

Daftar Isi

Daftar Isi.....	1
Daftar Gambar	2
Daftar Tabel	3
Bab 1 Pendahuluan.....	4
Bab 2 Sistem Operasi Symbian	5
2.1 Sejarah.....	6
2.2 Arsitektur Sistem Operasi	6
2.3 Klasifikasi Sistem Operasi	8
Bab 3 Tinjauan Sistem Keamanan	12
3.1 Keamanan Platform Aplikasi.....	12
3.2 Keamanan Komunikasi Client-Server.....	13
3.3 Antarmuka Pengguna	14
3.4 Keamanan Dalam Menjalankan Plug-ins	15
3.5 Keamanan Instalasi Aplikasi.....	16
3.6 Keamanan Data.....	17
3.1.1 Modul Kriptografi	19
3.1.2 Modul Manajemen Sertifikasi	19
Bab 4 Potensi Ancaman Terhadap Sistem Keamanan	20
Bab 5 Penutup	21
Daftar Pustaka	21

Daftar Gambar

Gambar 1: Arsitektur Sistem Operasi Symbian	7
Gambar 2: Proses penanganan input oleh antarmuka pengguna	15
Gambar 3: Framework ECOM	15
Gambar 4: Prosedur penandaan aplikasi oleh Symbian	16
Gambar 5: Ilustrasi mekanisme penginstalan aplikasi.....	17
Gambar 6: Arsitektur sistem keamanan data.....	18
Gambar 7: Komponen utama modul manajemen sertifikasi.....	20

Daftar Tabel

Tabel 1: (API) Symbian Umum	9
Tabel 2: (API) Symbian Umum Tergantikan	9
Tabel 3: (API) Symbian Opsional.....	10
Tabel 4: (API) Symbian Opsional Tergantikan.....	11

Bab 1 Pendahuluan

Sistem operasi Symbian (Symbian OS) populer sebagai salah satu sistem operasi peralatan mobile. Penggunaannya dari sisi vendor mobile phone pun terdiri dari berbagai jenis. Saat ini Symbian ini sendiri dimiliki oleh Ericsson (15,6%), Nokia (47,9%), Panasonic (10,5%), Samsung(4,5%), Siemens / BenQ (8,4%), Sony Ericsson (13,1%).[4] Ketika semakin berkembangnya penggunaan dan aplikasi pada jenis produk mobile phone ini maka keamanan menjadi hal yang penting. Munculnya beberapa aplikasi yang berniat memberikan efek negatif di peralatan telepon bergerak menjadi tantangan baru bagi sistem operasi ini untuk memperlengkapi diri dengan sistem keamanan yang terintegrasi sehingga tetap dapat memfasilitasi kebutuhan dari berbagai vendor produk dan dari sisi pengembang aplikasi.

Makalah ini bertujuan untuk memberi gambaran mengenai bagaimana bentuk arsitektur sistem keamanan pada Symbian OS serta keandalannya menangani kebutuhan aspek keamanan bagi berbagai vendor produk peralatan telepon bergerak. Akan dijelaskan pula bagaimana Symbian OS ini mendukung aspek-aspek keamanan seperti: *data confidentiality*, *integrity* dan *authentication*

Pada versi terbaru, Symbian OS 9, dikembangkan konsep baru sistem keamanan *capability-based security* yang menjamin mekanisme instalasi yang aman yang mendukung pengembang aplikasi. Hal ini menjadi salah satu poin pembahasan pada makalah ini. Sebagai tambahan, pada makalah ini juga diberikan contoh beberapa aplikasi memanfaatkan celah apa sistem keamanan untuk melakukan tindakan negatif yang telah diidentifikasi dan gambaran bagaimana dia bekerja pada lingkungan sistem operasi ini.

Bab 2 Sistem Operasi Symbian

Saat ini Symbian OS banyak telah banyak digunakan oleh berbagai vendor produk peralatan komunikasi mobile pada berbagai jenis produk mereka yang bervariasi. Variasi dari sisi hardware ini dimana Symbian OS diimplementasi dapat dimungkinkan karena sistem operasi ini memiliki antarmuka pemrograman aplikasi (*Application Programming Interface*; API). API mendukung terhadap komunikasi dan tingkah laku yang umum pada *hardware* yang dapat digunakan oleh objek aplikasi lain. Hal ini dimungkinkan karena API merupakan objek antarmuka yang didefinisikan pada level aplikasi, yang berisikan prosedur dan fungsi (dan juga variabel serta struktur data) yang mengelola/memanggil kernel dimana sebagai penghubung antara *software* dan *hardware*. Dengan adanya standar API ini membantu pihak pengembang untuk melakukan penyesuaian atas aplikasi yang dibuatnya agar dapat diinstal pada produk telepon bergerak yang bermacam-macam.

Mirip seperti system operasi desktop, Symbian OS mampu melakukan operasi secara *multithreading*, *preemptive multitasking* dan pengamanan terhadap memori [5]. Dan semua pemrograman pada Symbian dilakukan secara *event-based*, artinya hardware CPU menjadi tidak aktif ketika tidak ada inputan berupa aktifitas tertentu. Namun perlu dipahami sistem operasi ini memang ditujukan untuk diinstal pada peralatan *mobile* dengan keterbatasan sumber daya. *Multithread* dan *multitasking* memberikan kemampuan Symbian OS untuk menjalankan lebih dari satu aplikasi sekaligus. Namun khusus ini, adanya *preemptive multitasking* kernel akan memberi tiap-tiap program suatu pembagian waktu pemrosesan yang dilakukan bergantian dengan cepat sehingga nampak bagi pemakai seolah-olah proses ini dieksekusi secara bersamaan. Untuk itu telah didefinisikan algoritma penjadwalan berdasar prioritas tertentu untuk menentukan proses mana yang berjalan terlebih dahulu dan proses apa berikutnya serta berapa banyak waktu akan jadi diberi [7].

