Studi dan Implementasi Sistem Anti Pembajakan Perangkat Lunak Menggunakan Pengamanan Data

Aqsath Rasyid Naradhipa – NIM : 13506006
Program Studi Teknik Informatika
Institut Teknologi Bandung
Jl. Ganeca 10, Bandung
E-mail : aqsath@gmail.com

Abstrak

Dewasa ini, intensitas pembajakan perangkat lunak relatif tinggi di Indonesia. Hal ini bisa menyebabkan industri kreatif di Indonesia lambat laun menjadi mati karena daya jual dari perangkat lunak semakin kecil. Beberapa metode anti pembajakan telah dicoba mulai dari nomor serial, penanda waktu, hingga program yang mempunya fungsi validasi tertentu. Namun, metode – metode tersebut masih kurang mampu untuk menahan laju pembajakan perangkat lunak.

Menilik metode anti pembajakan yang sudah ada, dapat disimpulkan bahwa yang selama ini dilakukan adalah bagaimana cara untuk mengamankan program. Berdasarkan hal tersebut, solusi yang ditawarkan adalah tidak lagi mengamankan program namun mengamankan data yang akan dipakai dalam program tersebut seperti gambar, suara, dan lain – lain. Dengan menggunakan metode pengamanan data, dimungkinkan perangkat lunak yang sudah diinstalasi di sebuah komputer belum tentu dapat digunakan selama data yang akan dipakai di program tersebut masih dalam keadaan terenkripsi.

Kata Kunci : Sistem Anti Pembajakan, Pengamanan Data

1. Pendahuluan

Dewasa ini, perilaku pembajakan marak terjadi di seluruh dunia. Antara lain, sebagaimana yang terjadi di Asia Pasifik dari jumlah perangkat lunak yang beredar 53% diantaranya adalah bajakan dan menimbulkan kerugian sebesar $7,5 milyar [1]. Di Indonesia sendiri, dari jumlah perangkat lunak yang beredar 85% diantaranya adalah bajakan dan menimbulkan kerugian sebesar $544 juta [BSA208]. Hal ini membuat industri kreatif di Indonesia lambat laun menjadi mati karena kita terbiasa memakai barang bajakan dan cenderung malas untuk menciptakan sesuatu yang baru dan kreatif. Pendapat orang yang mengatakan bahwa pembajakan membuat masyarakat Indonesia menjadi lebih pintar karena dapat menikmati perangkat lunak komersil dengan harga yang lebih terjangkau tidak selamanya benar. Ketika kita melihat efek jangka panjangnya, pembajakan bisa menjadikan masyarakat Indonesia semakin bodoh dan malas untuk berkreasi [2].

Perilaku pembajakan yang marak terjadi disebabkan oleh lemahnya metode proteksi yang digunakan. Beberapa metode proteksi perangkat lunak yang pernah dan / atau saat ini adalah metode proteksi dengan nomor serial, metode proteksi dengan penanda waktu, dan metode proteksi dengan fungsi validasi. Menilik perkembangan metode sistem anti

pembajakan yang sudah ada, sistem yang ada sekarang berfokus pada cara melindungi program ataupun *source code* program [4]. Namun, metode – metode tersebut sudah terbukti tidak berhasil. Hal ini terlihat banyaknya metode yang dapat melumpuhkan metode – metode proteksi tersebut [3]. Oleh karena itu, makalah ini akan menggunakan pendekatan yang berbeda, yaitu dengan cara melindungi data yang akan digunakan. Dengan menggunakan pendekatan ini, bukan program yang akan dilindungi namun data (gambar, tulisan, atau *library*) yang akan digunakan program tersebut sehingga program yang terpasang di komputer pengguna tidak dapat digunakan selama data yang akan dipakai program tersebut masih dalam keadaan terenkripsi.

2. Perkembangan Sistem Anti Pembajakan Perangkat Lunak

Pada tahun 1990-an, perangkat lunak disimpan di sebuah media penyimpanan khusus yang relatif mahal dan sulit dibajak, contohnya cartridge yang dipakai untuk memainkan console Nintendo. Media penyimpanan tersebut sulit untuk didapatkan dan kalau berhasil mendapatkannya pun harganya sangat mahal. Hal ini membuat pembajak enggan melakukan pembajakan. Namun, hal ini juga membuat resah para pengembang karena biaya produksi perangkat lunak habis di media penyimpanannya.

