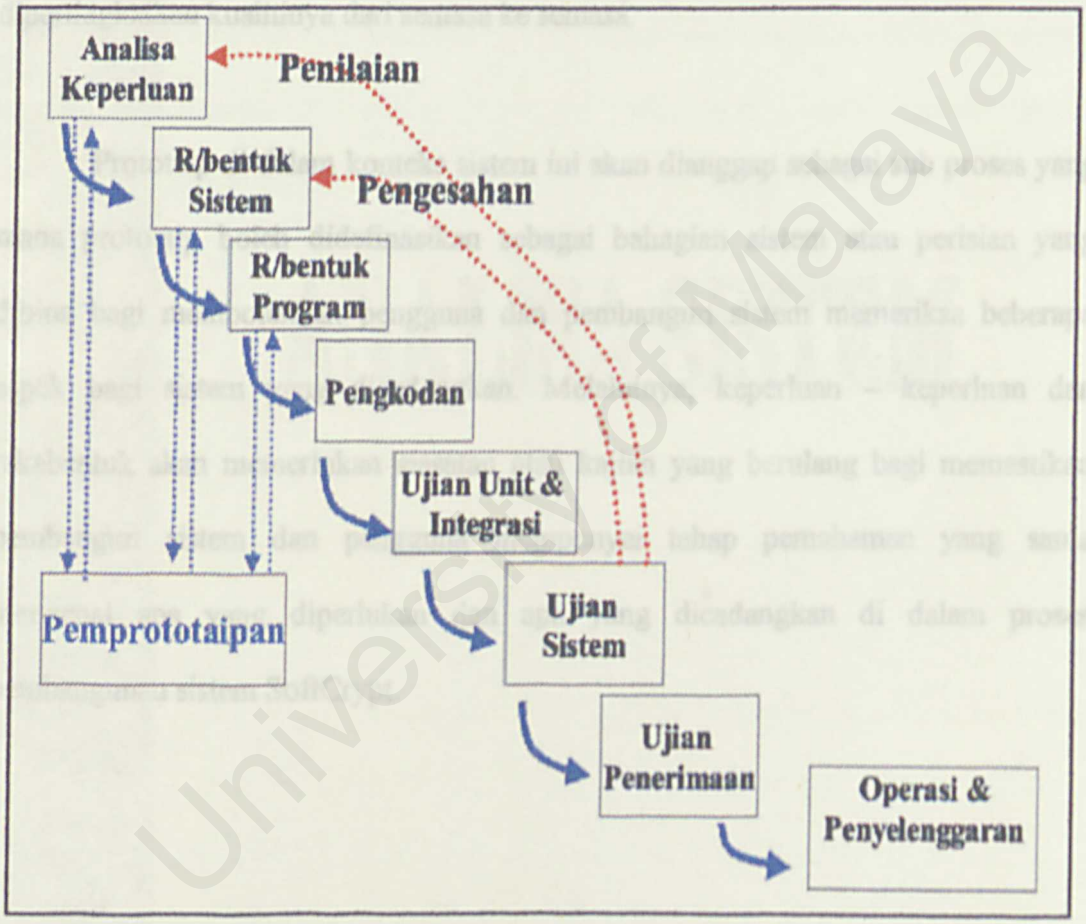
Fasa – fasa yang terlibat di dalam Model Prototaip ini adalah seperti yang ditunjukkan di dalam Rajah 3.2



Rajah 3.2: Model Prototaip

3.4 Model Air Terjun dengan Prototaip

Sistem SoftCrypt ini akan dibangunkan dengan menggabungkan pendekatan Model Air Terjun dengan Model Prototaip. Pembangunan sistem akan melalui semua fasa iaitu analisis dan keperluan sistem, rekabentuk, perlaksanaan, integrasi serta pengujian dan penyelenggaraan seperti yang terdapat di dalam Rajah 3.3.



Rajah 3.3: Model Air Terjun dengan Prototaip

Model Air Terjun yang telah diubahsuai daripada model asalnya digunakan kerana ianya merupakan model berjujukan, sistematik dan mempunyai ciri – ciri kitaran yang sangat berguna dalam pembangunan sistem. Melalui model ini, proses pembangunan dari satu fasa ke fasa seterusnya adalah jelas dan sekiranya berlaku

kesilapan di dalam sesuatu fasa, ianya boleh diperbetulkan semula tanpa perlu menunggu sehingga fasa seterusnya siap. Selain itu, model ini juga digunakan secara meluas oleh pembangun – pembangun sistem.

Prototaip sistem yang dibangunkan pada fasa tertentu akan diuji bagi memastikan sistem memenuhi keperluan yang telah ditetapkan sebagaimana yang dikehendaki oleh pengguna. Seterusnya, prototaip akan diperbaiki dan dipertingkatkan kualitinya dari semasa ke semasa.

Prototaip di dalam konteks sistem ini akan dianggap sebagai sub proses yang mana prototaip boleh didefinasikan sebagai bahagian sistem atau perisian yang dibina bagi membolehkan pengguna dan pembangun sistem memeriksa beberapa aspek bagi sistem yang dicadangkan. Melaluinya, keperluan – keperluan dan rekabentuk akan memerlukan siasatan atau kajian yang berulang bagi memastikan pembangun sistem dan pengguna mempunyai tahap pemahaman yang sama mengenai apa yang diperlukan dan apa yang dicadangkan di dalam proses pembangunan sistem SoftCrypt.

ANALISA SISTEM

4.1 Pengenalan

Analisa sistem merupakan salah satu fasa yang penting dalam sesebuah proses pembangunan perisian. Ianya perlu dilakukan bagi memahami sistem yang bakal dibangunkan dengan lebih teliti. Pada peringkat permulaan, keperluan kefungsian dan keperluan bukan kefungsian ditakrifkan. Ini diikuti dengan analisa mengenai bahasa pengaturcaraan, perisian dan perkakasan yang akan digunakan. Kebiasaannya, kesemua maklumat yang diperlukan dikumpul daripada buku rujukan, jurnal dan termasuklah sumber – sumber yang diambil dari Internet semasa menjalankan kajian literasi.

4.2 Teknik Pengumpulan Maklumat

Terdapat pelbagai teknik yang telah digunakan bagi mengumpulkan segala maklumat yang berkaitan dengan pembangunan SoftCrypt di dalam fasa ini. Pencarian dan pengumpulan fakta adalah salah satu keperluan yang penting dalam memahami dengan lebih jelas akan sistem yang ingin dibangunkan. Teknik pengumpulan maklumat yang digunakan di sini termasuklah pembacaan buku – buku rujukan, pelayaran dalam Internet, penggunaan borang soal selidik dan perbincangan.

1) Pembacaan : kaedah pembacaan merupakan kaedah utama yang digunakan di dalam kajian ini. Bahan – bahan pembacaan adalah di dalam bentuk buku, artikel, majalah, kamus dan sebagainya dan kebanyakan daripada bahan – bahan tersebut telah diperolehi dari Perpustakaan Utama Universiti Malaya.
2) Melayari Internet : pelayaran dalam Internet adalah merupakan kaedah yang agak berkesan dalam mendapatkan sebarang maklumat yang dikehendaki. Kebanyakan bahan yang didapati melalui kaedah ini adalah lebih terkini dan mengikut arus teknologi masa kini.
3) Borang soal selidik : borang soal selidik telah diedarkan kepada beberapa kelompok individu bagi mendapatkan maklumat tambahan, anggapan umum, cadangan serta maklum balas mereka terhadap sistem yang akan dibangunkan ini.
4) Perbincangan : perbincangan yang dilakukan bersama penyelia projek adalah bertujuan untuk memastikan wujudnya persefahaman pendapat serta mengenalpasti beberapa aspek – aspek penting seperti objektif sistem, keperluan kefungsian dan sebagainya. Perbincangan ini dilakukan dari

semasa ke semasa bagi mendapatkan panduan dan bimbingan agar sistem yang dibangunkan berjalan dengan lancar.

4.3 Analisa Keperluan

Keperluan sistem merupakan satu perkara yang penting dan perlu dipertimbangkan dengan teliti semasa membangunkan sesuatu sistem. Analisa keperluan merangkumi keperluan kefungsian dan keperluan bukan kefungsian bagi sistem SoftCrypt. Keperluan kefungsian merupakan perkhidmatan sistem yang dijangkakan oleh pengguna sistem. Bagi keperluan bukan kefungsian pula, ianya melibatkan definasi properti sistem dan kekangan operasian sistem.

4.3.1 Keperluan Kefungsian

Keperluan kefungsian menunjukan apa yang harus dilakukan oleh sesuatu sistem bagi membolehkan sistem berjalan dengan lancar dan teratur. Di dalam SoftCrypt terdapat beberapa modul yang telah dikenalpasti. Modul – modul tersebut adalah seperti berikut:

- Modul *Encryption*: Modul yang penting di dalam SoftCrypt kerana ia mengendalikan segala urusan berkenaan enkripsi termasuklah fungsi yang berkaitan dengan pemilihan algoritma - algoritma enkripsi iaitu RSA ataupun DES.
- Modul *Decryption*: Sama seperti proses enkripsi, proses dekripsi juga adalah proses yang penting di dalam perisian SoftCrypt. Dengan menggunakan modul ini, pengguna boleh melakukan dekripsi ke atas data atau maklumat – maklumat yang telah dienkripkan.

- Modul *Sign*: Modul yang membenarkan pengguna meletakkan tandatangan digital mereka di dalam maklumat atau mesej yang mereka kehendaki.
- Modul *Verify*: Modul yang menguruskan hal – hal yang berkaitan dengan pengesahan tandatangan digital yang terdapat di dalam mesej – mesej yang diterima oleh pengguna.
- Modul *Key Manager*: Modul ini pula akan menguruskan hal – hal yang berkaitan dengan pengurusan kunci bagi algoritma DES, RSA dan DSA serta ianya juga akan menyimpan maklumat berkaitan enkripsi yang telah dilakukan oleh pengguna.

4.3.2 Keperluan Bukan Kefungsian

Di antara keperluan – keperluan bukan kefungsian bagi sistem SoftCrypt pula ialah menyediakan antaramuka yang menarik, mudah digunakan dan difahami serta memastikan keselamatan sistem terjamin.

Sistem ini akan menampilkan antaramuka yang memenuhi ciri – ciri kebolehgunaan bagi meningkatkan keberkesanan sistem tersebut. Antaramuka yang baik haruslah menarik dan konsisten. Kebolehgunaan bermaksud pengguna mudah untuk melakukan tugas – tugas mereka dengan berkesan. Antaramuka sistem ini juga akan disusun dengan teratur agar ia mudah difahami oleh pengguna serta untuk mengurangkan risiko berlakunya kesilapan kemasukan maklumat.

Bagi memenuhi keperluan keselamatan, pengguna perlu melalui proses identifikasi sebelum dibenarkan menggunakan sistem ini. Hanya pengguna yang

berdaftar dan telah memasukkan katalaluan yang betul sahaja akan dibenarkan masuk ke dalam sistem dan menggunakan perkhidmatan yang disediakannya.

4.4 Pemilihan Bahasa Pengaturcaraan

Bagi membangunkan sistem SoftCrypt ini, bahasa pengaturcaraan yang telah dipilih ialah Java dan ianya akan digunakan bersama perisian visual yang dikenali sebagai Symantec Visual Café 4.0 Expert Edition bagi memudahkan penulisan dan penghasilan program Java. Bahasa pengaturcaraan Java telah dipilih kerana ia mempunyai banyak kelebihannya. Antaranya ialah:

- Perpustakaan bagi algoritma kriptografi yang diperlukan adalah mencukupi dan ianya boleh dilengkapkan dengan menggunakan pakej perpustakaan Java yang lain seperti pakej Bouncy Castle Crypto ataupun Cryptix JCE.
- Pengguna dibenarkan mentakrif atau menambah pembekal kriptografinya sendiri
- Menyediakan pakej – pakej dan komponen antaramuka bergrafik yang lengkap.

4.5 Keperluan Sistem

Keperluan perkakasan dan perisian ini penting bagi memastikan sistem yang dibina dapat berfungsi dengan baik dan lancar. Berikut merupakan keperluan perkakasan dan perisian yang dicadangkan untuk mengimplementasikan sistem SoftCrypt ini.

Keperluan perkakasan:

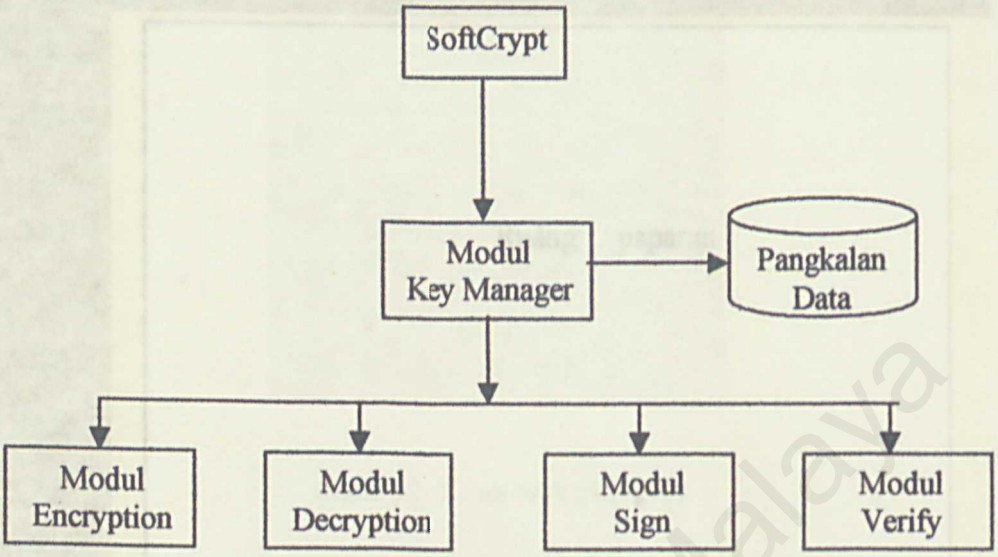
- Mikropemproses Pentium 533 MHz
- 64 MB RAM
- 1.44" FDD
- Monitor
- Windows 95/98/2000

Keperluan perisian:

- Symantec Visual Café 4.0 Expert Edition
- Java 2 SDK Versi 1.4.1

REKABENTUK SISTEM

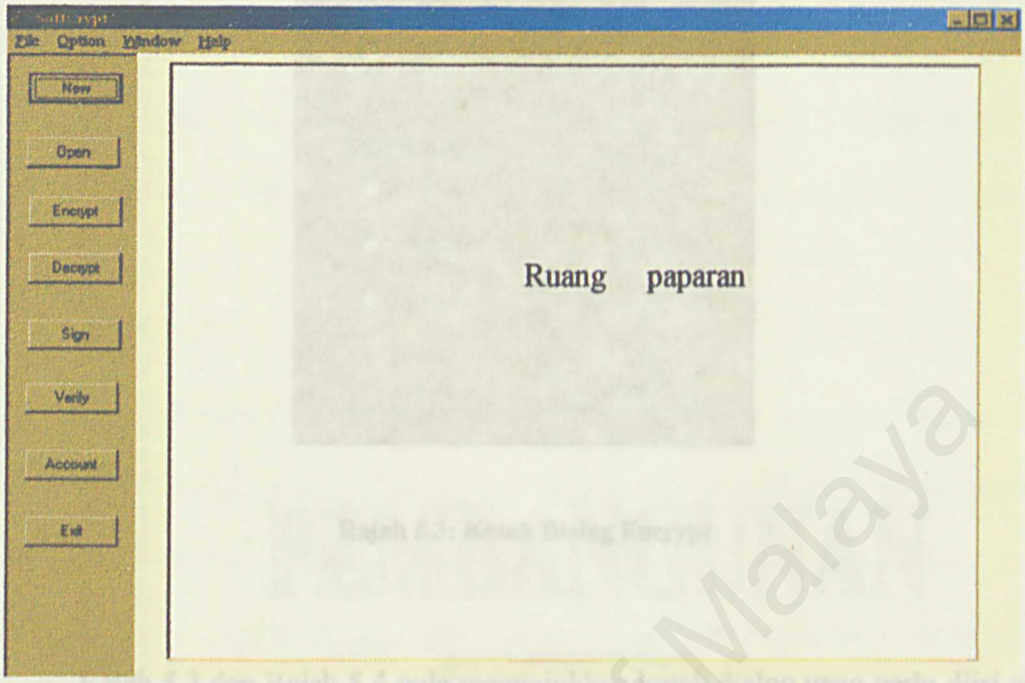
5.1 Rekabentuk Struktur Sistem



Rajah 5.1: Senibina Struktur SoftCrypt

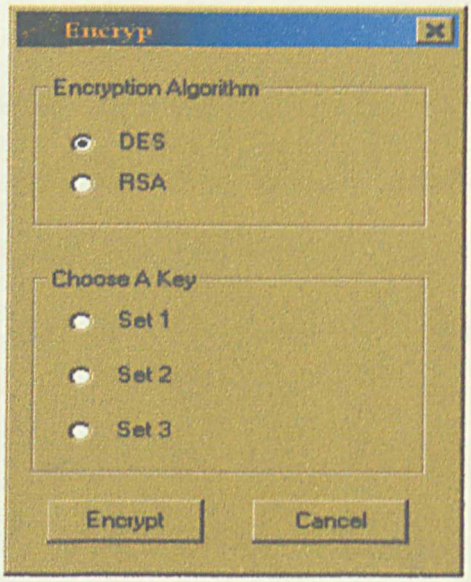
Rajah 5.1 menunjukkan struktur senibina bagi sistem SoftCrypt ini. Di mana Modul Key Manager akan dihubungkan terus dengan sebuah pangkalan data yang akan menyimpan kunci – kunci pengguna. Kunci – kunci tersebut akan digunakan di dalam Modul Encryption, Modul Decryption, Modul Sign dan Modul Verify.