Symbian OS sendiri bukanlah software yang sifatnya *open source* secara penuh. Hal ini dikarenakan meskipun ketersediaan API dan dokumentasinya, yang banyak membantu pihak pengembang aplikasi untuk membuat software yang berjalan di atas sistem operasi ini, dipublikasi untuk umum namun tidak untuk *source code* sendiri[4].

2.1 Sejarah

Pada tahun 1980, berdiri perusahaan pengembang software Psion yang didirikan oleh David Potter. Produk dari Psion saat itu diberi nama EPOC. Sistem operasi ini lebih difokuskan pada penggunaannya di telepon bergerak. Symbian sendiri merupakan kerjasama yang dibentuk pada tahun 1998 antara perusahaan Ericsson, Nokia, Motorola dan Psion untuk mengeksplorasi lebih jauh kekonvergensi antara PDA dan telepon bergerak. Hingga akhirnya Psion menjual sahamnya pada tahun 2004. Hasil dari kerjasama ini menghasilkan EPOC Release 5 yang berikutnya dikenal sebagai Symbian OS v5. Sistem operasi ini sudah mulai mengintegrasikan kebutuhan implementasi aplikasi pada perangkat seperti Personal Data Assistant (PDA) selain telepon bergerak. Dengan kata lain mendukung perangkat yang lebih sering dikenal sebagai *smartphone*.

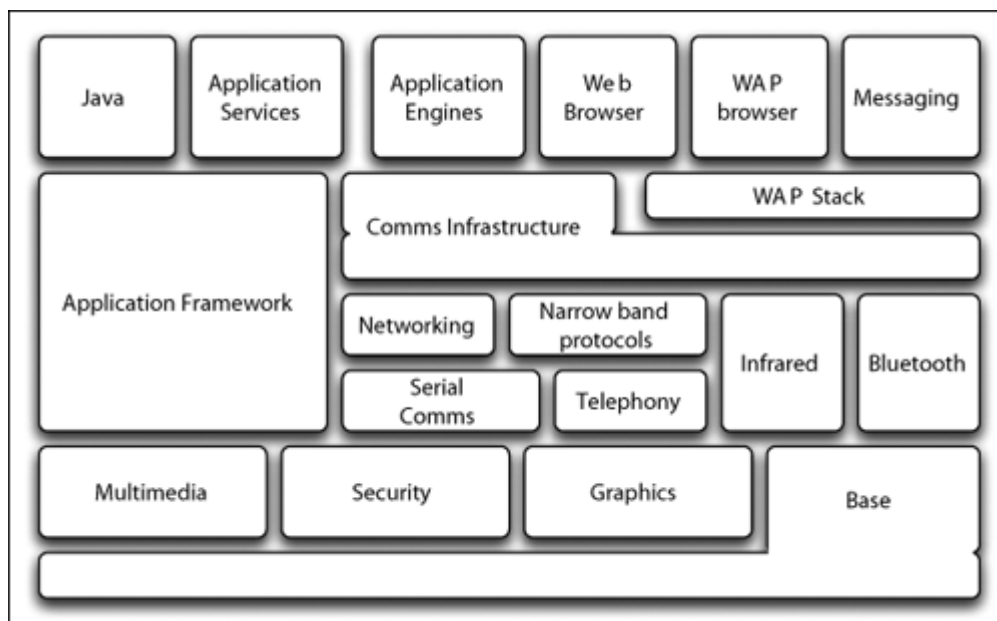
Berikutnya muncul versi-versi terbaru dari Symbian OS. Hingga muncul Symbian v6.0 yang merupakan versi pertama dari Symbian OS yang sifatnya terbuka (*open*) karena pada sistem ini dapat dilakukan instalasi software oleh berbagai pengembang aplikasi. Versi ini juga terkenal dengan nama ER6.

Pada awal tahun 2005 muncul Symbian OS v9.1 dengan sistem keamanan platform baru yang dikenal sebagai *capability-based security*. Pada dasarnya sistem ini mengatur hak akses bagi aplikasi yang akan diinstal pada peralatan dalam hal mengakses API. Hal ini akan dijelaskan lebih lanjut pada bagian selanjutnya. Berikutnya pada Symbian OS v9.2 dilakukan perbaharuan pada teknologi konektivitas Bluetooth dengan digunakannya Bluetooth v.2.0. Sedangkan yang terbaru, Symbian mengeluarkan Symbian OS v9.3 (dirilis pada tanggal 12 Juli 2006) telah mengusung teknologi wifi 802.11 dan HSDPA sebagai bagian dari komponen standarnya.

2.2 Arsitektur Sistem Operasi

Secara umum arsitektur Symbian OS sendiri dapat gambarkan menjadi empat lapisan atau grup berdasarkan penggunaan API yang tersedia, yaitu: lapisan pendukung aplikasi (*application utility layer*), lapisan layanan dan framework antarmuka grafis (*GUI framework*), lapisan komunikasi, dan system API dasar.