Seiring dengan berkembangnya teknologi, muncullah 5.25-inch *floppy disk* atau yang biasa dikenal dengan disket.Bahan bakunya yang murah dan universal membuat pengembang beralih ke disket. Hal ini membuat pembajak tidak ragu lagi untuk membajak perangkat lunak. Dengan media penyimpanan yang lebih murah dan mudah untuk didapatkan, pembajak dengan mudahnya menyalin isi dari disket asli yang didistribusikan pengembang ke disket kosong. Dengan cara ini, pembajak dapat menjualnya dengan harga yang lebih murah dari pengembang. Proteksi yang ada dilakukan dengan cara proteksi offline. Perangkat lunak didistribusikan bersama buku manual atau dokumentasi. Di tengah perangkat lunak dijalankan akan ditanyakan bagian dari dokumentasi tersebut, misalnya kata kedua pada baris ke 16 di halaman 24. Metode ini berkembang sampai pengembang menggunakan kertas ber-*watermark* untuk mencetak dokumentasinya.

Teknologi media penyimpanan terus berkembang, sampai munculnya teknologi *compact disc* (CD). CD teknologi yang lebih menjanjikan dibandingkan dengan disket karena selain dari kapasitasnya yang lebih besar, biaya produksi CD lebih murah daripada disket. Proteksi pada CD dilakukan dengan cara menambahkan beberapa *byte* yang sengaja dirusak atau *bad sector* sehingga dengan adanya byte tersebut akan terjadi kesalahan ketika proses penyalinan CD karena adanya *byte* yang tidak dapat dibaca. Dari beberapa metode proteksi yang ada nampaknya masih belum cukup untuk menangkal pembajakan. Beberapa metode proteksi yang dipaparkan sebelumnya masih berpaku pada media penyimpanannya. Seiring dengan berkembangnya teknologi informasi dan komunikasi, perangkat lunak tidak lagi didistribusikan melalui media penyimpanan konvensional lagi, namun bisa dengan cara di-download, e-mail, atau langsung disalin antar komputer menggunakan *USB flash disk*. Hal itu membuat munculnya paradigma baru cara memproteksi perangkat lunak yaitu tidak lagi bagaimana cara melindungi agar pembajak tidak dapat menyalin data namun bagaimana cara melindungi perangkat lunak walaupun data telah tersalin. Dari dasar tersebut muncullah era baru dari sistem anti pembajakan perangkat lunak [5].

Metode yang pertama adalah metode proteksi menggunakan nomor serial. Metode ini akan mencocokkan nomor serial yang diberikan oleh pengguna dengan nomor serial yang ada pada perangkat lunak, ketika nomor serial tersebut cocok maka perangkat lunak dapat digunakan. Metode yang kedua adalah metode proteksi menggunakan penanda waktu. Metode ini akan memberikan waktu kepada pengguna untuk menggunakan perangkat lunak secara bebas, ketika sudah sampai batas waktu penggunaan dan perangkat lunak masih teridentifikasi ilegal maka perangkat lunak tersebut akan menonaktifkan dirinya sendiri sehingga

pengguna tidak dapat menggunakan perangkat lunak tersebut. Metode yang ketiga adalah pemakaian fungsi validasi. Metode ini menggunakan fungsi validasi tertentu yang digunakan untuk menentukan keabsahan dari perangkat lunak yang digunakan, fungsi validasi ini dapat menggunakan cara yang bermacam – macam contohnya *digital signature*, aktivasi, registrasi, dan lain – lain. Namun, dari beberapa metode yang sudah disebutkan masih belum dapat menangkal pembajakan. Hal ini terlihat dari masih maraknya pembajakan yang ada.