5.2 Rekabentuk Antaramuka



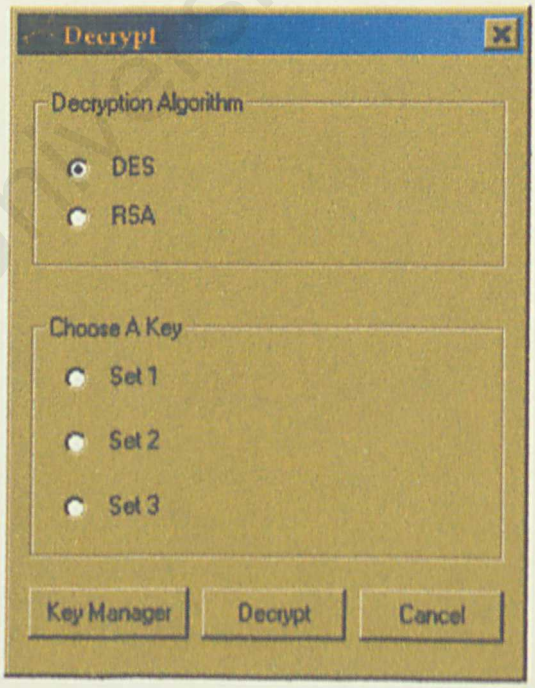
Rajah 5.2: Antaramuka Sistem SoftCrypt

Rajah 5.2 menunjukan antaramuka utama yang akan ditemui oleh pengguna setelah selesai *'login'* ke dalam sistem SoftCrypt ini. Di sini, pengguna boleh terus menaip teks yang ingin dienkripkan di dalam ruang paparan teks yang tersedia kosong. Pengguna juga boleh membuka fail untuk dienkripkan dengan menekan butang Open. Fail yang telah dibuka akan dipaparkan di dalam ruang paparan teks. Bagi melakukan enkripsi, dekripsi, menurunkan tandatangan digital serta mengesahkan tandatangan tersebut, ianya boleh dilaksanakan dengan menekan butang – butang Encrypt, Decrypt, Sign atau Verify. Butang Account pula, boleh digunakan untuk memaparkan butir – butir berkenaan senarai kunci yang dimiliki oleh pengguna tersebut



Rajah 5.3: Kotak Dialog Encrypt

Rajah 5.3 dan Rajah 5.4 pula menunjukkan kotak dialog yang perlu diisi oleh pengguna semasa hendak melakukan enkripsi dan dekripsi.



Rajah 5.4: Kotak Dialog Decrypt

PEMBANGUNAN SISTEM

6.1 Pengenalan

Pembangunan sistem banyak melibatkan aktiviti - aktiviti pengaturcaraan. Pengaturcaraan merupakan suatu proses menterjemahkan logik – logik setiap spesifikasi aturcara yang telah dikenalpasti dan ditakrifkan semasa fasa rekabentuk sistem ke dalam bentuk kod – kod arahan di dalam bahasa pengaturcaraan yang telah dipilih.

Selain daripada itu, persekitaran pembangunan sistem yang telah dipilih juga memainkan peranan yang tidak kurang pentingnya. Persekitaran pembangunan sistem merangkumi pemilihan dan penggunaan perisian serta perkakasan semasa membangunkan sistem. Penggunaan perisian dan perkakasan yang bersesuaian dengan sistem yang dibangunkan adalah penting supaya ia dapat memenuhi dan menyokong keperluan sistem yang dibangunkan.

6.2 Persekitaran Pembangunan Sistem

Persekitaran pembangunan sistem boleh dibahagikan kepada dua bahagian iaitu perkakasan dan perisian yang digunakan semasa membangunkan sistem. Berikut merupakan senarai perkakasan dan perisian yang telah digunakan semasa membangunkan sistem SoftCrypt ini.

Perkakasan yang telah digunakan semasa membangunkan SoftCrypt terdiri daripada sebuah komputer mudah-alih serta sebuah Unit Pemprosesan Pusat (CPU) tambahan. CPU tambahan diperlukan bagi menyokong pemprosesan komputer mudah-alih tersebut:

Komputer mudah-alih *('Laptop')*:

- Mikropemproses Intel Celeron 566 Mhz
- 64 MB RAM
- 4.0 GB ruang cakera keras
- Pemacu CD-ROM 24x
- Pemacu cakera liut
- Komponen – komponen lain yang terdapat pada komputer biasa

Unit Pemprosesan Pusat (CPU) tambahan yang digunakan:

- Mikropemproses Intel Pentium 4 1.6GHz
- 256Mb RAM
- 20.0 GB ruang cakera keras

Perisian yang telah digunakan semasa membangunkan SoftCrypt:

- Sistem Pengoperasian Windows 98
- Symantec Visual Café 4.0 Expert Edition – pengedit visual bagi java
- Java 2 SDK Versi 1.4.1 (JDK)
- JCE Versi 1.2.1 – pakej perpustakaan kriptografi yang merupakan pakej tambahan kepada JDK
- Pakej Bouncy Castle Crypto – pakej perpustakaan java yang menyokong kelas - kelas kriptografi
- Cryptix 3.0 – pakej perpustakaan java yang turut menyediakan implementasi bagi kelas – kelas kriptografi
- Notepad dan Wordpad – pengedit bagi kod – kod aturcara java
- Microsoft Visual J++
- IconCoolEditor – pengedit bagi ikon yang digunakan di dalam SoftCrypt
- SmartDraw – pengedit grafik

6.3 Pendekatan Pembangunan Sistem

Kemahiran pengaturcaraan yang baik dapat menghasilkan sistem yang mudah difahami dan diselenggarakan. Kebiasaannya, pendekatan kepada teknik pengaturcaraan yang baik melibatkan beberapa perkara seperti berikut:

- Kebolehbacaan di mana kod – kod aturcara mudah dibaca dan difahami oleh pengaturcara yang lain, pemilihan nama pembolehubah yang bersesuaian, komen di dalam kod aturcara serta penyusunan keseluruhan kandungan aturcara.
- Teknik penamaan iaitu nama – nama yang diberi kepada pembolehubah, struktur kawalan dan modul dapat menyediakan identifikasi yang mudah kepada pengaturcara.
- Dokumentasi dalaman yang merujuk kepada komen yang ditulis di dalam kod aturcara dapat dijadikan sebagai panduan untuk memahami aturcara tersebut.

Kaedah pengaturcaraan pula boleh dibahagikan kepada 2 kaedah umum iaitu pengaturcaraan bermodul dan pengaturcaraan berstruktur. Pengaturcaraan bermodul melibatkan pembahagian satu masalah yang kompleks kepada bahagian – bahagian yang kecil agar mudah ianya diaturcarakan dan difahami. Dengan itu, kekompleksan sesuatu sistem itu juga dapat diatasi ataupun dikurangkan. Pengaturcaraan berstruktur pula melibatkan kaedah pengaturcaraan yang teratur dan mengikut turutan tertib.

Memandangkan Java merupakan salah satu bahasa pengaturcaraan berorientasikan objek di mana ia menekankan penggunaan konsep kelas dan objek di

dalam kod aturcaranya, maka kaedah pengaturcaraan secara bermodul lebih sesuai digunakan semasa membangunkan sistem SoftCrypt.

Oleh yang demikian, pada peringkat awal pembangunan SoftCrypt, beberapa kod aturcara tunggal telah ditulis bagi mensimulasikan fungsi – fungsi asas yang utama di dalam SoftCrypt seperti operasi enkripsi dan dekripsi menggunakan algoritma DES, RSA dan DSA, pengurusan kunci – kunci kriptografi dan sebagainya. Oleh kerana kod aturcara tersebut ditulis hanya untuk mensimulasikan fungsi yang asas sahaja maka kebanyakan kod – kod aturcara itu ditulis di dalam bentuk yang ringkas dan mudah, tidak mempunyai antaramuka bergrafik dan ianya perlu dilarikan di dalam persekitaran baris arahan *('command line environment')*.

Tujuan utama kod aturcara tersebut ditulis adalah untuk menyediakan asas atau tapak kepada sistem SofyCrypt dan mensimulasikan fungsi – fungsi penting di dalam SoftCrypt. Antara fungsi – fungsi penting tersebut ialah pemprosesan enkripsi dan dekripsi bagi algoritma DES, RSA dan DSA, pengurusan penyimpanan katalaluan pengguna SoftCrypt, pengurusan penyimpanan kunci – kunci kriptografi dan pengurusan kunci awam bagi algoritma RSA dan DSA.

Setiap kod aturcara tersebut telah ditulis dengan menggunakan Notepad dan dilarikan secara berasingan. Setelah dipastikan kod – kod aturcara itu dapat dilarikan dengan betul tanpa ralat dan berfungsi seperti yang dikehendaki, kod aturcara tersebut akan dikompil bersama – sama dan diintegrasikan di dalam pengedit java bergrafik iaitu Symantec Visual Café. Kod – kod aturcara itu akan digabungkan

untuk membentuk satu rangkakerja antaramuka bergrafik yang boleh dianggap sebagai titik permulaan bagi sistem SoftCrypt.

Kemudiannya, kod – kod aturcara yang telah diintegrasikan itu akan dilakukan pengubahsuaian dan penambahan fungsi – fungsi lain ke atasnya bagi memenuhi dan melengkapi keperluan sistem SoftCrypt. Setelah selesai menambahkan fungsi – fungsi yang lain, semua komponen akan diuji dan dilarikan sebagai satu sistem tunggal iaitu sistem bagi SoftCrypt.

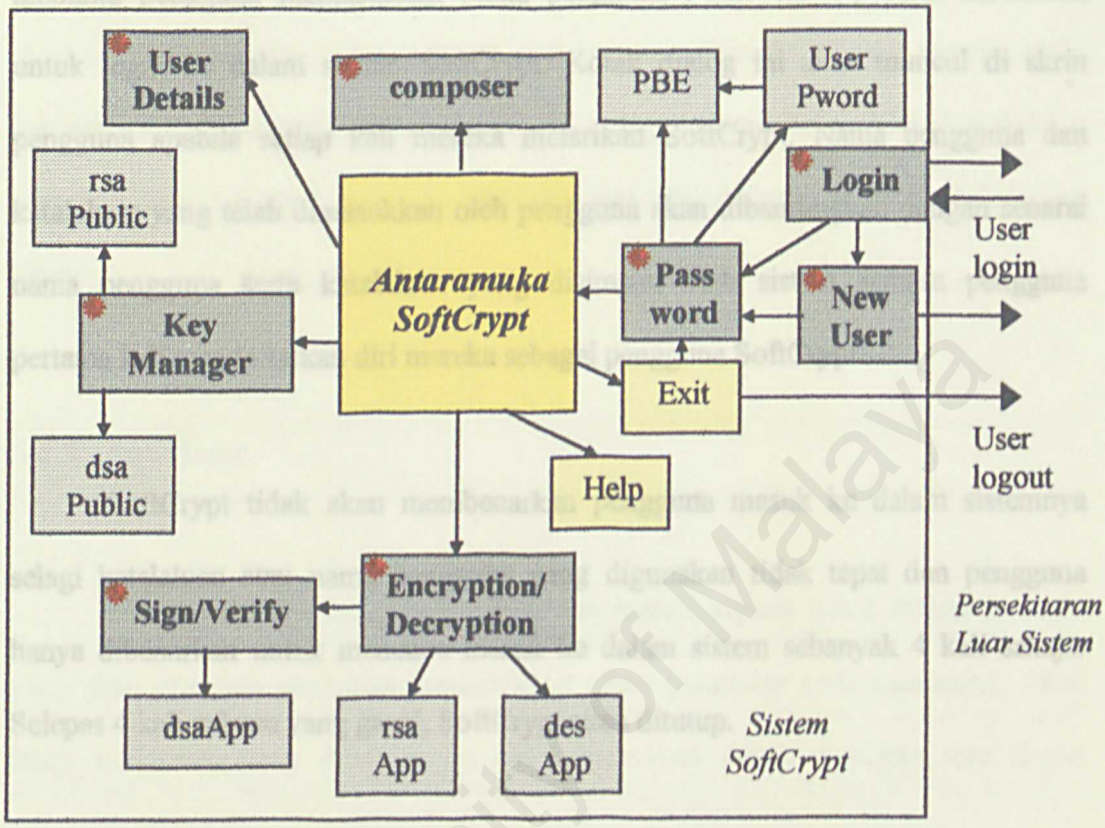
6.4 Implementasi Sistem

Seperti yang pernah dinyatakan sebelum ini, SoftCrypt mempunyai 5 modul utama iaitu Modul *Key Manager*, Modul *Encryption*, Modul *Decryption*, Modul *Sign* dan Modul *Verify*. Bagi melaksanakan fungsi – fungsi yang dilakukan oleh modul di atas, modul – modul tersebut telah dihuraikan atau diterjemahkan kepada lapan komponen utama bagi antaramuka SoftCrypt. Lapan komponen tersebut adalah sebenarnya merupakan kelas – kelas java yang akan membentuk objek – objek atau komponen – komponen yang diperlukan semasa melarikan sistem SoftCrypt. Penggunaan pangkalan data pada awalnya yang bertujuan untuk menyimpan kunci – kunci kriptografi telah digantikan dengan penggunaan kelas – kelas java yang telah disediakan di dalam pakej JDK

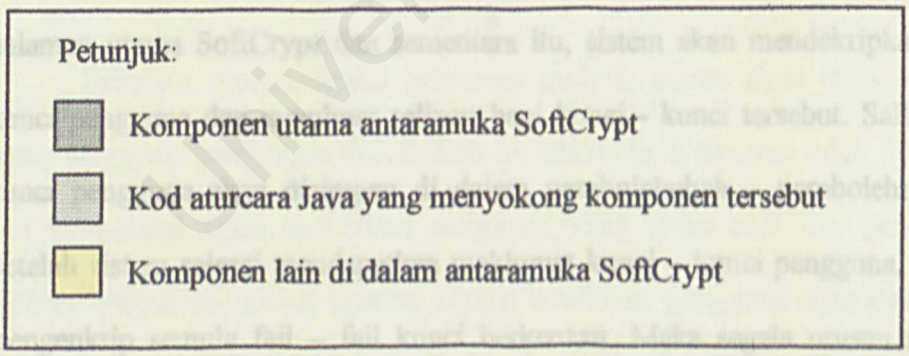
Rajah 6.1 menunjukkan perhubungan di antara komponen – komponen yang terdapat di dalam SoftCrypt. Komponen - komponen yang terdapat di dalam rajah tersebut boleh terdiri daripada rujukan ke panel skrin SoftCrypt (java.awt.Panel), butang pada skrin SoftCrypt (java.awt.Button), rangka SoftCrypt (java.awt.Frame), kotak dialog SoftCrypt (java.awt.Dialog), program yang boleh dilarikan *('executable program'- .exe)* atau merujuk kepada kelas - kelas java (*.java),

Merujuk kepada Rajah 6.1, pengguna hanya boleh memasuki sistem SoftCrypt melalui komponen *Login* sahaja manakala untuk keluar dari sistem ini, terdapat beberapa cara. Jika pengguna telah berjaya memasuki ke dalam sistem SoftCrypt di mana mereka telah melepasi proses *login* maka pengguna hanya boleh keluar dari sistem melalui komponen *Exit* sahaja tetapi, jika pengguna masih belum

lagi melepasi proses *login* maka pengguna boleh keluar dari sistem dengan menekan butang *Cancel* yang terdapat pada kotak dialog *Login* ataupun pada kotak dialog *New User*.