- **Lapisan pendukung aplikasi:** terdiri dari berbagai pendukung yang berorientasi pada aplikasi. Hal ini memungkinkan aplikasi lain (diluar sistem operasi) untuk berintegrasi dengan aplikasi dasar yang tersedia pada sistem operasi. Bentuk layanan lain termasuk proses pertukaran data dan manajemen data.
- **Lapisan layanan dan framework antarmuka grafis:** merupakan framework API yang tersedia untuk memberi dukungan terhadap penanganan input user secara grafis maupun suara yang dapat digunakan oleh aplikasi lain.
- **Lapisan komunikasi:** tentu saja sebagai sistem operasi yang fokus diimplementasi pada peralatan komunikasi mobile, Symbian OS memiliki kumpulan API yang fokus pada lapisan komunikasi. Bagian teratas pada lapisan ini terdapat dukungan pencarian dan pengiriman pesan teks. Berikutnya adalah antarmuka yang memberi dukungan komunikasi seperti Bluetooth dan infrared (IrDA) serta USB. Yang terakhir pada lapisan ini adalah protocol komunikasi berupa TCP/IP, HTTP, WAP dan layanan telepon.
- **Lapisan sistem API dasar:** merupakan kumpulan API yang mendukung pengaksesan data memori, tanggal dan waktu, serta sistem dasar lainnya.



Gambar 1: Arsitektur Sistem Operasi Symbian

2.3 Klasifikasi Sistem Operasi

Klasifikasi yang dijelaskan disini adalah klasifikasi berdasar fungsionalitas dan hak akses dari API tertentu. Tujuan dari pendefinisian sistem ini selain untuk membedakan API mana saja yang bisa diakses oleh aplikasi yang dibuat oleh pihak pengembang aplikasi namun tetap memelihara integrasi dari layanan yang disediakan bagi pihak pengembang aplikasi dengan API yang umum digunakan. Hal ini juga dilakukan untuk memaksimalkan interoperabilitas antara berbagai produk yang menggunakan Symbian OS.

Terdapat empat kategori dalam klasifikasi API yang tersedia, yaitu: (API) Symbian Umum, (API) Symbian Opsional, (API) Umum Tergantikan dan (API) Opsional Tergantikan.

Symbian Umum

Komponen ini merupakan komponen (API) inti dari Symbian OS. Setiap pengembang aplikasi dapat berasumsi bahwa komponen ini terdapat pada setiap versi Symbian OS sehingga dapat digunakan pada setiap perangkat telepon bergerak yang menggunakan Symbian OS sebagai sistem operasinya. Dengan kata lain setiap kode program yang hanya menggunakan API pada kategori ini dapat dikompil dan dijalankan tanpa kesalahan pada setiap telepon yang menggunakan Symbian OS. Dengan adanya lisensi kerjasama, pengembang aplikasi dapat menambahkan dengan syarat tidak mengganti ataupun mengubah fungsi API standar yang dikategorikan pada bagian ini.

Category	Component	Symbian OS version				
		v6.0	v6.1	v7.0	v7.0s	v8.0
Common Symbian	Application Utilities					
	C32 Server					
	ESOCK Server					
	ETEL Server					
	File server					
	INSOCK	N/A	N/A			
	Kernel					
	MBuf Manager	N/A	N/A	N/A	N/A	
	Root Server	N/A	N/A	N/A	N/A	
	User libraries					
	TCP/IP					
	TLS					
	UIKON					
	Window Server					

Tabel 1: (API) Symbian Umum

Symbian Umum Tergantikan

Komponen yang memerlukan kostumisasi dari komponen Symbian Umum yang diperlukan untuk bekerja dengan ROM dari sistem dimana ia diinstal. Komponen ini merupakan komponen yang bekerja pada low-level dari hardware tertentu. Untuk mendapatkan komponen ini pihak pengembang aplikasi memerlukan lisensi dengan pihak Symbian karena versi komponen ini disediakan oleh pihak Symbian. Namun pada dasarnya komponen ini merupakan komponen standar (umum) yang tersedia pada semua versi Symbian OS.

Category	Component	Symbian OS version				
		v6.0	v6.1	v7.0	v7.0s	v8.0
Common replaceable	Comms modules (CSYs)					
	Filesystem drivers					
	J2ME Java and/or PersonalJava					
	Font/bitmap server					
	Screen drivers					
	Telephony driver					

Tabel 2: (API) Symbian Umum Tergantikan

Symbian Opsional

Komponen-komponen ini sifatnya opsional (tidak selalu ada) pada semua versi Symbian OS. Namun jika tersedia, maka pengembang aplikasi mendapat jaminan bahwa aplikasinya dapat menggunakan API pada kategori ini pada versi Symbian OS yang sama.

Category	Component	Symbian OS version				
		v6.0	v6.1	v7.0	v7.0s	v8.0
Optional Symbian	Bio Messaging					
	Bluetooth	N/A				
	Calendar engine					
	Comms channels	N/A	N/A	N/A	N/A	
	Comms logging					
	Contacts engine					
	Cryptography library					
	Database engine					
	FORM layout engine					
	Help					
	IrDA					
	Messaging engine					
	Internet mail MTM					
	GSM SMS Stack					
	Store					
	SyncML	N/A	N/A			
	Task scheduler				N/A	N/A
	vCard and vCal					
	Web browser		N/A	N/A	N/A	N/A

Tabel 3: (API) Symbian Opsional

Symbian Opsional Tergantikan

Bentuk kategori ini mirip dengan kategori Symbian Opsional adalah kumpulan API yang tidak terikat dengan API umum yang ada pada versi Symbian OS dan dapat ditambahkan oleh pihak pengembang dengan suatu lisensi dari pihak Symbian.