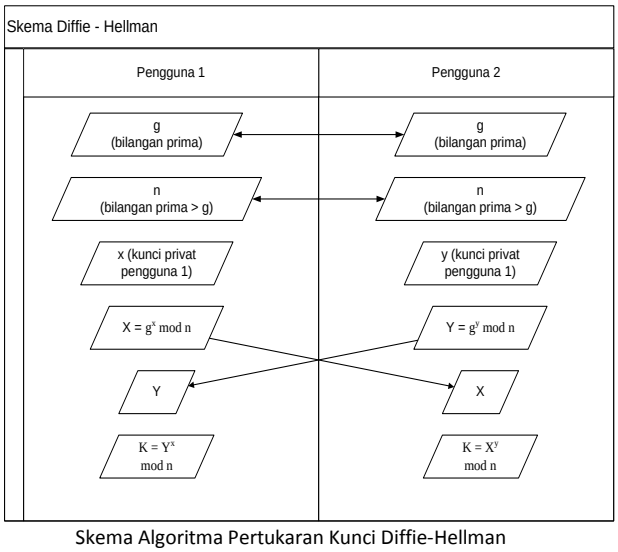
3. Kriptografi

Kriptografi adalah ilmu untuk menjaga kerahasiaan informasi, dengan adanya kriptografi kita dapat mengamankan informasi yang mau kita rahasiakan dan hanya dapat dibaca atau digunakan oleh orang yang kita kehendaki. Beberapa layanan keamanan yang dicakup oleh kriptografi adalah kerahasiaan, integritas data, otentikasi, dan nirpenyangkalan.

Dari kunci yang digunakan untuk mengenkripsi dan mendekripsi, kriptografi dibedakan menjadi dua tipe yaitu kriptografi kunci publik dan kunci privat. Jika menggunakan skema kriptografi kunci privat, kunci yang digunakan untuk mendekripsi data harus sama dengan kunci yang digunakan untuk mengenkripsi data, sedangkan jika menggunakan skema kriptografi kunci publik, kunci yang digunakan untuk mendekripsi data tidak harus sama dengan kunci yang digunakan untuk mengenkripsi data. Dalam sistem anti pembajakan perangkat lunak menggunaakn metode pengamanan data, skema kriptografi kunci privat lebih cocok digunakan karena skema kunci publik mengharuskan pengguna dan pengembang berhubungan lebih dulu sebelum data akan dienkripsi untuk melakukan perhitungan kunci sedangkan dalam keadaan nyata komunikasi antara pengguna dan pengembang sebelum data dienkripsi tidak mungkin dilakukan.

Salah satu algoritma kriptografi kunci privat adalah AES (Advanced Encryption Standard). AES merupakan standard enkripsi hasil sayembara yang diselenggarakan oleh NIST. Sayembara ini dilakukan karena standard sebelumnya, Data Encryption Standard, sudah tidak aman lagi. Dalam sayembara in ada beberapa persyaratan yang harus dipenuhi untuk mengikuti sayembara ini. Persyaratan itu adalah algoritma yang ditawarkan termasuk ke dalam algoritma kelompok simetri berbasis *cipher* blok, seluruh rancangan algoritma harus publik (tidak dirahasiakan), panjang kunci fleksibel (128, 192, dan 256 bit), Ukuran blok yang dienkripsi adalah 128 bit, dan algoritma dapat diimplementasikan baik sebagai perangkat lunak maupun perangkat keras. Secara garis besar, algoritma AES hanya memiliki empat proses yaitu SubBytes, ShiftRows, MixColumns, dan AddRoundKey. Dari keempat proses tersebut akan diulang – ulang sebanyak N kali. Pada dasarnya, sistem anti pembajakan perangkat lunak menggunakan metode pengamanan data tidak membutuhkan algoritma khusus untuk implementasinya namun AES akan dijadikan contoh algoritma [9].

Salah satu kelemahan dari kriptografi kunci privat adalah saluran yang digunakan untuk mengirimkan kunci harus aman. Hal ini sangat sulit dilakukan ckarena pihak pengguna tidak dapat kita percaya sepenuhnya karena memiliki kemungkinan bahwa pengguna yang akan berkomunikasi dengan kita adalah seorang pembajak. Hal ini dapat diselesaikan dengan algoritma pertukaran kunci Diffie-Hellman.



Skema Algoritma Pertukaran Kunci Diffie-Hellman

Secara umum, skema Diffie-Hellman digambarkan dengan gambar di atas. Kunci privat akan

didefinsikan oleh pengembang perangkat lunak dan pengguna perangkat lunak lalu menggunakan perhitungan yang telah didefinisikan akan diperoleh kunci yang sama yaitu K besar. Kunci K ini yang akan digunakan untuk proses yang akan dilakukan oleh pengguna dan pengembang [6].