Rajah 6.1: Perhubungan di antara komponen – komponen utama SoftCrypt



Berikut adalah merupakan penerangan mengenai 8 komponen yang penting di dalam sistem SoftCrypt.

6.4.1 Login

Komponen *Login* adalah merupakan sejenis kotak dialog java.awt yang akan meminta pengguna memasukkan nama pengguna *('user name')* serta katalaluan untuk *login* ke dalam sistem SoftCrypt. Kotak dialog ini akan muncul di skrin pengguna apabila setiap kali mereka melarikan SoftCrypt. Nama pengguna dan katalaluan yang telah dimasukkan oleh pengguna akan dibandingkan dengan senarai nama pengguna serta katalaluan yang disimpan oleh sistem semasa pengguna pertama kali mendaftarkan diri mereka sebagai pengguna SoftCrypt.

SoftCrypt tidak akan membenarkan pengguna masuk ke dalam sistemnya selagi katalaluan atau nama pengguna yang digunakan tidak tepat dan pengguna hanya dibenarkan untuk mencuba masuk ke dalam sistem sebanyak 4 kali sahaja. Selepas 4 kali cubaan yang gagal, SoftCrypt akan ditutup.

Jika pengguna berjaya masuk ke dalam sistem, mereka akan dibawa ke halaman utama SoftCrypt dan sementara itu, sistem akan mendekripkan fail – fail kunci pengguna dan membuat salinan bagi kunci - kunci tersebut. Salinan kunci – kunci pengguna akan disimpan di dalam pembolehubah – pembolehubah sistem. Setelah sistem selesai mendapatkan maklumat kunci – kunci pengguna, sistem akan mengenkrip semula fail – fail kunci berkenaan. Maka segala urusan mengenkrip, mendekrip, menandatangani fail ataupun mengesahkan tandatangan akan dilakukan dengan menggunakan salinan kunci yang telah dibuat oleh sistem. Ini dilakukan bagi memastikan fail – fail kunci penggguna sentiasa berada di dalam bentuk yang telah dienkrip dan fail – fail tersebut hanya akan didekrip apabila perlu sahaja. Kaedah ini

juga digunakan ke atas senarai nama pengguna SoftCrypt dan katalaluannya di mana senarai tersebut akan disimpan di dalam bentuk yang telah dienkripkan dan ianya akan didekripkan apabila perlu sahaja.

Bagi pengguna yang pertama kali menggunakan SoftCrypt, mereka perlu mendaftarkan diri mereka ke dalam sistem SoftCrypt. Ianya boleh dilakukan dengan menekan butang *'Create New User'* dan pengguna akan dibawa ke kotak dialog New User.

6.4.2 New User

Komponen New User juga adalah merupakan sejenis kotak dialog java.awt yang akan meminta pengguna memasukkan nama pengguna serta katalaluan. Akan tetapi maklumat yang diterima itu akan digunakan untuk mencipta satu akaun pengguna yang baru.

Sebelum sesuatu akaun pengguna dicipta, sistem akan memeriksa samada nama pengguna yang ingin didaftarkan itu adalah unik ataupun tidak. Istilah unik di sini bermaksud tiada dua nama pengguna yang sama bagi dua pengguna yang berbeza. Aspek ini adalah penting kerana katalaluan pengguna akan dirujuk melalui nama pengguna. Jika terdapat dua nama pengguna yang sama maka sistem tidak dapat mengenalpasti katalaluan manakah yang perlu diambil untuk dibuat perbandingan semasa proses *login*. Selain itu, sistem juga akan memeriksa panjang katalaluan pengguna. Panjang minimum katalaluan ialah sebanyak 4 aksara manakala panjang maksimum adalah sebanyak 8 aksara.

Jika maklumat yang dimasukkan oleh pengguna tiada sebarang masalah, sistem akan menyalin nama pengguna dan katalaluannya ke dalam fail yang dinamakan sebagai "users.skm". Fail tersebut merupakan satu senarai nama pengguna dan katalaluan mereka yang telah berdaftar dengan SoftCrypt dan sistem akan menggunakan fail tersebut untuk membuat perbandingan semasa proses *login*. Selepas sistem selesai mengemaskinikan senarai nama pengguna, sistem akan menyetikakan satu objek dari kelas PBE dan objek itu akan digunakan untuk mengenkrip fail "users.skm" tadi.

Kemudiannya, sistem akan mencipta satu *folder* baru yang dinamakan dengan nama pengguna dan menjanakan satu set kunci bagi pengguna itu. Satu set kunci terdiri daripada satu kunci rahsia DES yang bersaiz 56 bit, sepasang kunci awam RSA iaitu satu kunci awam dan satu kunci peribadi serta sepasang kunci awam DSA. Setiap kunci RSA dan DSA di atas adalah bersaiz 1024 bit. Fail – fail kunci yang dihasilkan akan disimpan di dalam *folder* pengguna itu. Akhir sekali, sebelum berpindah ke halaman utama SoftCrypt, sistem akan memasukkan nama pengguna, kunci awam RSA dan kunci awam DSA pengguna ke dalam senarai kunci awam RSA *(rsaPublic)* dan senarai kunci awam DSA *(dsaPublic)*.

6.4.3 Password

Komponen Password terdiri daripada dua kod aturcara java yang mengandungi kelas yang dikenali sebagai userPword dan PBE. Kelas userPword adalah merupakan suatu kelas yang mengimplementasikan kelas *Properties* yang terdapat di dalam pakej java.util.*.

Menurut Sun Microsystems Inc (2002), kelas *Properties* adalah untuk mewakili satu set properti atau item – item yang perlu disimpan untuk jangka masa yang lama. Properti tersebut boleh disimpan ke dalam fail dan kemudian fail itu boleh dibaca semula apabila ianya hendak digunakan.

Kelas userPword akan digunakan untuk menguruskan segala urusan yang berkaitan dengan senarai nama pengguna SoftCrypt dan katalaluan yang sepadan dengan nama pengguna. Di dalam kelas ini, satu objek *Properties* akan dicipta dan dikenali sebagai *table*. *table* inilah yang akan dirujuk sebagai senarai nama pengguna SoftCrypt dan katalaluan yang sepadan dengan nama pengguna.

```
private Properties table = new Properties();
Object value = table.setProperty (String nameStr, String valueStr);
```

Setiap objek atau jadual *Properties* akan menerima dan menyimpan data bagi sepasang nilai *String* sahaja. Nilai *String* pertama akan dijadikan sebagai kunci yang boleh digunakan untuk merujuk, mencari dan mendapatkan nilai *String* yang kedua. *table* di dalam kelas userPword akan menggunakan nama pengguna sebagai kunci manakala nilai yang dipegang oleh setiap kunci itu adalah merupakan katalaluan pengguna.

Antara metod atau fungsi – fungsi yang telah disediakan di dalam kelas userPword ialah fungsi untuk menambah pengguna baru ke dalam table, fungsi untuk memadam data pengguna yang terdapat di dalam table, fungsi untuk

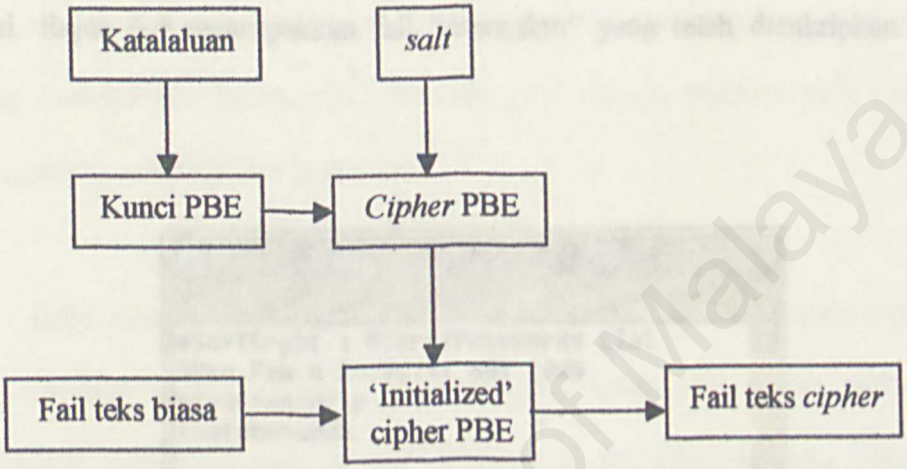
mendapatkan katalaluan pengguna mengikut nama pengguna yang diberi, fungsi untuk menukar katalaluan pengguna dan fungsi untuk membaca dan menulis kandungan table ke dalam suatu fail. Fail itu akan dinamakan sebagai "users.skm" di mana sambungan *.skm merujuk kepada fail yang berkaitan dengan SoftCrypt Key Manager. Berikut adalah contoh bagi sebahagian daripada fungsi – fungsi di atas:

```
// create new user
public boolean put (String name, String pvalue)
{  // ...
   nameStr = name;
   Object value = table.setProperty(nameStr, pvalue);
   return true;
   // ...
}

// get user's password
public String get (String name)
{  // ...
   nameStr = name;
   Object value = table.getProperty(nameStr);
   // ...
   return valueStr;
}

// save table contents to a file
```

```
public void save ( )
{ // …
   FileOutputStream output = new FileOutputStream("users.skm");
   table.store( output, "SoftCrypt : Users/Passwords List" );
   output.close();
   // …
}

// read contents from file
public void load ( )
{ // …
   FileInputStream input = new FileInputStream( "users.skm" );
   table.load( input );
   input.close();
   // …
}
```

Kelas PBE pula adalah merupakan suatu kelas yang melakukan enkripsi atau dekripsi ke atas suatu fail mengikut katalaluan yang telah dibekalkan. Kaedah enkripsi atau dekripsi yang berdasarkan katalaluan ini dinamakan sebagai *'Password Based Encryption (PBE)'*. PBE adalah merupakan satu penyelesaian bagi masalah penyimpanan kunci – kunci kriptografi dan ia juga boleh digunakan untuk menjamin keselamatan data sensitif yang lain.

PBE akan menggunakan katalaluan atau sebaris ayat untuk menjanakan kunci enkripsi dan satu nilai rawak akan digunakan dan ianya dirujuk sebagai *salt*. *Salt* digunakan bagi meningkatkan kekuatan enkripsi PBE. *Salt* ini akan digabungkan dengan katalaluan dan ia akan digunakan semasa melakukan enkripsi. (Jaworski, 2000; Knudsen,1998). Rajah 6.2 menunjukkan bagaimana enkripsi fail dilakukan dengan menggunakan PBE.



Rajah 6.2: Enkripsi fail menggunakan PBE

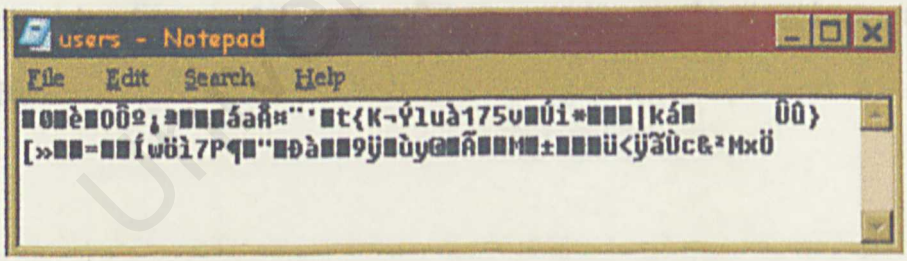
Bagi kelas PBE di dalam sistem SoftCrypt, mekanisme PBE diimplementasikan dengan menggunakan algoritma *PBEWithMD5AndDES* beserta sokongan kelas – kelas java yang berkaitan yang diambil dari pakej java.security.*, javax.crypto.* dan javax.crypto.spec.*.

Kelas PBE ini akan digunakan untuk mengenkripkan fail – fail kunci penguna SoftCrypt. Oleh kerana PBE memerlukan katalaluan untuk melakukan enkripsi, maka katalaluan pengguna yang digunakan semasa proses *login* akan dijadikan sebagai katalaluan untuk mengenkrip fail – fail kunci pengguna tersebut. Dengan itu,

kerahsiaan katalaluan pengguna adalah penting kerana ianya akan digunakan untuk mengenkrip fail – fail kunci pengguna tersebut.

Fail "users.skm" yang dihasilkan oleh kelas userPword pada asalnya adalah di dalam bentuk teks biasa seperti di dalam Rajah 6.3. Maka fail tersebut perlu dienkripkan dengan menggunakan PBE bagi melindungi kerahsiaan katalaluan pengguna. Bagi enkripsi ini, katalaluan akan dibekalkan oleh sistem SoftCrypt sendiri. Rajah 6.4 menunjukkan fail "users.skm" yang telah dienkripkan dengan PBE.