Category	Component	Symbian OS version				
		v6.0	v6.1	v7.0	v7.0s	v8.0
Optional replaceable	Media server				N/A	N/A
	Multimedia	N/A	N/A	N/A		
	Browsing framework	N/A	N/A			
	Code converters					
	Comms utilities					
	Content handling framework	N/A	N/A	N/A	N/A	
	Fax MTM					
	FTP, Telnet					
	Grid layout engine					
	HTTP Client	N/A	N/A		N/A	N/A
	Internet Protocols	N/A	N/A		N/A	N/A
	Application Level Internet Protocols	N/A	N/A	N/A		
	Locales					
	Logging engine				N/A	N/A
	Networking options	N/A	N/A			
	Onboard converters					
	Other app engines					
	Printer drivers					
	WAP Stack					
	MMS	N/A	N/A			
	WAP-browser engine					
	Alarm/World server					
	Data-type recognizers	N/A	N/A			
	Serial comms utilities					
	On-board connectivity				N/A	N/A
	PC Connectivity toolkit	N/A	N/A	N/A		
	Comms database					
	Security					
	View server					
	System agent					
	Power					
	C standard library					
Wserv plugin						

Tabel 4: (API) Symbian Opsional Tergantikan

Bab 3 Tinjauan Sistem Keamanan

Karena fleksibilitas yang disediakan bagi pihak pengembang aplikasi setelah Symbian OS mulai bersifat *open*, maka hal ini menyebabkan rentannya sistem keamanan Symbian OS akibatnya adanya kemudahan pengaksesan sumber daya dan komponen fungsionalitas dari telepon. Oleh karena itu pada dasarnya pengamanan yang dilakukan oleh Symbian OS yaitu membatasi/memfilter hak akses dari aplikasi yang akan diinstal pada sistem operasi ini.

Dengan membatasi pembahasan pada sistem keamanan Symbian OS terbaru, versi 9, terkait dengan pengamanan terhadap keberadaan aplikasi maka terdapat lima fokus utama yang akan dibahas pada makalah kali ini yaitu:

1. Keamanan platform aplikasi
2. Komunikasi client/server
3. Antar muka pengguna
4. Plug-ins
5. Instalasi aplikasi
6. Data

3.1 Keamanan Platform Aplikasi

Platform aplikasi merupakan struktur standar yang disediakan oleh sistem operasi dalam hal ini Symbian OS yang menjelaskan ketersediaan dan kaitan berbagai antarmuka, API, yang dapat digunakan dalam pengembangan aplikasi.

Keamanan platform dapat dicapai dengan dengan cara:

- menjaga integritas dari telepon dan software yang terinstal
- menjaga kerahasiaan (*confidentiality*) data khusus seperti file sistem
- melakukan kontrol terhadap operasi sensitif dan antarmuka yang disediakan

Untuk merealisasikan ketiga hal ini menerapkan fitur utama yaitu model *capability-based* dan pengurungan data (*data caging*)

Ketika suatu aplikasi diinstal pada Symbian OS, aplikasi ini akan diberikan hak untuk melakukan suatu operasi. Kemampuan (*capability*) untuk melakukan operasi ini diatur oleh kernel, inti dari sistem operasi yang berhubungan langsung dengan sumber daya hardware (CPU, memory dan peralatan input/output) [7].

Kernel ini sendiri tidak dapat diubah dengan adanya instalasi suatu aplikasi. Model *capability-based* ini merupakan pendefinisian level terhadap hak pengaksesan/penggunaan dari kernel dari penggunaan antarmuka API. Capability ini terdiri dari dua:

- *basic capability*, yaitu hak pengoperasian yang memerlukan ijin dan mudah dimengerti oleh pengguna
- *extended capability*, yaitu hak pengoperasian seijin pihak Symbian. Biasanya merupakan operasi yang jauh lebih kompleks.

Namun tidak semua operasi yang melibatkan sumber daya hardware bekerja berdasar hak akses yang dikategorikan diatas karena dianggap cukup aman. API yang bekerja berdasar pembatasan tersebut hanya berkisar 40 persen dari total API yang tersedia.

Terkait dengan model di atas.maka file-file yang diakses/bekerja dalam pembatasan ini dikumpulkan pada folder tersendiri. Metoda ini yang dimaksudkan dengan *data caging*. Folder-folder yang dilindungi ini yaitu:

- \Sys. Folder ini berisikan file-file sistem penting dan dapat dieksekusi. File ini hanya dapat dimodifikasi oleh kernel, server file atau file-file penginstal aplikasi.
- \Resource. Folder ini berisi file yang digunakan bersama oleh beberapa aplikasi. File ini hanya dapat dimodifikasi oleh file penginstal aplikasi
- \Private. Merupakan folder berisikan file-file yang telah ditandai/diijinkan oleh pengguna untuk melakukan operasi tertentu.

3.2 Keamanan Komunikasi Client-Server

Yang dimaksud dengan framework client/server pada Symbian OS adalah framework yang memungkinkan suatu aplikasi untuk memberikan layanan kepada beberapa aplikasi yang lain. Aplikasi ini bertugas sebagai server yang menangani data/proses yang dibutuhkan dan diberikan oleh aplikasi lain sebagai client [11].

Arsitektur client-server pada Symbian OS menggunakan *Interprocess Communication* (IPC) sebagai basisnya [8]. IPC merupakan kapabilitas yang tersedia oleh sistem operasi yang mengizinkan suatu proses berkomunikasi dengan proses lain. Proses lain tersebut dapat berjalan pada computer yang sama atau computer yang terkoneksi melalui jaringan [9]. Dengan adanya fasilitas ini suatu aplikasi dapat mengontrol dan mengakses data yang sama dengan aplikasi lain.

IPC juga didesain juga untuk melakukan manajemen batasan penggunaan memori. Hal ini dilakukan oleh kernel sehingga dapat dipercaya, tidak akan dipengaruhi oleh aplikasi tambahan yang terinstal, dan juga aman dalam melakukan komunikasi.