4. Sistem Anti Pembajakan Perangkat Lunak Menggunakan Metode Pengamanan Data

Sistem anti pembajakan perangkat lunak yang dirancang memperhatikan dua aspek yaitu kecepatan dan keamanan. Pada dasarnya, kedua aspek itu bergantung pada algoritma enkripsi yang digunakan sistem anti pembajakan perangkat lunak, namun di sistem ini kecepatan dan keamanan tersebut akan dioptimalkan. Hal yang dapat dilakukan untuk mengoptimalkan kedua aspek itu adalah perancangan skema enkripsi data dan skema pertukaran kunci yang digunakan untuk mendekripsi data yang digunakan perangkat lunak.

Skema enkripsi data yang digunakan perangkat lunak dengan cara membaca isi dan struktur file resource yang digunakan perangkat lunak. Dari data yang diperoleh dari hasil pembacaan itu resource tersebut akan digabungkan menjadi satu file dengan struktur

```
<path_file1><separator><byte_file1><separator><path_file2><separator><byte_file2><separator> … <path_fileN><separator><byte_fileN>
```

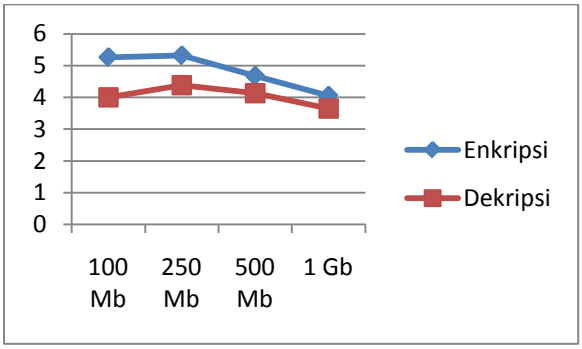
Hal ini dilakukan agar mempercepat proses enkripsi. Dari analisis yang dilakukan mengenkripsi banyak file akan memperlambat performansi komputer, dengan menggabungkannya menjadi sebuah file yang besar akan mempercepat performansi komputer. Setelah resource digabungkan menjadi satu file, file tersebut dienkripsi menggunakan algoritma enkripsi dan kunci yang telah ditentukan sebelumnya.

Algoritma pertukaran kunci Diffie-Hellman akan dimanfaatkan dalam pendistribusian kunci dari pengembang perangkat lunak ke pengguna perangkat lunak. Skema pendistribusian kunci yang dirancang adalah pertama sistem akan mengambil nomor serial *Harddisk Drive* (HDD) sebagai kunci privat (y kecil) dari pengguna. Dengan perhitungan yang telah didefinisikan di algoritma Diffie-Hellman maka kunci privat akan menjadi kunci publik (y besar) dari pengguna. Kunci publik itu yang akan dikirimkan ke pengembang perangkat lunak. Dengan perhitungan yang telah didefinisikan di algoritma pertukaran kunci Diffie-Hellman maka akan dihasilkan kunci publik pengembang (x besar) dan kunci bersama (K). Kunci yang digunakan untuk mengenkripsi resource akan dienkripsi dengan kunci K. Setelah kunci dienkripsi, kunci ini akan didistribusikan ke pengguna dengan kunci publik pengembang. Ketika pengguna akan mendekripsi resource yang digunakan perangkat lunak, maka kunci itu harus didekripsi terlebih dahulu menggunakan kunci bersama yang didapatkan dari perhitungan yang dihasilkan algoritma pertukaran kunci Diffie-Hellman. Dengan cara ini pengguna tidak mengetahui kunci asli yang digunakan untuk mengenkripsi resource.