Rajah 6.3: Fail "users.skm" sebelum dienkrip



Rajah 6.4: Fail "users.skm" setelah dienkrip

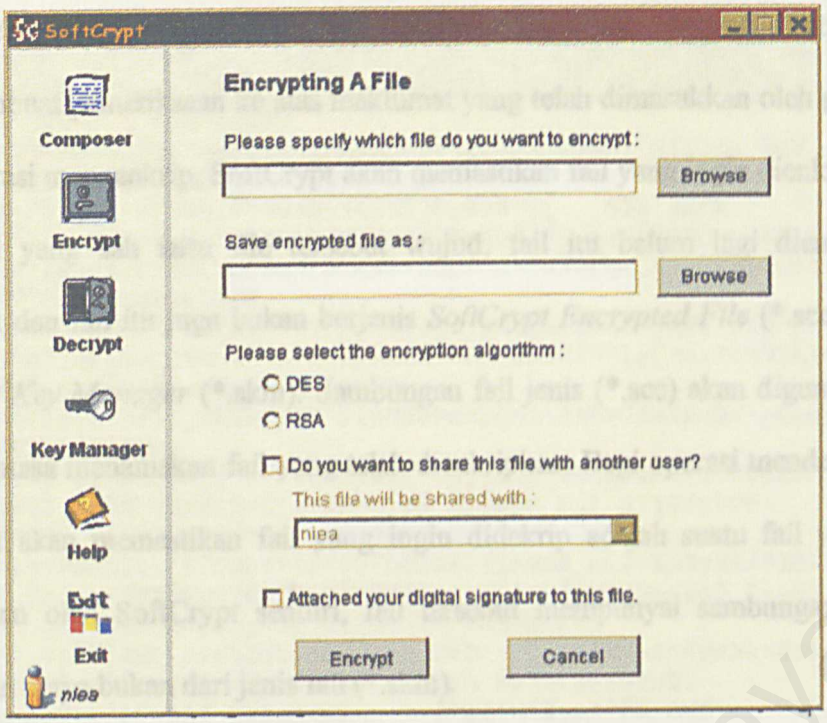
6.4.4 Encryption, Decryption, Sign dan Verify

Encryption, decryption, sign dan *verify* adalah merupakan fungsi – fungsi yang paling penting dan utama di dalam sistem SoftCrypt. Merujuk kepada Rajah 6.5

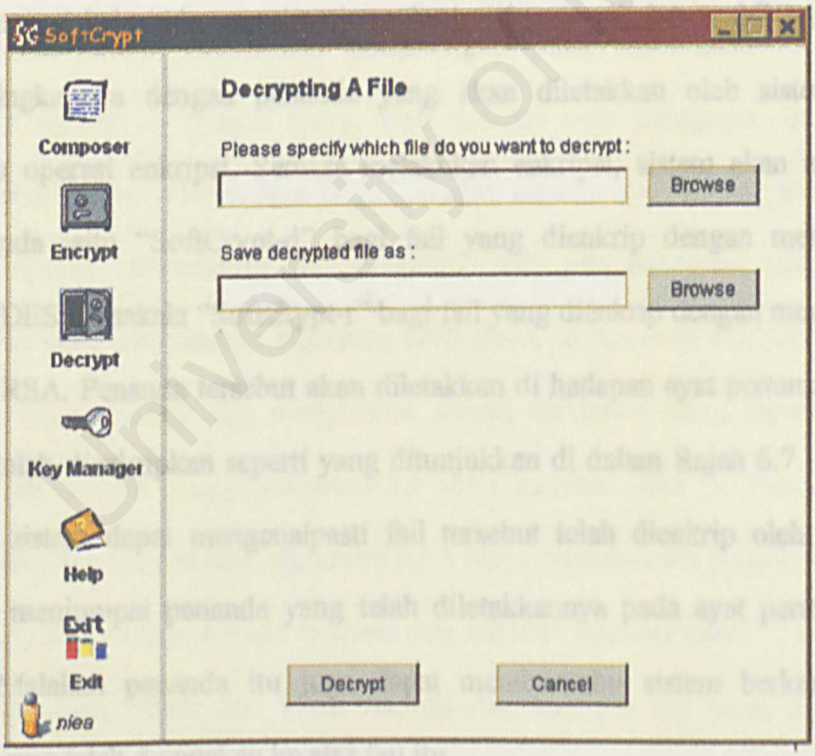
dan Rajah 6.6, kawalan atau komponen – komponen yang digunakan semasa proses pengumpulan maklumat sebelum melakukan enkripsi, dekripsi, menandatangani fail ataupun mengesahkan tandatangan akan dikumpulkan dan dipamirkan di dalam satu *panel.*

Operasi enkripsi akan diaktifkan apabila pengguna menekan butang *Encrypt* dan komponen yang berkaitan dengan enkripsi akan dipamirkan seperti di dalam Rajah 6.5 manakala operasi dekripsi akan diaktifkan setelah pengguna menekan butang *Decrypt* dan komponen yang berkaitan dengan dekripsi pula yang akan dipaparkan seperti di dalam Rajah 6.6.

Bagi operasi menandatangai fail, ianya akan diaktifkan apabila pengguna telah menekan butang *Encrypt* dan menanda betul pada kotak pilihan yang menyatakan *"Attached your digital signature to this file"*. Operasi mengesahkan tandatangan akan diaktifkan secara otomatik apabila operasi dekripsi telah diaktifkan iaitu semasa melakukan dekripsi ke atas fail, jika sistem menjumpai satu tandatangan telah disertakan bersama fail tersebut maka operasi mengesahkan tandatangan akan diaktifkan.



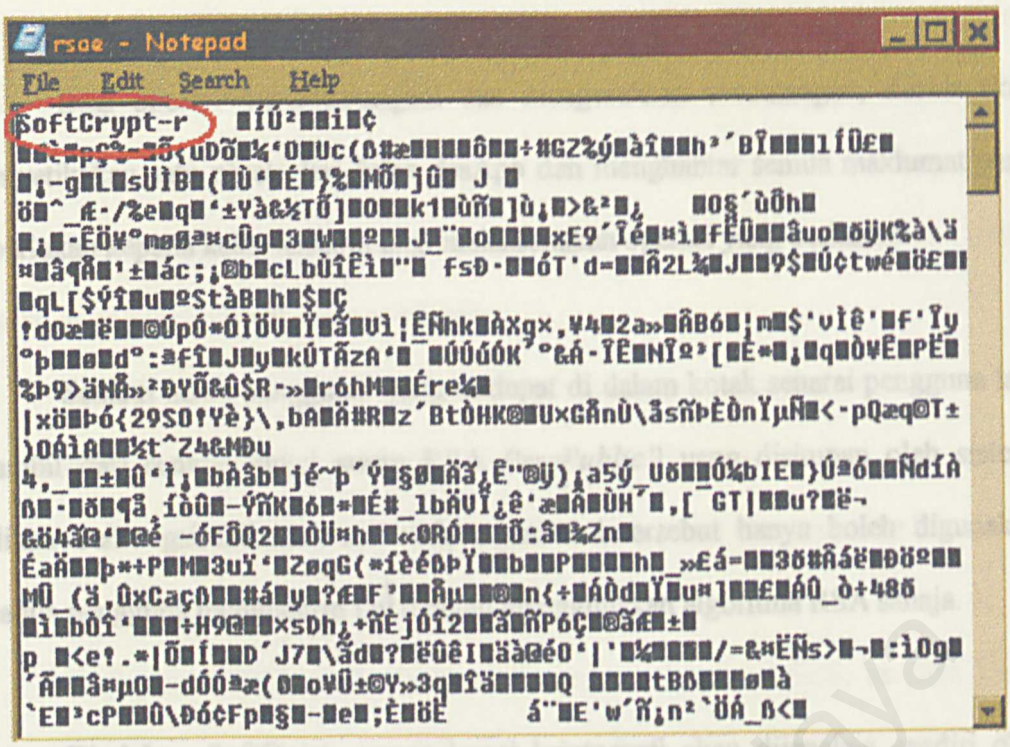
Rajah 6.5: SoftCrypt setelah butang *Encrypt* ditekan



Rajah 6.6: SoftCrypt setelah butang Decrypt ditekan

Sebelum sistem membenarkan perlaksanaan operasi – operasi di atas, sistem akan membuat pemeriksaan ke atas maklumat yang telah dimasukkan oleh pengguna. Bagi operasi mengenkrip, SoftCrypt akan memastikan fail yang ingin dienkrip adalah suatu fail yang sah iaitu file tersebut wujud, fail itu belum lagi dienkrip oleh SoftCrypt dan fail itu juga bukan berjenis *SoftCrypt Encrypted File* (*.sce) ataupun *SoftCrypt Key Manager* (*.skm). Sambungan fail jenis (*.sce) akan digunakan oleh sistem semasa menamakan fail yang telah dienkripkan. Bagi operasi mendekrip pula, SoftCrypt akan memastikan fail yang ingin didekrip adalah suatu fail yang telah dienkripkan oleh SoftCrypt sendiri, fail tersebut mempunyai sambungan berjenis (*.sce) dan ianya bukan dari jenis fail (*.skm).

SoftCrypt dapat mengenalpasti suatu fail itu sah untuk melakukan opersi enkripsi atau dekripsi dengan membaca baris pertama di dalam fail tersebut dan membandingkannya dengan penanda yang akan diletakkan oleh sistem semasa melakukan operasi enkripsi. Semasa melakukan enkripsi, sistem akan meletakkan satu penanda iaitu "SoftCrypt-d" bagi fail yang dienkrip dengan menggunakan algoritma DES manakala "SoftCrypt-r" bagi fail yang dienkrip dengan menggunakan algoritma RSA. Penanda tersebut akan diletakkan di hadapan ayat pertama di dalam fail yang telah dienkripkan seperti yang ditunjukkan di dalam Rajah 6.7. Oleh yang demikian, sistem dapat mengenalpasti fail tersebut telah dienkrip oleh SoftCrypt apabila ia menjumpai penanda yang telah diletakkannya pada ayat permulaan fail tersebut. Malahan penanda itu juga dapat memberitahu sistem berkenaan jenis algoritma yang telah digunakan ke atas fail itu.



Rajah 6.7: Fail yang telah dienkrip menggunakan algoritma RSA

Setelah selesai melakukan pemeriksaan yang diperlukan, sistem akan meneruskan operasi – operasi yang telah diminta oleh pengguna. Sistem akan merujuk kepada kelas – kelas java yang berkaitan dengan operasi yang telah diminta iaitu bagi operasi mengenkrip dan mendekrip fail, jika pengguna telah memilih butang radio yang menyatakan algoritma DES maka sistem akan menyetikakan satu objek dari kelas desApp dan menghantar semua maklumat yang diperlukan bagi melaksanakan operasi yang berkenaan ke kelas tersebut. Jika pengguna memilih butang radio yang menyatakan algoritma RSA dan menanda betul pada kotak pilihan yang menanyakan *"Do you want to share this file with another user?"* , sistem akan menyetikakan satu objek dari kelas rsaApp dan enkripsi fail yang dibuat akan menggunakan kunci awam milik nama pengguna yang telah dipilih di dalam kotak senarai pengguna lain. Jika kotak pilihan tersebut tidak ditandakan, maka enkripsi fail akan dilakukan dengan menggunakan kunci awam pengguna itu sendiri.

Bagi operasi menandatangani dan mengesahkan tandatangan, sistem akan menyetikakan satu objek dari kelas dsaApp dan menghantar semua maklumat yang diperlukan kepada kelas tersebut bagi melaksanakan operasi yang berkaitan.

Senarai nama pengguna yang terdapat di dalam kotak senarai pengguna lain diambil dari senarai kunci awam RSA *('rsaPublic')* yang disimpan oleh sistem. Pilihan berkongsi fail atau menandatangani fail tersebut hanya boleh digunakan apabila pengguna mengenkrip fail dengan menggunakan algoritma RSA sahaja.

Di dalam SoftCrypt, semua kunci kriptografi akan dijanakan sendiri oleh sistem dan setiap kelas algoritma kepunyaan SoftCrypt iaitu kelas desApp, kelas rsaApp dan kelas dsaApp akan bertanggungjawab untuk menjanakan dan menguruskan kunci mengikut algoritma masing – masing. Selain daripada itu, setiap kelas di atas akan mnguruskan segala urusan yang berkaitan dengan algoritma masing – masing.

Kelas desApp akan bertanggungjawab ke atas pengurusan algoritma DES di dalam SoftCrypt, kelas rsaApp pula akan menguruskan segala hal yang berkaitan dengan algoritma RSA dan akhir sekali, penggunaan algoritma DSA di dalam SoftCrypt akan diuruskan oleh kelas dsaApp. Antara perkara – perkara yang diuruskan atau disokong oleh kelas – kelas tersebut ialah perkara yang berkaitan dengan penjanaan kunci, penyimpanan kunci, melakukan enkripsi dan dekripsi bagi algoritma DES dan RSA dan menandatangani serta mengesahkan tandatangan bagi algoritma DSA. Bagi kelas rsaApp dan dsaApp, kelas itu juga menyediakan

sokongan untuk menngimport dan mengeksport kunci awam mengikut algoritma masing – masing.

Secara umumnya terdapat tiga langkah yang perlu dilakukan untuk menjanakan kunci – kunci kriptografi iaitu:

1) Dapatkan objek penjana kunci bagi algoritma yang ingin digunakan.
2) Berikan nilai awal *(initializing)* kepada penjana kunci tersebut
3) Arahkan penjana kunci untuk menjanakan satu kunci atau sepasang kunci.

Bagi sistem kriptografi kunci rahsia seperti algoritma DES, penjana kunci yang akan digunakan ialah KeyGenerator dari kelas javax.crypto.KeyGenerator. Berikut adalah kod aturcara yang akan menjanakan kunci bagi algoritma DES dan kunci itu bersaiz 56 bit.

```
// Generate a secret key for DES
KeyGenerator keygen = KeyGenerator.getInstance("DES");
Keygen.init(new SecureRandom());
SecretKey key = keygen.generateKey();
```

Bagi sistem kriptografi kunci awam pula yang melibatkan algoritma seperti RSA dan DSA, penjana kunci yang akan digunakan ialah KeyPairGenerator dari kelas java.security.KeyPairGenerator. Walaupun pakej JCE membenarkan penjanaan kunci bagi algoritma RSA tetapi ia tidak mnyediakan kelas - kelas bagi melakukan operasi enkripsi atau dekripsi yang menggunakan algoritma RSA. Oleh itu pakej perpustakaan Bouncy Castle Crypto akan digunakan bagi menyokong operasi

enkripsi dan dekripsi yang menggunakan algoritma RSA. Pembekal kriptografi bagi pakej perpustakaan ini dikenali sebagai BC. Berikut adalah kod aturcara yang akan menjanakan sepasang kunci awam bagi algoritma RSA dan DSA dan setiap kunci itu akan bersaiz 1024 bit.

```
// Generate a key pair for RSA
KeyPairGenerator keygen = KeyPairGenerator.getInstance("RSA", "BC");
keygen.initialize(1024,new java.security.SecureRandom());
KeyPair pair = keygen.genKeyPair();

// Generate a key pair for DSA
KeyPairGenerator kg = KeyPairGenerator.getInstance("DSA");
Kg.initialize(1024);
KeyPair pair = kg.genKeyPair();
```

Sebelum melakukan operasi enkripsi atau dekripsi ke atas data, satu seketikaan (*instances*) C*ipher* dari kelas javax.crypto.Cipher perlu disediakan. *Cipher* ini akan menukarkan satu blok data kepada satu blok data yang lain semasa melakukan enkripsi ataupun dekripsi. Penggunaan *Cipher* melibatkan tiga langkah umum iaitu:

1) Dapatkan seketikaan *Cipher* dengan menggunakan kaedah getInstance().
2) Berikan nilai awal kepada *Cipher* melalui penggunaan metod init() bagi menetapkan samada *Cipher* tersebut akan melakukan enkripsi atau dekripsi. Metod init() akan menerima dua nilai iaitu *mode* bagi *Cipher* yang terdiri

daripada Cipher.ENCRYPT_MODE atau Cipher.DECRYPT_MODE dan kunci yang akan digunakan semasa melakukan operasi enkripsi atau dekripsi.

3) Lakukan enkripsi atau dekripsi ke atas data dengan menggunakan metod update() atau metod doFinal().

Di dalam SoftCrypt, seketikaan *cipher* bagi algoritma DES akan menggunakan *mode* ECB dan PKCS5Padding. Mode ECB (*'Electronic Code Book'*) akan menentukan bagaimana satu blok teks biasa dienkripkan ke satu blok teks cipher.

Berikut adalah langkah – langkah umum yang digunakan untuk menghasilkan tandatangan dengan menggunakan algoritma DSA:

1) Dapatkan objek Signature dengan menggunakan metod getInstance().
2) Berikan nilai awalan iaitu kunci rahsia penandatangan kepada objek Signature dengan menggunakan metod initSign().
3) Gunakan metod update() untuk menambah data yang dibaca ke dalam tandatangan.
4) Kira tandatangan tersebut dengan menggunakan metod sign(). Metod ini akan memulangkan satu tatasusunan byte.

Di bawah ini pula adalah langkah – langkah umum untuk mengesahkan tandatangan yang telah dihasilkan dengan menggunakan algoritma DSA:

1) Dapatkan objek Signature dengan menggunakan metod getInstance().
2) Berikan nilai awalan iaitu kunci awam penandatangan kepada objek Signature dengan menggunakan metod initVerify().

3) Gunakan metod update() untuk menambah data yang dibaca ke dalam tandatangan.

4) Bandingkan tandatangan tersebut samada sepadan atau tidak dengan menggunakan metod verify().

6.4.5 Key Manager

Oleh kerana urusan penjanaan dan penyimpan kunci telah diserahkan kepada kelas - kelas mengikut algoritma masing – masing maka peranan yang perlu dimainkan oleh komponen *Key Manager* telah berkurangan. Dengan itu, *Key Manager* hanya perlu menguruskan segala urusan yang berkaitan dengan penyimpanan senarai kunci awam bagi algoritma RSA dan DSA serta hal – hal berkenaan mengimport dan mengeksport kunci awam bagi pengguna SoftCrypt.