Setiap file aplikasi yang dapat dieksekusi memiliki *Secure Identifier* (SID) yang menentukan proses mana yang dapat dijalankan. SID merupakan hasil dari proses penandaan oleh pihak Symbian (penjelasan lebih jauh pada bagian 3.5). Selain SID juga terdapat *Vendor Identifier* (VID) yang mengidentifikasi pembuat aplikasi.

Untuk singkatnya IPC menggunakan kebijakan keamanan standar yang dapat digunakan oleh server (aplikasi), sehingga dapat didefinisikan dalam proses pembuat oleh pihak pengembang aplikasi. Kebijakan itu melibatkan hal:

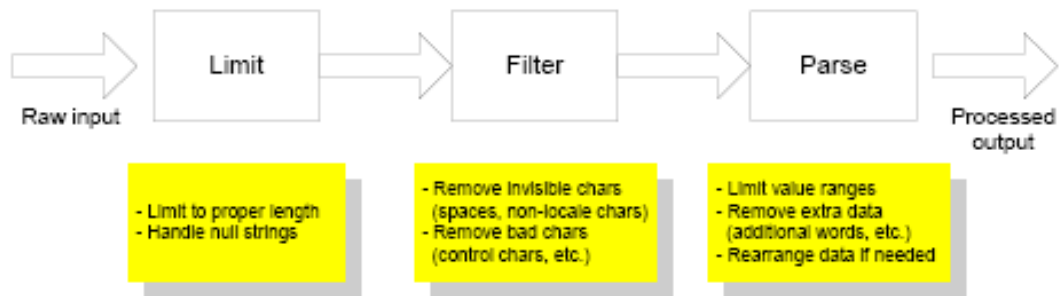
- Kapabilitas apa saja yang diberikan dari pihak client (caller)
- SID dan VID client mana yang dapat menerima layanan
- Tindakan apa saja yang dapat/harus dilakukan ketika informasi diberikan oleh client

3.3 Antarmuka Pengguna

Untuk aplikasi yang menyediakan antarmuka pengguna, berikut hal-hal yang perlu diperhatikan:

- aplikasi mana saja yang mempunyai akses terhadap inputan dari pengguna
- aplikasi yang berbeda dapat memiliki tampilan antarmuka yang sama

Secara khusus, menggunakan fitur *TrustedUI*, menyediakan dialog input password yang didesain agar user dapat mengenali secara mudah aplikasi yang sedang berjalan. Antarmuka yang tersedia pada Symbian OS didesain untuk mengatasi proses pengimputan untuk menjamin kesesuaian input dengan yang dibutuhkan dengan melewati beberapa fase validasi input. Gambar di bawah ini mengilustrasikan kurang lebih proses tersebut.

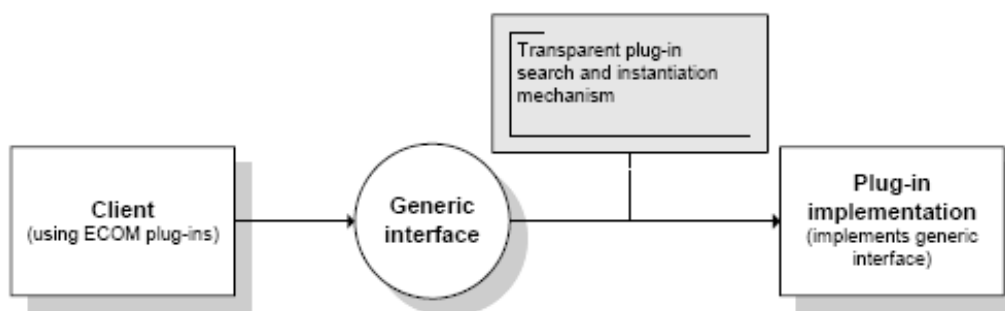


Gambar 2: Proses penanganan input oleh antarmuka pengguna

3.4 Keamanan Dalam Menjalankan Plug-ins

Mulai versi 7, Symbian OS menerapkan metode yang mengatur bagaimana cara plug-ins berhubungan dengan sistem. Metode ini disebut EPOC Component Object model (ECOM). ECOM berbasiskan arsitektur client/server dan menyediakan layanan untuk menginstansiasi dan menghapus (objek) plug-ins yang berjalan.

Dalam penggunaan plug-ins ini, platform aplikasi juga melakukan verifikasi menggunakan model kapabilitas sebelum pluggin diaktifkan, berjalan bersama aplikasi. Hal ini perlu karena ketidaksesuaian dan kerusakan plug-ins dapat mengakibatkan kebocoran keamanan sistem atau bahkan terjadinya *crash*. Plug-ins tidak akan dijalankan apabila suatu proses (aplikasi) yang menggunakannya memiliki kapabilitas yang lebih terbatas dari plug-ins itu sendiri.

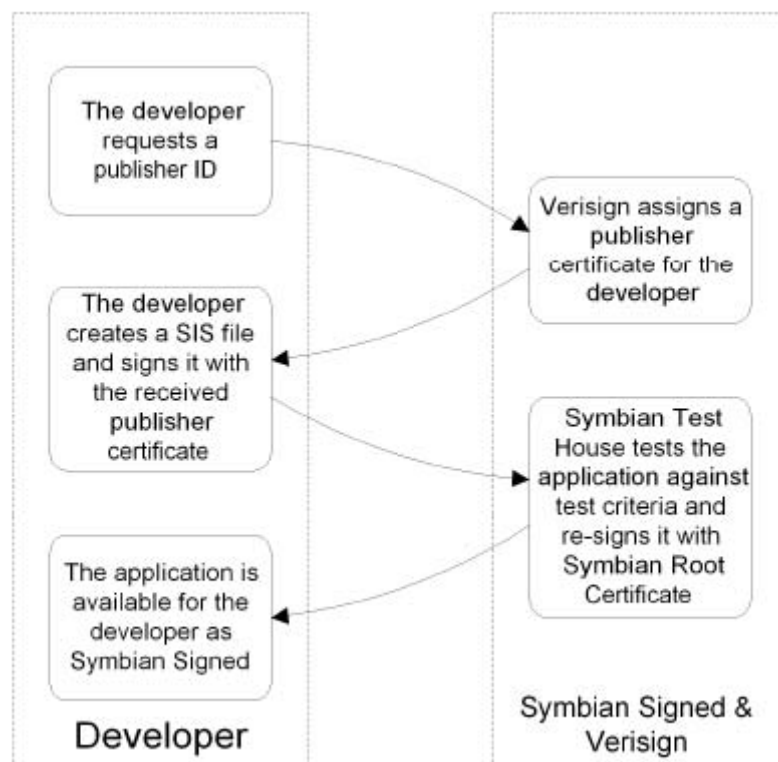


Gambar 3: Framework ECOM

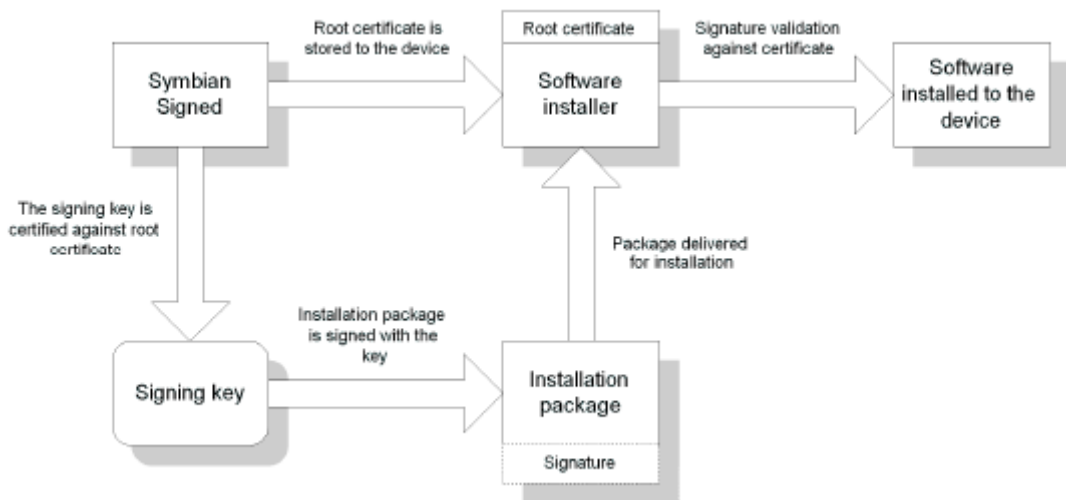
3.5 Keamanan Instalasi Aplikasi

Sistem instalasi aplikasi pada Symbian OS menyediakan proses instalasi yang aman bagi aplikasi. Sistem ini mendukung proses otentifikasi software menggunakan penandaan digital dan sertifikasi untuk memastikan bahwa aplikasi yang diinstal berasal dari pengembang aplikasi dikenali oleh pihak Symbian (Gambar 4).

Pada dasarnya proses penandaan ini menggunakan model Public Key Infrastruktur (PKI) dimana file SIS akan ditandai menggunakan *private key* milik pengembang. Sedang dalam proses instalasinya, symbian OS akan memverifikasi isi file yang akan diinstal dan pembuatnya menggunakan *public key* yang bersesuaian. Namun demikian isi file instalasi ini tidak dilakukan proses enkripsi.



Gambar 4: Prosedur penandaan aplikasi oleh Symbian



Gambar 5: Ilustrasi mekanisme penginstalan aplikasi

Terdapat dua macam paket aplikasi yang dapat diinstal pada Symbian OS, yaitu: paket SISX dan arsip file Java dalam bentuk JAR. Jika dalam proses intallasi proses pencocokan sertifikat sesuai maka proses tersebut dapat dilakukan. Aplikasi akan meminta kapabilitas untuk melakukan sejumlah operasi khusus, menyimpan file yang dapat dieksekusi pada direktori \sys\bin dan memastikan tidak ada aplikasi dengan *Secure Identifier* (SID), yang dikirim bersama saat proses penandaan digital oleh pihak Symbian, yang sama. Hal ini sedikit berbeda untuk aplikasi yang belum melakukan penandaan digital hanya mendapatkan kapabilitas standar namun tetap memerlukan konfirmasi dari pengguna.