Senarai kunci awam tersebut akan dihasilkan dengan menggunakan kaedah yang sama seperti penghasilan senarai nama pengguna SoftCrypt serta katalaluannya iaitu dengan menggunakan objek dari kelas *Properties*. Kelas rsaPublic akan menghasilkan satu senarai kunci awam bagi algoritma RSA manakala kelas dsaPublic pula akan menghasilkan satu senarai kunci awam bagi algoritma DSA.

6.4.6 User Details

Komponen User Details akan memaparkan maklumat berkenaan nama pengguna, katalaluan pengguna dan kunci umum bagi algoritma RSA dan DSA.

Fungsi komponen ini ialah untuk membenarkan pengguna menukar katalaluan mereka dan menjanakan semula set kunci kepunyaan mereka.

6.4.7 Composer

Composer adalah merupakan satu – satunya komponen dari jenis program yang boleh dilarikan (*.exe). Ia adalah suatu program pengedit teks yang hampir sama seperti Notepad tetapi jenis fail yang boleh disokong oleh *composer* ini hanyalah dari jenis teks sahaja iaitu yang mempunyai sambungan fail (*.txt). *composer* ini telah dibina dengan menggunakan *wizard* yang terdapat pada Microsoft Visual J++.

PENGUJIAN

7.1 Pengenalan

Pengujian adalah merupakan elemen yang penting dan kritikal di dalam memastikan kualiti sesuatu sistem yang telah dibangunkan. Pengujian dilaksanakan bagi memastikan sistem yang telah dibina itu dapat dilarikan dan melakukan fungsi – fungsinya seperti yang telah ditetapkan. Ianya juga bagi mengesahkan bahawa sistem tersebut telah memenuhi keperluan – keperluan dan jangkaan penggunanya. Selain daripada itu, pengujian adalah perlu bagi tujuan pentahkikan perisian di mana ianya untuk memastikan sistem yang telah dibangunkan adalah betul.

Kebiasaannya, pengujian yang dilakukan melibatkan pengujian statik seperti pemeriksaan kod, pengujian kotak hitam yang berdasarkan spesifikasi secara luaran, pengujian integrasi, pengujian regrasi, pengujian sistem dan sebagainya. Sesuatu pengujian itu perlulah dimulakan dari awal dan dilakukan secara berterusan di sepanjang kitar hayat pembangunan sistem agar ralat – ralat dapat dikenalpasti di peringkat awal dan dibetulkan dengan segera.

7.2 Teknik Pengujian

Pengujian - pengujian yang telah dilakukan ke atas SoftCrypt merangkumi beberapa peringkat dan ianya dimulakan dengan pengujian unit, pengujian integrasi dan diakhiri dengan pengujian sistem. Kebanyakan pengujian – pengujian tersebut telah dilakukan dengan memasukkan data – data ujian ke dalam sistem dan kemudian hasil daripada pengujian tersebut akan dianalisa bagi mengenalpasti sebarang ralat yang telah berlaku atau wujudnya hasil – hasil di luar jangkaan.

Secara keseluruhannya, teknik pengujian yang menggunakan pendekatan secara menokok telah digunakan semasa melakukan pengujian - pengujian di atas. Dua aspek penting di dalam pendekatan pengujian secara menokok ialah pengujian unit dan pengujian kefungsian.

Di dalam pengujian unit, setiap komponen akan dipastikan bebas dari ralat dan dapat dilarikan dengan sempurna dan pengujian kefungsian pula akan menyemak samada sistem masih berfungsi dengan betul atau tidak selepas penokokan unit kecil kod ataupun selepas penambahan komponen – komponen ke dalam sistem dilakukan.

Perbezaan antara teknik pengujian biasa dengan teknik pengujian secara menokok ialah selepas setiap penambahan unit kecil komponen ke dalam sistem dilakukan, pengujian kefungsian bagi keseluruhan sistem akan dijalankan. Dalam erti kata lain, pengujian kefungsian bagi keseluruhan sistem akan dilakukan berulang kali. Walaupun pada zahirnya teknik ini kelihatan seperti memakan masa yang lama kerana pengujian kefungsian perlu dilakukan bagi setiap penambahan komponen

baru ke dalam sistem, tetapi sebenarnya teknik ini dapat memendekan tempoh masa dan memudahkan aktiviti pengawalansilap *('debugging')* yang dilakukan selepas selesai proses penulisan kod aturcara sistem.

Melalui pendekatan secara menokok, lokasi di mana ralat berlaku dapat dikenalpasti dengan mudah dan tepat dan kerja – kerja pengubahsuaian kod mudah untuk dilakukan. Ini kerana, selepas setiap penambahan komponen baru ke dalam sistem, pengujian yang melibatkan aktiviti pengesanan, pembasmian serta penjejakan ralat akan dilakukan berulang kali sehingga tiada ralat dijumpai bagi mempastikan sistem tersebut berfungsi dengan betul mengikut komponen – komponen yang ada padanya ketika itu dan bebas dari sebarang ralat sebelum penambahan komponen yang lain pula dilakukan.

Selain dari itu juga, pengujian secara menokok dapat memastikan setiap komponen yang baru ditambah ke dalam sistem tidak menghasilkan ralat – ralat yang lain pula dan jika terdapatnya ralat, ianya dapat disingkirkan di peringkat awal lagi..

7.3 Pengujian Unit

Pengujian unit memfokus kepada pengujian ke atas unit – unit atau bahagian yang terkecil iaitu yang terdiri daripada subrutin dan subfungsi – subfungsi utama yang terdapat di dalam komponen – komponen SoftCrypt.

Seperti yang telah dinyatakan sebelum ini, SoftCrypt mempunyai lima modul utama dan lima modul tersebut telah diterjemahkan ke dalam lapan komponen utama yang terdapat di dalam antaramuka SoftCrypt. Oleh yang demikian, pengujian telah dilakukan ke atas setiap komponen tersebut. Setiap komponen itu akan diuji secara berasingan bagi mempastikan ia dapat melaksanakan fungsi – fungsi spesifik yang perlu dilakukannya. Pengujian ke atas setiap komponen itu juga adalah untuk menguji keberkesanan aturcara serta untuk mempastikan setiap komponen bebas dari sebarang ralat sebelum ianya digabungkan menjadi satu sistem.

Antara langkah – langkah yang telah dilakukan semasa menjalankan pengujian unit ialah:

- Membaca dan memeriksa kod – kod aturcara dengan teliti bagi mengelakkan adanya kesalahan - kesalahan seperti kesalahan menaip, kesalahan sintaks dan kesalahan dari segi logik pengaturcaraan.
- Menyemak komponen antaramuka atau kelas – kelas yang telah digunakan di dalam aturcara berkenaan.
- Menguji antaramuka bagi setiap komponen – komponen yang berkaitan untuk memastikan aliran maklumat yang betul dan lancar.

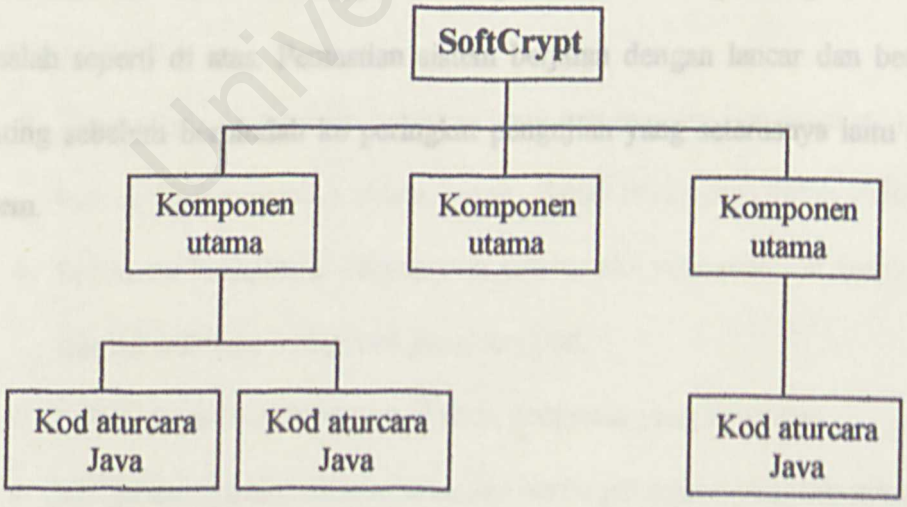
- Memastikan bahagian yang tidak bersandar yang terdapat di dalam struktur kawalan diuji sekurang – kurangnya sekali.

Di akhir proses pengujian unit ini, kod – kod aturcara Java yang penting seperti PBE.java, userPword.java, desApp.java, rsaApp.java, dsaApp.java dan beberapa lagi kod aturcara lain telah dipastikan ianya dapat dilarikan secara berasingan dengan tepat dan betul tanpa menimbulkan sebarang masalah atau ralat.

7.4 Pengujian Integrasi

Di dalam pengujian integrasi pula, pengujian dilakukan untuk mengenalpasti ralat – ralat yang terhasil apabila komponen – komponen antaramuka SoftCrypt mula digabungkan secara berperingkat. Pengujian dilakukan dengan menguji hubungan di antara komponen – komponen tersebut.

Secara umumnya, pengujian integrasi bagi SoftCrypt dilakukan dengan menggunakan teknik bawah-atas di mana komponen yang berada di bawah sekali seperti kod aturcara – kod aturcara Java yang mempunyai fungsi yang hampir sama akan dikumpulkan di dalam satu kumpulan dan ianya akan diintegrasikan dengan komponen utama antaramuka SoftCrypt yang lebih atas daripadanya. Komponen – komponen utama antaramuka SoftCrypt pula akan diintegrasikan untuk membentuk satu antaramuka tunggal bagi SoftCrypt. Rajah 7.1 menunjukkan bagaimana komponen – komponen di dalam SoftCrypt digabungkan secara umumnya.



Rajah 7.1: Gambaran umum perhubungan antara komponen di dalam SoftCrypt

Oleh kerana keseluruhan kerja – kerja mengintegrasikan komponen – komponen SoftCrypt dilakukan di dalam pengedit java bergrafik iaitu Symantec Visual Café maka ralat – ralat akan diperiksa secara otomatik oleh Visual Café semasa melakukan kompilasi ke atas sistem SoftCrypt.

Walaubagaimanapun, pemeriksaan yang dilakukan oleh pengkompil Visual Café adalah tidak mencukupi kerana masih wujud juga kesalahan – kesalahan yang lain yang dijumpai semasa menjalankan pengujian integrasi ini.Antaranya ialah:

- Kehilangan data apabila berpindah dari satu antaramuka ke antaramuka yang lain.
- Sesuatu komponen memberikan kesan yang bercanggah ke atas sesuatu komponen yang lain.
- Apabila digabungkan dengan komponen – komponen yang lain, output yang dihasilkan adalah berbeza dan di luar jangkaan.

Di akhir proses pengujian integrasi, segala ralat yang telah dijumpai telah dibetulkan dan sistem SoftCrypt telah dipastikan tidak lagi mengalami masalah – masalah seperti di atas. Pemastian sistem berjalan dengan lancar dan betul adalah penting sebelum berpindah ke peringkat pengujian yang seterusnya iaitu pengujian sistem.

7.5 Pengujian Sistem

Pengujian sistem yang dijalankan ke atas SoftCrypt adalah bertujuan untuk mencari kelemahan dan mengukur keupayaan SoftCrypt untuk berfungsi sebagai satu unit tunggal. Selain daripada itu, pengujian sistem juga berfungsi sebagai pengesahan yang membuktikan bahawa sistem telah memenuhi semua keperluan pengguna dan beroperasi seperti yang dikehendaki.

Berikut adalah pengujian - pengujian yang telah dilakukan ke atas sistem SoftCrypt dari segi:

7.5.1 Antaramuka

Pengujian dari aspek antaramuka melibatkan isu – isu susunatur, penampilan dan kebolehfungsian komponen – komponen antaramuka seperti butang, kotak dialog, penggunaan panel dan sebagainya. Antara perkara – perkara yang perlu dipastikan ialah:

- Penggunaan warna latar belakang yang seragam dan konsisten mengikut jenis bagi setiap antaramuka utama, kotak – kotak mesej serta kotak dialog.
- Susunatur komponen – komponen antaramuka yang tersusun dan penggunaan saiz serta format komponen yang seragam.
- Lokasi paparan antaramuka di skrin pengguna yang konsisten.
- Komponen – komponen antaramuka berfungsi seperti yang sepatutnya.
- Menyemak ejaan di dalam label – label paparan.

- Butang – butang berfungsi seperti yang dikehendaki iaitu melakukan tugas – tugas yang diingini oleh pengguna atau memaparkan skrin paparan antaramuka yang sepatutnya.
- Penggunaan mesej – mesej ralat yang seragam dan bersesuaian.
- Pengawalan kemasukan input – input pengguna melalui komponen – komponen antaramuka seperti membenarkan pengguna manaip hanya sebanyak lapan aksara sahaja bagi katalaluan semasa proses *login* atau semasa pendaftaran pengguna baru, membenarkan pengguna memilih hanya satu algoritma sahaja semasa hendak melakukan enkripsi dan lain – lain lagi kaedah pengawalan.
- Data – data yang dimasukkan oleh pengguna melalui antaramuka dapat diterima dan disimpan dengan tepat oleh sistem.

7.5.2 Kefungsian

Pengujian kefungsian yang dilakukan adalah untuk mempastikan sistem telah memenuhi semua keperluan fungsian yang telah dinyatakan sebelum ini serta ianya dapat melaksanakan fungsi – fungsi seperti:

- Mengenkrip dan mendekrip fail dengan menggunakan algoritma DES atau RSA.
- Menandatangani dan mengesahkan tandatangan digital yang menggunakan algoritma DSA.
- Pengurusan kunci – kunci kriptografi.
- Pengurusan akaun – akaun pengguna.
- Enkripsi dan dekripsi menggunakan algoritma PBE.

- Pengesahan data – data yang telah dimasukkan oleh pengguna.
- Pengurusan senarai kunci awam bagi algoritma RSA dan DSA.
- Pemastian keselamatan sistem secara keseluruhan.
- Kebolehfungsian *composer* untuk berfungsi sebagai pengedit teks.
- Memaparkan maklumat – maklumat pengguna.
- Pengurusan penyelenggaraan katalaluan pengguna dan kunci – kunci kriptografi.