3.6 Keamanan Data

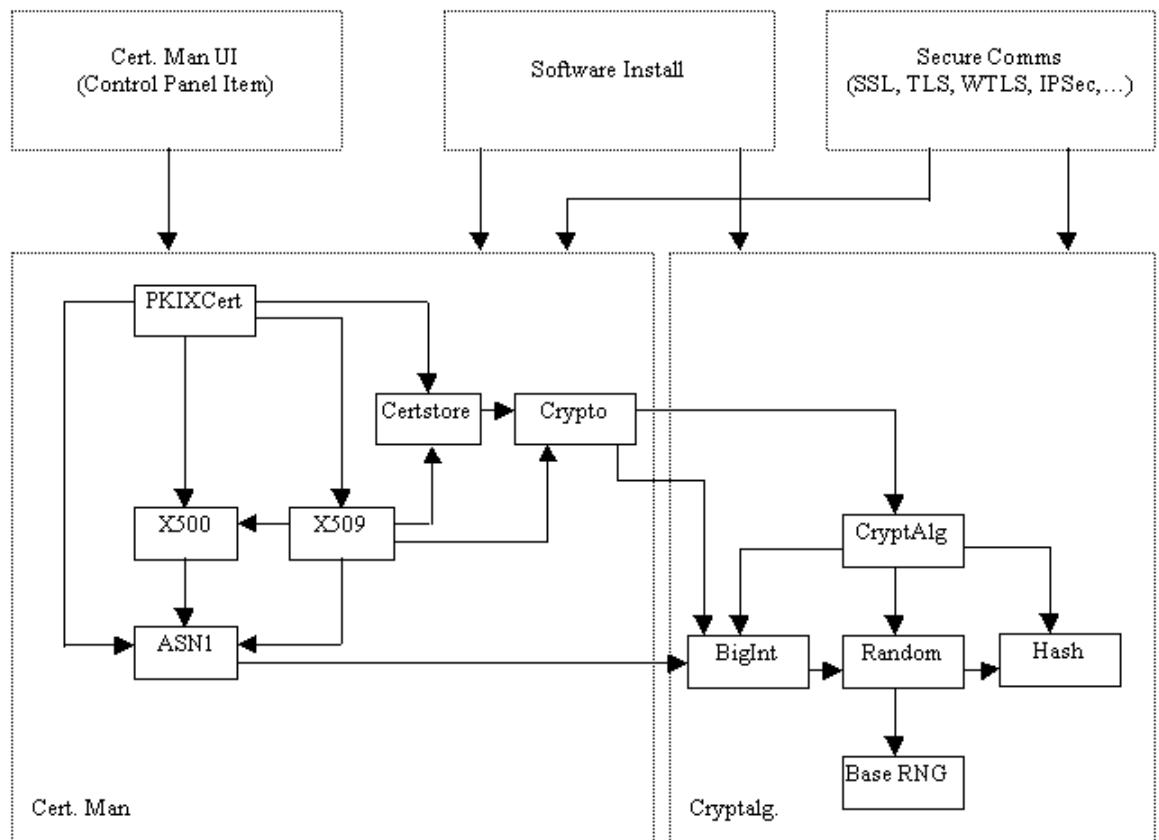
Sedangkan terkait dengan data yang digunakan dalam melakukan komunikasi sistem operasi ini menyediakan fasilitas yang mendukung/memelihara kerahasiaan data (*confidentiality*), integritas data (*integrity*) dan keaslian data (*authentication*). Hal ini dimungkinkan dengan penyediaan beberapa komponen API standar yang menangani algoritma kriptografi, pembangkitan kunci *hash*, pembangkitan bilangan acak, dan manajemen sertifikasi.

Arsitektur keamanan sistem Symbian OS pada dasarnya terdiri dari dua komponen (Gambar 6) [12] yang menjadi bagian dari API *Security*.

- Manajemen Sertifikasi (certman)
- Kriptografi (cryptalg)

Komponen ini juga menjadi bagian pada API lain ataupun aplikasi level tinggi seperti:

- Manajemen sertifikasi pada panel control antarmuka pengguna
- Instalasi software
- Pengamanan port komunikasi. Comms (misal SSL/TTL, WTLS, IPSec)



Gambar 6: Arsitektur sistem keamanan data

3.1.1 Modul Kriptografi

Modul Kriptografi yang tersedia pada Symbian OS terdiri dari komponen-komponen berikut:

- Algoritma kriptografi yang digunakan untuk melakukan proses enkripsi dan dekripsi data. Untuk model kriptografi menggunakan kunci simetris tersedia: DES, 3DES, RC2, RC4, dan RC5. Sedang yang menggunakan kunci asimetris: RSA, DSA, dan DH
- Fungsi hash: MD5, SHA1, and HMAC
- Pembangkit pseudo-acak bilangan untuk menghasilkan kunci kriptografi.

Perlu diingat bahwa pada sistem operasi Symbian OS, proses enkripsi dan dekripsi informasi bukan bagian dari file sistem secara default.

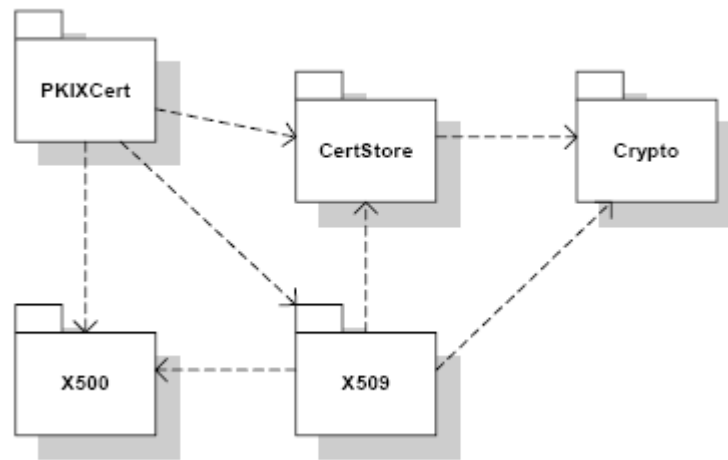
3.1.2 Modul Manajemen Sertifikasi

Modul manajemen sertifikasi pada sistem operasi Symbian OS pada umumnya digunakan melakukan otentifikasi aplikasi seperti telah dijelaskan pada bagian sebelumnya dan otentifikasi pengguna perangkat. Modul ini mendukung sertifikasi pada Wireless Transport Layer Security (WTLS) dan sertifikasi X.509 berdasarkan profile sertifikat PKIX (RFC 2459).

Modul manajemen sertifikasi ini memberikan layanan berupa:

- Menyimpan dan mengambil sertifikat menggunakan framework kriptografi
- Pemberian status kepercayaan berdasarkan sertifikat yang dimiliki aplikasi untuk melakukan operasi khusus
- Manajemen sertifikasi yang untuk melakukan validasi dan konstruksi aplikasi
- Menverifikasi sertifikat itu sendiri
- Pembatalan status validasi dari sertifikat menggunakan *Online Certificate Status Protocol (OCSP)*. OCSP merupakan protocol yang memungkinkan suatu aplikasi untuk menentukan penarikan status validitas terhadap suatu sertifikat (RFC 2560)

Manajemen sertifikasi terdiri dari lima komponen utama. Ilustrasi di bawah ini menggambarkan relasi antar komponen



Gambar 7: Komponen utama modul manajemen sertifikasi

Bab 4 Potensi Ancaman Terhadap Sistem Keamanan

Ancaman bagi sistem operasi Symbian OS secara umum dapat di klasifikasikan dalam beberapa kelompok. Berikut beberapa penyebab gangguan dalam sistem keamanan:

- tindakan dengan itekad bermusuhan. Penyerang berniat untuk merusak sistem
- tindakan memperlengah secara administratif manajemen telepon
- kesalahan pengguna
- kesalahan teknis yang disebabkan kerusakan data

Adapun dilihat dari sisi aplikasi, ancaman dari aplikasi yang sifatnya merusak dan perlu diperhatikan oleh pihak pengembang dapat dikategorikan berdasarkan bagaimana aplikasi tersebut menyebar yaitu:

- Aplikasi yang memerlukan host untuk menyebar. Aplikasi yang masuk kategori ini dapat berupa backdoors, logical bombs, Trojan dan virus. Aplikasi ini melakukan pemrograman terhadap aplikasi yang ingin dipengaruhi/disalahgunakan atas dasar motivasi tertentu.
- Aplikasi menyebar tanpa memerlukan host. Yang termasuk dalam kategori ini adalah aplikasi yang berbentuk bakteri dan worm. Prinsipnya aplikasi menghabiskan sumber daya yang ada di telepon ataupun network dengan menduplikasi diri.

Namun bentuk aplikasi yang diklasifikasikan seperti di atas sering kali tidak berdiri sendiri. Suatu aplikasi memerlukan bentuk aplikasi lain untuk merealisasikan tujuannya.

Contoh aplikasi yang memberikan efek negatif yang bekerja pada sistem operasi ini antara lain: *Cabir* yang merupakan virus yang menyebar melalui bluetooth, *Drever.A* merupakan Trojan yang bertujuan menon-aktifkan proses startup antivirus seperti Simwork dan Kaspersky, *Locknut.B* yang merupakan Trojan yang berpura-pura sebagai patch bagi Symbian yang digunakan oleh telepon bergerak seri 60. Aplikasi ini menyebabkan semua aplikasi lain tidak dapat dijalankan. *Frontal.A* merupakan bentuk Trojan yang menginstal suatu data yang rusak sehingga ketika sistem menjadi berhenti ketika proses reboot.

Bab 5 Penutup

Sistem operasi Symbian OS dalam perkembangannya memberikan kemudahan dan pilihan bagi pihak pengembang untuk membangun aplikasi untuk bekerja pada berbagai peralatan telepon bergerak menuju bentuk smartpone dengan fitur yang semakin memudahkan pengguna untuk melakukan berbagai aktifitas dengannya. Seiring itu pula muncul tantangan baru pada sistem operasi ini untuk melakukan pengamanan. Strategi utama untuk memperkuat pengamanan ini adalah dengan menerapkan proses kontrol yang melibatkan pengguna sendiri dan pihak Symbian OS dalam memberikan hak akses terhadap semua aplikasi yang ingin diinstal pada sistem operasi ini. Proses ini terutama dengan diterapkan dengan adanya model keamanan platform aplikasi yang berupa model Capability-based mulai pada versi 9.

Daftar Pustaka

- [1]. Shackman, Mark "*Platform Security-a Technical Overview*", www.symbian.com
- [2]. Martin de Jode, Colin Trufus "*Symbian OS System Definition*", www.symbian.com
- [3]. <http://www.symbianwatch.com/category/security>
- [4]. http://en.wikipedia.org/wiki/Symbian_OS, "*Symbian OS*"
- [5]. http://forum.nokia.com/main/newsletter/2005/may_17_2005.html

- [6]. http://developer.symbian.com/main/downloads/papers/SymbOS_cat/SymbianOS_cat.html
- [7]. [http://en.wikipedia.org/wiki/Kernel_\(computer\)](http://en.wikipedia.org/wiki/Kernel_(computer)), "Kernel (Computer Science)"
- [8]. Pagonis, John,
http://developer.symbian.com/main/downloads/papers/newipc/new_ipc_mechanisms_for_symbian_os.pdf, "New IPC Mechanism for Symbian OS", Symbian Ltd., 2005
- [9]. http://www.webopedia.com/TERM/I/interprocess_communication_IPC.html, "Interprocess Communication (IPC)"
- [10]. <http://www.rfc-archive.org/getrfc.php?rfc=2560> "OSCP"
- [11]. http://www.symbian.com/developer/techlib/v70sdocs/doc_source/devguides/cpp/Base/InterProcessCommunication/ClientServerOverview.guide.html "Client/Server Overview"
- [12]. http://www.symbian.com/developer/techlib/v9.1docs/doc_source/guide/N1010A/Architecture/index.html#SAOverview, "Symbian OS Security Architecture"