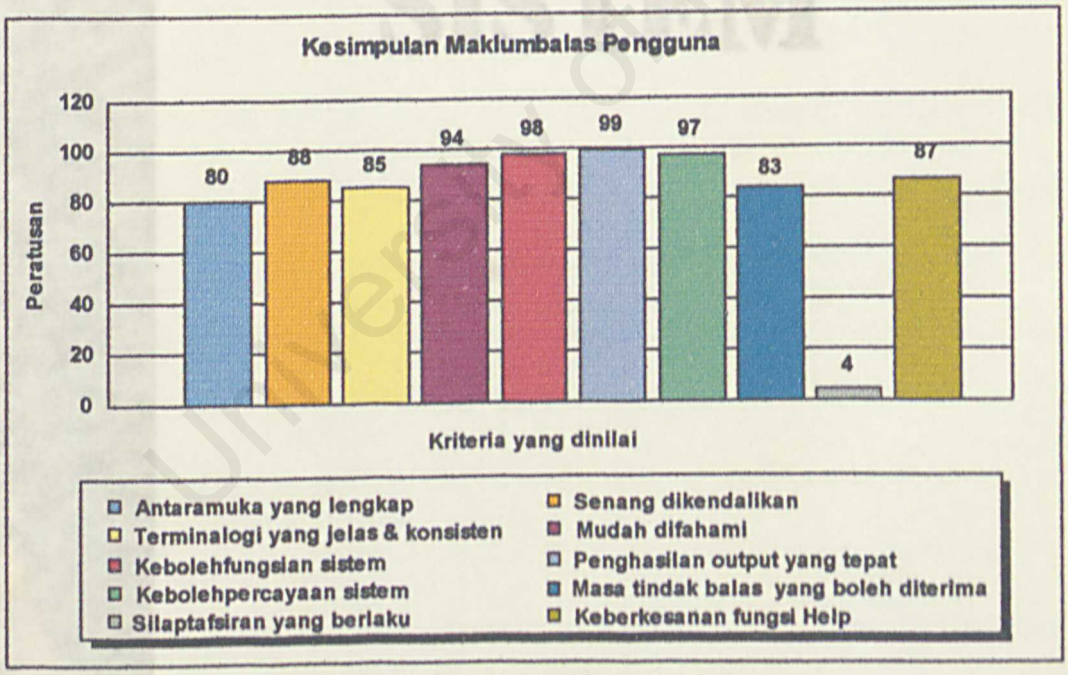
Bagi menguji sejauh mana kemampuan SoftCrypt dapat melakukan enkripsi dan dekripsi ke atas kepelbagaian jenis fail, beberapa jenis data yakni fail – fail yang berlainan jenis telah digunakan sebagai data ujian. Antara jenis – jenis fail yang telah diuji ialah fail teks biasa (*.txt), fail *'Word Document'* (*.doc), fail Adobe Acrobat (*.pdf) dan fail – fail imej seperti (*.gif) dan (*.jpeg). Selain daripada itu, kepelbagaian saiz fail juga diambil kira tetapi ianya hanya melibatkan jenis fail teks biasa sahaja. Antara aspek – aspek yang akan dipastikan dan diperiksa bagi setiap hasil enkripsi yang telah dilakukan ke atas fail – fail di atas ialah:

- Kandungan fail yang telah didekrip mestilah serupa dan sama seperti kandungan fail asalnya tanpa sebarang kehilangan huruf atau penambahan huruf pada fail tersebut.
- Saiz fail yang telah didekrip mestilah sama dan menepati saiz fail asalnya. Ini bagi mempastikan tiada berlakunya kehilangan data atau penambahan data setelah fail itu dienkrip dan kemudian didekripkan semula.
- Bagi fail – fail imej, perbandingan dibuat secara umum sahaja di mana imej yang telah dedekrip haruslah sama seperti imej asalnya.

7.5.3 Penerimaan

Ujian penerimaan dilakukan bagi memastikan SoftCrypt telah memenuhi keperluan pengguna dari sudut pandangan pengguna itu sendiri. Penilaian yang akan diberikan oleh pengguna adalah berdasarkan kefahaman mereka terhadap keupayaan sistem untuk berfungsi mengikut keperluan dan kehendak mereka.

Oleh yang demikian, beberapa orang pengguna telah diminta untuk mencuba sistem ini. Bagi mengumpulkan dan menganalisa maklumbalas penguna terhadap keberkesanan sistem ini, satu borang soal selidik juga telah diedarkan untuk diisi oleh mereka Setelah dianalisa, hasil daripada maklumbalas pengguna dapat dirumuskan seperti yang ditunjukkan di dalam Rajah 7.2.



Rajah 7.2: Kesimpulan Maklumbalas Pengguna

PENILAIAN SISTEM

8.1 Pengenalan

Setelah sesuatu sistem siap dibangunkan, ianya perlulah dinilai bagi menentukan kelebihan yang ditawarkan oleh sistem tersebut berbanding dengan sistem lain serta membuat penilaian terhadap kekurangan yang terdapat pada sistem itu. Selain daripada itu, masalah – masalah yang dihadapi ketika membangunkan sistem juga akan dibincangkan dan diberikan cadangan penyelesaian yang bersesuaian.

8.2 Hasil Akhir SoftCrypt

Walaupun terdapat beberapa perubahan yang terpaksa dilakukan semasa di dalam proses pembangunan SoftCrypt, sistem ini akhirnya telah dapat memenuhi objektif asal dan semua keperluan yang telah dinyatakan sebelum ini berjaya dipenuhi. Selain daripada itu, SoftCrypt juga telah ditambah dengan beberapa ciri tambahan bagi memudahkan pengguna menggunakan SoftCrypt.

8.2.1 Objektif dan Keperluan Asal SoftCrypt

Berikut adalah objektif serta keperluan asal sistem yang telah dipenuhi setelah selesai proses pembangunan SoftCrypt:

- SoftCrypt boleh dijadikan sebagai salah satu alternatif kepada program – program enkripsi yang sedia ada di pasaran dan menjadi panduan kepada pembangun perisian yang lain yang ingin membangunkan perisian enkripsi yang baru, ia boleh digunakan untuk tujuan pembelajaran bagi topik berkaitan dengan kriptografi dan yang terutama sekali ia boleh digunakan untuk tujuan enkripsi/dekripsi bagi kegunaan peribadi pengguna.
- SoftCrypt yang terhasil adalah merupakan aplikasi keselamatan tunggal (*'stand-alone application'*) yang menyediakan perkhidmatan seperti enkripsi, dekripsi serta tandatangan digital dengan menggunakan algoritma kriptografi berjenis kuat iaitu DES, RSA dan DSA.
- SoftCrypt juga menyediakan antaramuka yang mesra pengguna, sistem yang mudah digunakan serta senang dipelajari bagi semua tahap pengguna.

- SoftCrypt dapat menjamin dan melindungi keselamatan data penggunanya.
- SoftCrypt juga sesuai digunakan untuk kegunaan peribadi mahupun untuk urusan rasmi.

8.2.2 Perubahan pada SoftCrypt

Di bawah ini pula, adalah penambahan atau perubahan yang telah dilakukan serta perkara - perkara lain yang berkaitan dengan SoftCrypt. Sesetengah perubahan yang dilakukan ke atas SoftCrypt adalah untuk memantapkan dan meningkatkan keefisyenan SoftCrypt manakala ada sesetengah perubahan yang terpaksa dibuat untuk memastikan SoftCrypt dapat berjalan dengan lancar dan memenuhi objektif pembangunannya.

- SoftCrypt boleh digunakan oleh berbilang pengguna *('multiple users')*.
- SoftCrypt menyediakan pengedit teksnya sendiri bagi memudahkan pengguna melakukan urusan mengedit fail – fail teks mereka.
- Algoritma RSA yang terdapat di dalam SoftCrypt sebenarnya hanya digunakan untuk mengenkrip kunci rahsia DES di mana bagi perlaksanaan enkripsi dan dekripsi menggunakan algoritma RSA, fail data tersebut akan dienkripkan dengan menggunakan algoritma DES dahulu. Kunci rahsia DES tersebut akan dijanakan oleh sistem secara rawak dan ianya berbeza daripada kunci rahsia DES yang dipunyai oleh pengguna tersebut. Kunci rahsia DES yang baru dijanakan oleh sistem itu yang akan dienkrip dengan menggunakan algoritma RSA. Ini kerana perlaksanaan algoritma RSA adalah sangat perlahan maka ianya kurang sesuai untuk melakukan enkripsi ke atas data yang bersaiz besar. Ianya lebih sesuai digunakan untuk mengenkrip

katalaluan atau kunci - kunci rahsia yang bersaiz kecil seperti kunci rahsia DES. (Jaworski, 2000; Knudsen, 1998; Schneier, 1996). Selain daripada itu, Java juga telah meletakkan had ke atas saiz data yang boleh dienkrip menggunakan algoritma RSA iaitu 117 bit (lebih kurang 117 patah perkataan) bagi sekali enkripsi.

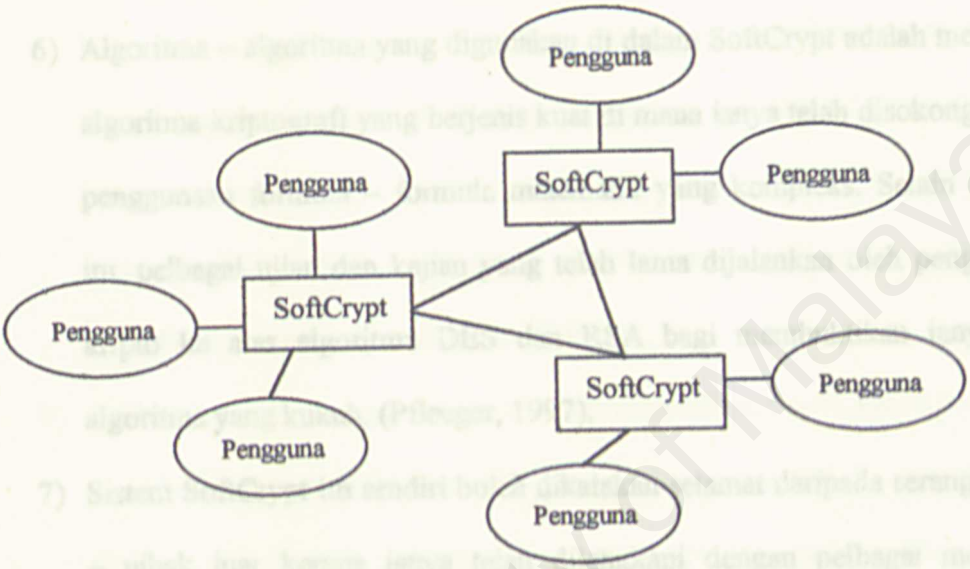
- Walaupun enkripsi menggunakan algoritma DES boleh dilakukan ke atas semua jenis fail tetapi enkripsi yang menggunakan algoritma RSA tidak dapat berbuat demikian. Algoritma RSA di dalam SoftCrypt hanya boleh menyokong enkripsi ke atas fail berjenis teks sahaja. Ini kerana hasil enkripsi dan dekripsi yang menggunakan algoritma RSA perlu dienkodkan dan dinyahkodkan dengan sejenis sistem pengkodan piawai bagi membolehkannya menyokong pelbagai jenis fail. Kaedah pengkodan tersebut agak rumit dan pembangun sistem kekurangan masa untuk mempelajari dan mengimplementasikan kaedah pengkodan tersebut ke dalam sistem SoftCrypt. Oleh yang demikian, jenis fail yang disokong oleh SoftCrypt tetap tidak berubah seperti yang telah dinyatakan di awal proses pembangunannya iaitu ia menyokong fail berjenis teks sahaja (*.txt).

8.3 Kelebihan SoftCrypt

SoftCrypt yang telah siap dibangunkan menawarkan pelbagai kelebihan berbanding dengan sistem enkripsi yang lain. Berikut adalah kelebihan yang boleh juga dikatakan sebagai tunggak kekuatan bagi SoftCrypt:

1) Setakat ini, didapati bahawa SoftCrypt sahaja yang menawarkan penggunanya dengan perkhidmatan enkripsi yang menggunakan gabungan algoritma seperti RSA, DES dan DSA. Walaupun terdapat perisian lain di pasaran yang menawarkan penggunaan algoritma di atas tetapi ianya hanya menawarkan penggunaan salah satu atau sebahagian sahaja daripada algoritma – algoritma tersebut. Tambahan pula, perisian yang lain perlu dibeli dahulu sebelum pengguna dapat menggunakannya manakala SoftCrypt boleh diedarkan kepada sesiapa yang ingin menggunakannya secara percuma.
2) SoftCrypt merupakan suatu sistem yang boleh digunakan oleh berbilang pengguna. Ini membolehkan seorang pengguna mempunyai beberapa akaun SoftCrypt yang berlainan untuk tujuan tugas – tugas yang berbeza ataupun satu salinan SoftCrypt boleh digunakan oleh beberapa orang pengguna tanpa melibatkan perkongsian kunci kriptografi yang sama kerana setiap akaun pengguna akan disediakan satu set kunci kriptografi yang baru.
3) Semasa pengguna baru mendaftarkan diri mereka ke dalam sistem SoftCrypt, secara otomatik sistem akan membuat satu salinan kunci – kunci awam pengguna baru itu dan memasukkannya ke dalam senarai kunci awam yang disimpan oleh SoftCrypt. Ini membolehkan setiap pengguna di dalam salinan sistem SoftCrypt yang sama berhubung antara satu sama lain melalui penggunaan kunci awam masing – masing. Selain daripada itu, SoftCrypt

juga telah menyediakan fungsi yang membolehkan penggunanya menambah kunci – kunci awam pengguna SoftCrypt di komputer yang lain ke dalam senarai kunci awam pengguna tersebut. Dengan itu, perkongsian data bukan sahaja melibatkan pengguna – pengguna di dalam satu salinan SoftCrypt yang sama malahan ia juga boleh dikongsi dengan pengguna di salinan SoftCrypt yang lain. Seperti yang ditunjukkan di dalam Rajah 8.1.



Rajah 8.1: Perkongsian kunci – kunci awam di kalangan pengguna SoftCrypt

4) Bagi setiap pengguna, mereka akan disediakan dengan satu set kunci kriptografi yang mengandungi satu kunci rahsia DES yang bersaiz 56 bit, sepasang kunci RSA yang terdiri daripada satu kunci awam dan satu kunci peribadi yang masing – masingnya bersaiz 1024 bit serta sepasang kunci DSA iaitu satu kunci awam dan satu kunci peribadi yang juga bersaiz 1024 bit. Saiz bagi kunci – kunci kriptografi yang digunakan di dalam SoftCrypt adalah merupakan saiz kunci yang biasa digunakan di dalam perisian – perisian enkripsi yang lain. Penggunaan saiz kunci yang besar akan

menggandakan bilangan kemungkinan kunci yang digunakan dan ini dapat menghalang daripada berlakunya serangan analisis-kripto ke atas data yang telah dienkrip menggunakan SoftCrypt.

5) Pengguna juga dibenarkan untuk menukar katalaluan mereka serta menjanakan semula kunci – kunci kepunyaan mereka apabila mereka merasakan kerahsiaan kunci atau katalaluan mereka telah terdedah atau terjejas.

6) Algoritma – algoritma yang digunakan di dalam SoftCrypt adalah merupakan algoritma kriptografi yang berjenis kuat di mana ianya telah disokong dengan penggunaan formula – formula matematik yang kompleks. Selain daripada itu, pelbagai ujian dan kajian yang telah lama dijalankan oleh penganalisis-kripto ke atas algoritma DES dan RSA bagi membuktikan ianya suatu algoritma yang kukuh. (Pfleeger, 1997).

7) Sistem SoftCrypt itu sendiri boleh dikatakan selamat daripada serangan pihak – pihak luar kerana ianya telah dilengkapi dengan pelbagai mekanisme keselamatan. Antaranya ialah pengguna perlu melalui proses autentikasi sebelum dapat menggunakan sistem ini, fail – fail yang mengandungi kunci kriptografi pengguna akan dienkrip oleh SoftCrypt dengan menggunakan katalaluan pengguna itu sendiri yang membuatkan penggunaan katalaluan yang tepat sahaja yang boleh mendekripkan semula fail – fail kunci tersebut, senarai katalaluan pengguna yang disimpan oleh SoftCrypt akan dienkrip menggunakan katalaluan sistem itu sendiri, fail - fail kunci pengguna dan senarai katalaluan pengguna tersebut dipastikan sentiasa berada di dalam keadaan yang telah dienkrip, pengguna hanya dibenarkan membuat empat kali cubaan sahaja semasa di proses *login* bagi mengelakkan berlakunya

serangan *'brute force'* ke atas katalaluan pengguna dan ciri – ciri keselamatan yang lain. Ciri – ciri keselamatan di atas adalah penting bagi menjamin integriti dan kebolehpercayaan sistem SoftCrypt.

8) SoftCrypt turut menyediakan antaramuka yang menarik serta mesra pengguna. Sistem ini dikatakan sebagai mesra pengguna kerana ianya telah dilengkapi dengan antaramuka bergrafik dan kebanyakan fungsinya boleh dilakukan melalui tetingkap *('window')*. Di samping itu, disediakan juga butang – butang yang membenarkan penggunanya melaksanakan fungsi – fungsi utama di dalam SoftCrypt. Kelebihan ini memudahkan pengguna kerana mereka hanya perlu menekan butang – butang tertentu bagi melaksanakan sesuatu tugas yang diingini tanpa perlu mengetahui secara spesifik bagaimana tugas tersebut dilakukan.

9) Pemaparan mesej – mesej ralat, mesej pemberitahuan serta mesej pengesahan yang boleh dijumpai di dalam SoftCrypt. Pemaparan mesej – mesej tersebut adalah penting untuk pengguna mengetahui apa yang sedang dilakukan oleh sistem dan kesan atau hasil akhir jika sesuatu pilihan itu digunakan serta ianya juga untuk memastikan sesuatu operasi yang penting seperti pemadaman telah mendapat pengesahan daripada pengguna sebelum operasi itu diteruskan.

10) Oleh kerana SoftCrypt merupakan sistem yang mesra pengguna, mudah untuk dipelajari serta senang dikendalikan maka ianya boleh digunakan oleh pelbagai jenis pengguna tanpa mengira latar belakang mereka mahupun tahap kemahiran komputer pengguna tersebut.

11) SoftCrypt juga merupakan suatu sistem yang agak pintar di mana ia boleh mengenalpasti secara otomatik fail – fail yang telah dienkrip olehnya tetapi

dengan anggapan pengguna tidak mengubah jenis sambungan fail (*.sce) yang telah dienkrip oleh SoftCrypt. Dan seterusnya ia juga boleh mengenalpasti jenis algoritma yang telah digunakan ke atas fail tersebut. Maka pengguna tidak perlu bersusah – payah untuk mengingati semula algoritma apakah yang mereka telah gunakan ke atas fail tersebut semasa hendak melakukan dekripsi.

12) Fail – fail yang telah dienkrip oleh SoftCrypt adalah sangat sensitif di mana sebarang pemadaman atau pengubahan ke atas kandungan fail termasuklah jenis sambungan fail itu sendiri *('File's extension')* akan menyebabkan kegagalan semasa melakukan dekripsi ke atas fail itu. Antara kemungkinan – kemungkinan kegagalan yang boleh berlaku ialah SoftCrypt enggan melakukan dekripsi ke atas fail tersebut dengan alasan jenis fail yang tidak sah, hasil dekripsi tidak sama dengan fail asalnya dan mungkin di dalam bentuk yang tidak difahami ataupun kandungan fail yang telah didekrip itu kosong. Dengan itu, SoftCrypt dapat menjamin dan memastikan bahawa data yang telah dienkripkannya tidak mengalami sebarang perubahan atau diuabahsuai oleh pihak – pihak tertentu.

13) Di samping itu, SoftCrypt juga menyediakan pengedit teksnya sendiri yang dikenali sebagai *Composer*. *Composer* ini boleh dijadikan sebagai alternatif kepada Notepad di mana ia turut menyediakan kebanyakan kemudahan untuk mengedit teks seperti di dalam Notepad.

8.4 Kekurangan SoftCrypt

Walaupun SoftCrypt menawarkan banyak kelebihan kepada penggunanya, sistem ini masih lagi mempunyai kekurangannya yang tersendiri. Antara kekurangan atau kelemahan tersebut ialah:

1) Masa pemprosesan yang agak lambat semasa pengguna melalui proses *login* atau proses pendaftaran pengguna baru. Masa pemprosesan bagi proses – proses tersebut bergantung kepada bilangan pengguna yang terdapat di dalam salinan SoftCrypt itu. Jika bilangan pengguna yang banyak maka masa yang lebih lama diperlukan bagi mengesahkan identiti pengguna atau menambah akaun baru ke dalam sistem SoftCrypt. Di samping itu, kadangkala SoftCrypt juga mengambil sedikit masa untuk berpindah dari satu antaramuka ke antaramuka yang lain. Namun begitu, masalah di atas dapat diatasi sekiranya pengguna menggunakan pemproses yang mempunyai kadar capaian rawak memori (RAM) yang tinggi.
2) SoftCrypt hanya menyokong enkripsi/dekripsi ke atas fail yang berjenis teks sahaja (*.txt) di mana pada masa kini, data boleh disimpan di dalam pelbagai bentuk dan jenis fail seperti *'Word Document'* (*.doc), fail *Adobe Acrobat* (*.pdf) dan lain – lain.
3) SoftCrypt perlu digunakan bersama – sama dengan mesin maya Java (*'Java Virtual Machine' - JVM)* serta sebuah perpustakaan sokongan. Pengguna perlu *'install'* dahulu pakej JDK dan pakej perpustakaan Bouncy Castle Crypto sebelum dapat menggunakan sistem ini. Mesin maya Java diperlukan kerana antara ciri utama yang terdapat pada bahasa Java itu sendiri ialah tidak bergantung kepada senibina komputer dan bersifat mudah alih. Oleh yang

demikian, Java tidak menyediakan sebarang kelas yang membolehkan program Java ditukarkan ke dalam bentuk EXE. Walaupun terdapat perisian – perisian sokongan yang menyediakan perkhidmatan penukaran program Java ke dalam bentuk EXE tetapi ianya agak terhad kepada jenis – jenis program yang tertentu sahaja dan sesetengah perkhidmatan tersebut perlu dibayar. Penggunaan perpustakaan sokongan pula diperlukan bagi menyokong perlaksanaan algoritma RSA.

8.5 Cadangan Peningkatan Pada Masa Hadapan

Di antara cadangan - cadangan bagi peningkatan yang boleh dilakukan ke atas SoftCrypt pada masa akan datang ialah seperti berikut:

1) Penambahan algoritma – algoritma enkripsi yang lain seperti AES, Blowfish, El Gamal dan sebagainya ke dalam SoftCrypt boleh dilakukan tanpa perlu melakukan pengubahsuaian ke atas keseluruhan sistem. Ini kerana merujuk kepada senibina sistem SoftCrypt, setiap algoritma enkripsi yang terdapat di dalam SoftCrypt telah ditulis di dalam kod aturcara yang berasingan. Dalam erti kata lain, ianya ditulis di dalam kelas – kelas yang berbeza, setiap kelas algoritma itu bertanggungjawab ke atas segala hal yang berkaitan dengan algoritmanya dan kebanyakan kelas – kelas tersebut tidak bergantungan antara satu sama lain.
2) Menambah jenis – jenis data yang lain yang boleh disokong oleh SoftCrypt seperti *Word Document* dan fail *Adobe Acrobat*.
3) Walaupun sebelum ini dinyatakan bahawa penggunaan mesin maya Java merupakan satu kekurangan bagi SoftCrypt tetapi jika dilihat di dalam konteks yang lebih luas, ia sebenarnya merupakan satu kelebihan bagi SoftCrypt. Di mana, dengan melakukan sedikit pengubahsuaian ke atas kod aturcara SoftCrypt, sistem ini akan dapat dilarikan di atas platform – platform yang lain seperti Linux. Ini dapat meningkatkan pasaran bagi penggunaan SoftCrypt kerana kebanyakan perisian lain hanya disokong oleh satu jenis platform sahaja tetapi SoftCrypt boleh disokong atau dilarikan di atas beberapa platform yang berlainan.

4) Menambah kemudahan atau ciri – ciri di dalam SoftCrypt agar ia boleh menyokong perlaksanaan enkripsi di atas talian Internet atau secara pelayan-pelanggan yang melibatkan penggunaan server.

5) Menyediakan perkhidmatan penjanaan sijil agar ianya boleh digunakan bersama – sama dengan tandatangan digital yang telah dihasilkan oleh SoftCrypt. Dan kemudiannya menghubungkan SoftCrypt dengan pihak *'Certificate Authority (CA)'* tempatan seperti MyCERT atau pihak *CA* umum seperti VeriSign agar tandatangan digital serta sijil tersebut akan lebih dipercayai dan ianya boleh digunakan dengan lebih meluas.

8.6 Masalah - masalah dan Penyelesaiannya

Berikut merupakan beberapa masalah yang dihadapi semasa di dalam proses pembangunan SoftCrypt serta jalan penyelesaian yang telah diambil:

8.6.1 Kekurangan Maklumat

Kekurangan maklumat mengenai pengaturcaraan bagi algoritma enkripsi yang ingin digunakan terutamanya bagi algoritma RSA. Ini kerana algoritma enkripsi yang ingin digunakan itu adalah berjenis kuat dan ianya berada di bawah kawalan undang – undang kerajaan Amerika Syarikat. Kebanyakan sumber – sumber yang dirujuk hanya menceritakan mengenai algoritma enkripsi secara umum serta formula - formula matematik yang digunakan di dalam algoritma berkenaan.

Oleh yang demikian, pembangun sistem menghadapi sedikit masalah semasa hendak melakukan pemetaan dan seterusnya mengimplemenkan algoritma enkripsi ke dalam bentuk kod – kod aturcara. Walaubagaimanapun setelah banyak melakukan pelayaran di Internet serta perbincangan dengan penyelia projek, masalah tersebut telah dapat diatasi.

8.6.2 Kerosakan Perkakasan

Kerosakan terhadap perkakasan utama yang digunakan semasa membangunkan sistem ini di mana komputer mudah alih pembangun sistem telah

mengalami kegagalan cakera keras *('hardisk failure')* beberapa hari sebelum tarikh penghantaran laporan serta sistem ini dilakukan. Pembangun sistem tidak menjangkakan bahawa penggunaan pengedit java bergrafik iaitu Symantec Visual Café memerlukan ruang memori yang begitu besar ketika masa lariannya serta penggunaan sumber – sumber sistem yang banyak di mana ianya tidak mampu ditampung oleh komputer mudah alih tersebut.

Setelah komputer tersebut siap dibaiki, pembangun terpaksa menulis semula sebahagian besar kod – kod aturcara bagi sistem ini berserta laporannya. Maka ianya telah menyebabkan kelewatan di dalam penghantaran laporan dan sistem SoftCrypt ini. Bagi mengelakkan masalah di atas berulang kembali, pembangun sistem telah menggunakan Unit Pemprosesan Pusat (CPU) tambahan bagi menyokong pemprosesan komputer mudah-alih tersebut.

8.6.3 Kekurangan Pengetahuan dan Pengalaman

Kekurangan pengetahuan dan pengalaman di dalam menggunakan bahasa pengaturcaraan Java. Walaupun pembangun sistem mempunyai asas di dalam melakukan pengaturcaraan menggunakan bahasa Java tetapi ianya tidak memadai. Ini kerana di dalam proses pembangunan sistem ini, ia melibatkan penggunaan teknik – teknik pengaturcaraan yang lebih kompleks dan mendalam. Antaranya ialah pengaturcaraan bagi kriptografi, pengaturcaraan bagi mengendalikan urusan input dan output yang dilakukan ke atas fail di mana ianya juga perlu dihubungkan dan diselaraskan dengan antaramuka bagi sistem ini serta pengaturcaraan bagi antaramuka sistem itu sendiri.

Masalah tersebut dapat diatasi dengan banyak melakukan rujukan terhadap buku – buku berkaitan pengaturcaraan Java serta sentiasa merujuk kepada dokumentasi yang telah disediakan oleh Sun Microsystems.

8.6.4 Kekurangan Masa

Masalah lain yang dihadapi adalah berkaitan dengan faktor kekurangan masa di mana semasa di peringkat awal proses pembangunan, sistem ini telah direkabentuk secara umum sahaja tanpa diperhalusi dengan maklumat – maklumat yang terperinci mengenai bagaimana ia akan dibangunkan. Di samping itu terdapat beberapa kekaburan di dalam rekabentuk senibinanya. Ini menyebabkan apabila tiba di fasa implementasi, pembangun sistem mengambil masa untuk memahami dan memperjelaskan kekaburan – kekaburan tersebut. Pembangun sistem juga terpaksa melakukan banyak perubahan ke atas rekabentuk asal sistem SoftCrypt ini bagi memenuhi keperluan kefungsian sistem serta objektif pembangunannya.

8.7 Kesimpulan

Secara keseluruhannya, projek ini telah berjaya mencapai dan memenuhi objektif, tujuan pembangunannya serta keperluan – keperluan sistem yang telah dinyatakan semasa di dalam fasa analisis sistem. SoftCrypt yang terhasil adalah merupakan suatu aplikasi tunggal yang menyediakan perkhidmatan enkripsi, dekripsi serta tandatangan digital dengan menggunakan algoritma DES, RSA dan DSA ke atas fail berjenis teks iaitu data – data di dalam bentuk huruf piawai ASCII.

Enkripsi data yang dilakukan oleh SoftCrypt dijamin selamat kerana ianya disokong dengan penggunaan algoritma enkripsi berjenis kuat. Selain itu, SoftCrypt juga mempunyai antaramuka yang menarik, mudah dikendalikan, senang untuk dipelajari cara penggunaannya serta dilengkapi dengan ciri – ciri keselamatan yang dapat menjamin keselamatan sistem SoftCrypt itu sendiri.

Semasa membangunkan sistem ini, pelbagai pengetahuan baru telah diperolehi. Antaranya ialah mempelajari bukan sahaja teknik – teknik pengaturcaraan lanjutan menggunakan bahasa Java malahan dapat mendalami konsep sebenar dan ciri – ciri keistimewaan yang ditawarkan oleh pakej – pakej Java serta mesin mayanya, meningkatkan kemahiran menggunakan perisian – perisian yang lain seperti Symantec Visual Café, Microsoft Visual J++, CoolIconEditor dan perisian – perisian yang lain serta dapat mempelajari cara penulisan skrip bagi fail MS-DOS yang akan dilaksanakan secara berkelompok *('MS-DOS Batch File')*. Skrip tersebut akan digunakan bagi membolehkan SoftCrypt dilarikan secara *'double-click'* seperti program EXE yang lain.

Melalui projek ini, ianya telah dapat menambahkan pemahaman mengenai konsep – konsep yang berkaitan dengan aspek keselamatan komputer serta mengenai bidang kriptografi. Di samping itu, perlaksanaan projek ini juga merupakan satu ruang untuk mempraktikkan segala apa yang telah dipelajari semasa di dalam kelas sepanjang tempoh pengajian

Kesimpulannya, pengalaman dan pengetahuan yang telah diperolehi semasa membangunkan sistem ini adalah amat berharga dan tidak ternilai. Perlaksanaan projek – projek sebegini memberikan peluang kepada para pelajar untuk mengadaptasikan serta melatih kemahiran mereka di dalam membangunkan sesuatu sistem yang berkualiti. Ianya juga boleh dijadikan sebagai panduan dan persediaan sebelum melangkah masuk ke alam pekerjaan.

RUJUKAN

Pfleeger, Charles P.(1997). *Security in Computing*. 2nd ed. Prentice Hall.

White, Gregory B., Fisch, Eric A., Pooch, Udo W.(1996). *Computer System and Network Security*. CRC Press Inc.

Russell, Deborah, Gangemi, G.T.(1992). *Computer Security Basics*. O'Reilly & Associates Inc.

Jaworski, Jamie.(2000). *Java Security Handbook*. SAMS Publishing.

Knudsen, Jonathan.(1998). *Java Cryptography*. O'Reilly & Associates Inc.

Schneier, Bruce.(1996). *Applied Cryptography*. 2nd ed. John Wiley & Sons Inc.

Sufian Idris, Marini Abu Bakar, Norleyza Jailani.(2001). *OO.Java Pengaturcaraan Berorientasikan Objek Menggunakan Java*. McGraw-Hill.

SoftCrypt

User's Manual

Table of Contents

INTRODUCTION

1.1 What is SoftCrypt ?

SoftCrypt Version 1.0 is a stand - alone encryption system which is able to encrypt and decrypt a text files consisting of all standard ASCII characters using DES or RSA algorithm.

SoftCrypt also provides capabilities for signing and verifying digital signatures attached to the file.

1.2 System Requirement

The minimum requirements for running SoftCrypt version 1.0 are:

- Microsoft Windows 95/98/NT/ME/2000
- Celeron 566Mhz
- 64Mb RAM
- 800 x 600 screen and 16 bit graphics
- 60Mb of free hard drive space
- mouse, keyboard and CD-ROM

1.3 Installing SoftCrypt

Generally, these are the steps involved during SoftCrypt installation:

1) **Install the JDK.**

   Note: The installer for Java 2 SDK v1.4.0 is in the CD. You may used the older version of Java but it advisable to use Java v1.4.0 and above because the JCE provider has been integrated into those Java version.

2) **Unzip and install the Bouncy Castle Crypto library package.**

3) **Copy and paste US_export_policy.jar into Java folder.**

4) **Add the JDK and Bouncy Castle Crypto provider in java.SECURITY file and in autoexec.bat.**

   Note: If you are using the older version of java, you also have to add JCE provider in the files above.

5) **Copy folder labeled SoftCrypt and its shortcut into C drive.**

6) **Copy 3 files from folder Patches (in the diskette) into the SoftCrypt's folder.**

7) **Installation finished. You may now run the SoftCrypt.**

Following are the details instruction on how to install SoftCrypt. If you used a different directory or the older version of Java, you have to customize the path and the instructions below according to your installation.

Run from the CD:

1) Install j2sdk-1_4_0-win into the C drive and follow the self-guiding instructions. The path for the installed jdk should be: *C:\j2sdk1.4.0*

2) Unzip bcprov-jdk14-117 using WinZip and install it into the C drive also. Please make sure the library package is unzip to C:\. You don't have to add extra folder to store the library. The path for the library should be: *C:\bcprov-jdk14-117.*

3) Copy and paste US_export_policy.jar to *C:\ProgramFiles\Java\j2re1.4.0\lib\security.*

4) Copy folder labeled SoftCrypt with its shortcut 

into the C drive. Please make sure, the folder is in the C drive but for the shortcut, you can paste it to the desktop or anywhere else. The path for the SoftCrypt's folder should be: *C:\SoftCrypt.*

Run from the diskette:

5) If you already have a file named "autoexec.bat" in your C drive, copy the contents of autoexec.txt from the diskette and add to *C:\autoexec.bat.* If not,

create a new file using Notepad, paste the contents of autoexec.txt into it and save the file as "autoexec.bat" in your C drive.

6) Add the bold line below to the java.SECURITY file. The file can found in *C:\Program Files\Java\j2re1.4.0\lib\security.*

security.provider.n=org.bouncycastle.jce.provider.BouncyCastleProvider

n is the number of the provider.

7) Copy all the 3 files in folder named "patches" and paste the files into the SoftCrypt's folder that you have copied earlier. Replace/overwrite all the older files. The 3 files are:

   - rsaAppWaitDialog.class
   - rsaApp.class
   - rsaApp.java

8) Now, you can start using the SoftCrypt. To make sure the installation is success, you can try encrypting and decrypting test.txt file provided in the diskette. Please refer to the How to Guide Section on how to encrypt and decrypt files.

Note:

There are examples of autoexec.bat file in the diskette and java.SECURITY file in the cd for your references. If you are using the older version of java, you have to copy sunjce_provider.jar in the cd to *C:\Program Files\Java\j2re1.4.0\lib\ext* and then add the sunjce provider in the java.SECURITY file.

1.4 Starting SoftCrypt

You can start using SoftCrypt by:

- Double-click the shortcut 

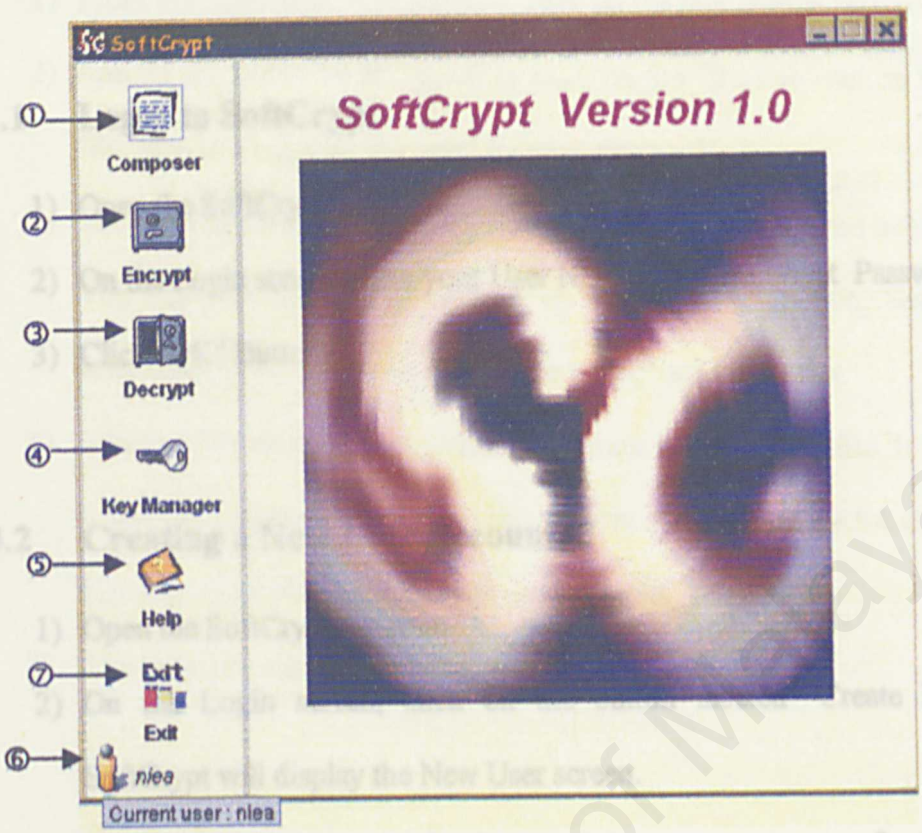
, it will automatically run the SoftCrypt without showing Java environment window.

- If there are errors occurred during SoftCrypt's execution, open the SoftCrypt's folder and double-click the 

. Java's environment window for SoftCrypt will show you what kind of errors had occurred and where the errors occurred.

- SoftCrypt also can be run from the DOS-Prompt but first you must set the path to C:\SoftCrypt>. Then enter java mainFrame.

OVERVIEW (SoftCrypt's Main Screen)



SoftCrypt Main Screen

Num.	Name	General descriptions
①	Composer Button	Open SoftCrypt's Composer.
②	Encrypt Button	To do encryption and/or signed the files.
③	Decrypt Button	To do decryption and/or verify the signature.
④	Key Manager Button	To add, delete or view RSA/DSA public keys.
⑤	Help Button	Open SoftCrypt's Quick Help
⑥	User Details Button	To change user's password or regenerate the keys.
⑦	Exit Button	Close and exit from SoftCrypt

3.1 Login to SoftCrypt

1) Open the SoftCrypt program.
2) On the Login screen, enter your User Name with the correct Password.
3) Click "OK" Button.

3.2 Creating a New User Account

1) Open the SoftCrypt program
2) On the Login screen, click on the button labeled "Create New User". SoftCrypt will display the New User screen.
3) Enter the User Name and the Password that you want to use for that account. Capitalization matters for your User Name and Password.
4) You need to re-enter your password to confirm it.
5) Then, click "OK" button.

3.3 Using the Composer

1) From the main SoftCrypt window, click on the icon button labeled "Composer".
2) SoftCrypt will display the Composer window.
3) You can start using it like a normal Notepad.

3.4 How to Encrypt Files

1) From the main SoftCrypt window, click on the icon button labeled "Encrypt".
2) Specify the file to be encrypted or click on the "Browse" button to select the file. You must specify the exact location for the file.
3) Then, specify the name for the output file and where do you want to save it. You can also do this by clicking on the "Browse" button. SoftCrypt will automatically save your encrypted file with (*.sce) extention.
4) Select which encryption algorithm you want to use on the file. It is advisable to use DES algorithm for your own personal files and use RSA algorithm for files that you will be sharing with other people.
5) When you are ready to encrypt, click the "Encrypt" button.

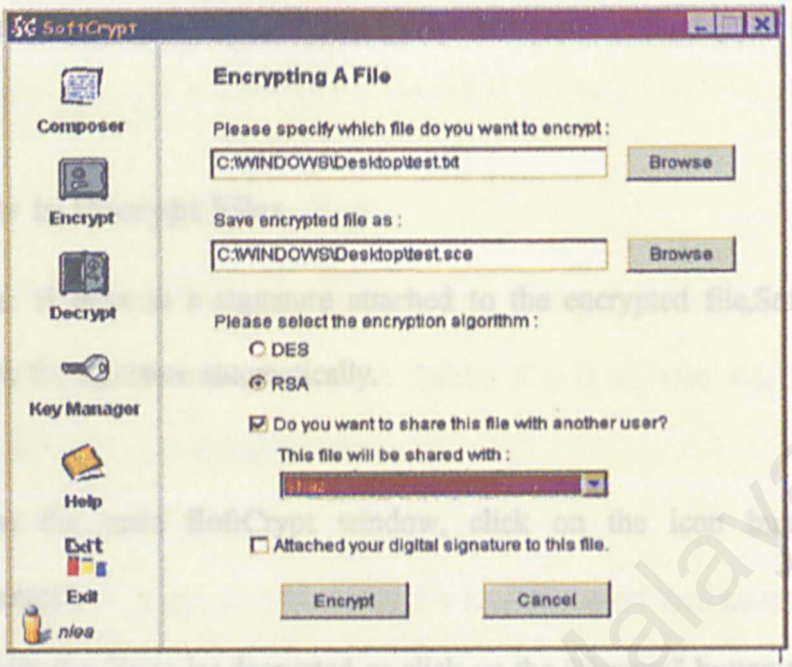
3.5 How to Share The File With Other User

Note: This option is same as encrypting a file for other user. Use this option if you want to make sure only the desired user can decrypt and view the file.

1) From the main SoftCrypt window, click on the icon button labeled "Encrypt".
2) Specify the file to be encrypted or click on the "Browse" button to select the file.
3) Then, specify the name for the output file and where do you want to save it. You can also do this by clicking on the "Browse" button.
4) You must select RSA for the encryption algorithm in order to use this option.
5) Checked the check box labeled "Do you want to share this file with another user?" to enable the other user list.

6) Select the other user from the list.

7) When you are ready to encrypt, click the "Encrypt" button.



Encrypt a file for other user

3.6 How to Sign a File

1) From the main SoftCrypt window, click on the icon button labeled "Encrypt".

2) Specify the file to be encrypted or click on the "Browse" button to select the file.

3) Then, specify the name for the output file and where do you want to save it. You can also do this by clicking on the 'Browse" button.

4) You must select RSA for the encryption algorithm in order to use this option.

5) You can either share the file with other user or use this option for your owned purpose. If you choose to share the file with another user, please refer to steps 5 and 6 in "How to share the file with other user", then continue with the steps below.

6) To include your digital signature in the encrypted file, checked the check box labeled "Attached your digital signature to this file."

7) When you are ready to encrypt, click the "Encrypt" button.

3.7 How to Decrypt Files

Note: If there is a signature attached to the encrypted file,SoftCrypt will verify the signature automatically.

1) From the main SoftCrypt window, click on the icon button labeled "Decrypt".

2) Specify the file to be decrypted or click on the "Browse" button to select the file. If you are using the "Browse" button, it ca only detect and show you the encrypted files with (*.sce) extension.

3) Then, specify the name for the output file and where do you want to save it. You can also do this by clicking on the "Browse" button.

4) When you are ready to decrypt, click the "Decrypt" button.

3.8 Using the Key Manager

1) From the main SoftCrypt window, click on the icon button labeled "Key Manager".

2) You will be prompted with the Key Manager dialog box.

3) Select the public key list that you want to view or edit.

4) Click on the button labeled "Open".

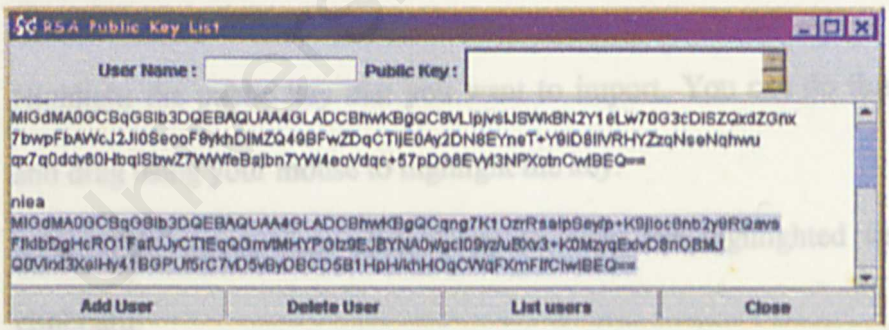
5) Then, SoftCrypt will display the appropriate public key list window.

6) For each window, you can add or delete other user's public key and view the list of users which SoftCrypt already have their public key. The other users which has been automatically added by SoftCrypt cannot be deleted.

3.9 How to Export Public Key

Note: The following steps can be applied to both public key lists, which is RSA Public Key List and DSA Public Key List. Use this function to distribute your public key to other user.

1) Please refer to steps 1 - 5 in "Using the Key Manager" then continue with the steps below.
2) From the public key list window, highlight only the public key that you want to export. You can do this by click and drag using your mouse to highlight the key.



Highlighting the niea's public key

3) Then, press (Ctrl+C) to copy the selected public key to the clipboard.
4) Open the Composer. Please refer to steps 1 - 3 in "Using the Composer".
5) From the Composer window, select Edit menu, then click Paste.

6) Select File menu, then click Save As to save the public key into a file. For easier to identify the file, use the name of the public key's owner as the file name.

7) You can now distribute the file that contains the public key to other user.

3.10 How to Import Public Key

Note: The folowing steps can be applied to both public key lists, which is RSA Public Key List and DSA Public Key List. Use this function to add other user's public key into SoftCrypt public key list

1) Please refer to steps 1 - 3 in "Using the Composer" then continue with the steps below.

2) From the Composer window, select File menu, then click Open to open the file containing the public key that you want to add to the SoftCrypt public key list.

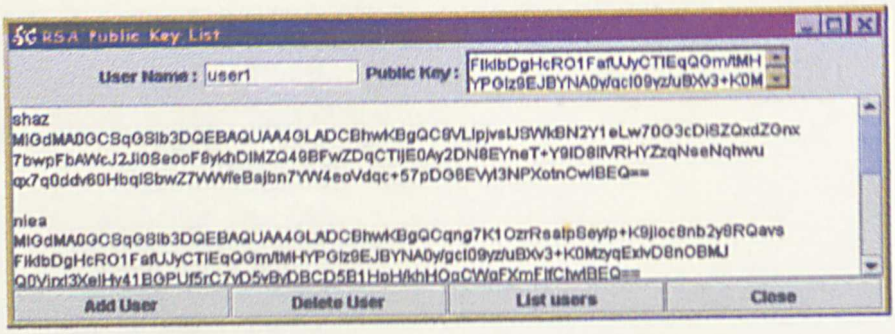
3) Highlight the public key that you want to import. You can do this by click and drag using your mouse to highlight the key.

4) Select Edit menu, then click Copy to copy the highlighted key to the clipboard.

5) Open the related public key list. Please refer to steps 1 - 5 in "Using the Key Manager".

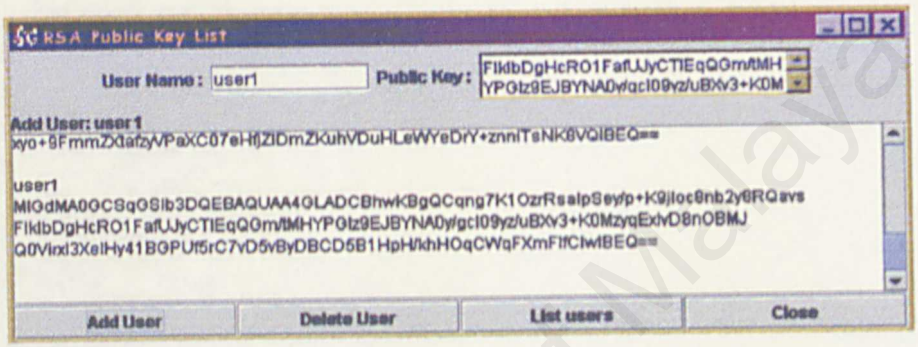
6) From the public key list window, enter the user name in the User Name text field.

7) Put the cursor in the Public Key text area. Then, press (Ctrl+V) to paste the public key from the clipboard.



Adding new public key into SoftCrypt public key list

8) Click on "Add User" button. You can now see the user name and the public key is in the list.



After adding user1 into the list

9) The public key list will be save when you close the public key list window by clicking the "Close" button.