Dari pengujian yang dilakukan terhadap sistem yang diimplementasikan berdasarkan analisis skema yang telah dibuat didapatkan hasil sebagai berikut

Ukuran Resource	Waktu Proses Enkripsi	Waktu Proses Dekripsi
1 Gb	4 Menit 7 Detik	4 Menit 34 Detik
500 Mb	1 Menit 47 Detik	2 Menit 1 Detik
250 Mb	47 Detik	57 Detik
100 Mb	19 Detik	25 Detik

Hasil Pengujian Sistem Anti Pembajakan



Grafik Pengujian Sistem Anti Pembajakan

Grafik menunjukkan ukuran yang dienkripsi per detik dalam satuan Mb/s. Dapat dilihat bahwa proses enkripsi memakan waktu yang lebih sedikit dibandingkan proses enkripsi. Namun, dapat dilihat

juga ada keanehan dalam grafik tersebut dimana di ukuran 250 Mb mengalami penaikan performansi namun di 500 Mb dan seterusnya mengalami penurunan performansi. Dari hasil analisis lebih lanjut, didapatkan bahwa proses yang memakan waktu lebih dari 1 menit (enkripsi dan dekripsi diatas 250 Mb) akan dialihkan ke proses *input / output*. Sedangkan untuk proses di bawah 1 menit akan tetap memakai DMA (Direct Memory Access).

Selain itu, sistem anti pembajakan perangkat lunak juga dapat mengenkripsi berbagai jenis macam data seperti *file* dokumen (doc, docx, xls, xlsx, ppt, pptx, dan txt), *file* gambar (jpeg, png, bmp, dan tiff), *file* musik (wma, mp3, dan wav), *file video* (wmv, avi, mpeg, dan mkv), *file library* (dll, asm, ocx), dan *file text* biasa seperti txt.

Hasil pengujian yang dilakukan hanya terbatas kecepatan dari sistem anti pembajakan perangkat lunak. Hal ini dikarenakan untuk melakukan pengujian keamanan dan membuktikan bahwa sistem ini aman dibutuhkan waktu yang sangat lama. Selain itu keamanan dari sistem ini juga bergantung dari beberapa faktor, faktor yang utama adalah algoritma enkripsi yang digunakan untuk mengenkripsi *resource* yang digunakan perangkat lunak.

5. Kesimpulan

Metode pengamanan data dapat dijadikan solusi untuk permasalahan pembajakan perangkat lunak. Hal ini didasari dari pendekatan yang berbeda dari solusi – solusi yang sudah ada sebelumnya. Diharapkan dengan dikembangkannya metode pengamanan data dapat menjadi era baru dari sistem anti pembajakan perangkat lunak.

Performansi dari sistem anti pembajakan perangkat lunak dipengaruhi beberapa faktor. Faktor yang pertama adalah algoritma yang digunakan untuk mengenkripsi resource, faktor kedua adalah algoritma yang digunakan untuk mengenkripsi kunci dalam proses pendistribusian kunci ke pengguna perangkat lunak.

Performansi dari sistem anti pembajakan perangkat lunak pengamanan data hanya meliputi kecepatan dan keamanan.

Daftar Referensi

[1] BSA – Major Study Finds 53 Percents of Software Use in Asia Pacific Region is Pirated
http://w3.bsa.org/asia-eng/press/newsreleases/Major-Study-Finds-53-Percent-of-Software-in-Use-in-Asia-Pacific-Region-is-Pirated.cfm
Tanggal Akses : 24 Agustus 2009

[2] Tingkat Pembajakan Piranti Lunak di Indonesia Naik Menjadi 85% di Tahun 2008
http://global.bsa.org/globalpiracy2008/pr/pr_indonesia.pdf
Tanggal Akses : 30 Agustus 2009

[3] Software Cracking
http://it.toolbox.com/wiki/index.php/Software_cracking
Tanggal Akses : 25 Agustus 2009

[4] Cracking : The Removal of Software Copy Protection
http://www.mindspring.com/~win32ch/Crackit.htm
Tanggal Akses : 28 Agustus 2009

[5] A History of Copy Protection
http://www.edge-online.com/features/a-history-copy-protection
Tanggal Akes : 19 November 2009

[6] Munir, Rinaldi. Kriptografi, Institut Teknologi Bandung, 2006.

[7] Stalling, W., *Cryptography and Network Security, Principal and Practice 2nd Edition,* Pearson Education, Inc., 1998.

[8] Menezes, Alfred J., Paul C van Oorschot, dan Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[9] Advanced Encryption Standard
http://en.wikipedia.org/wiki/Advanced_Encryption_Standard
Tanggal Akes : 21 Desember 2009.

Pernyataan

Dengan ini saya menyatakan bahwa tidak ada plagiasi pada makalah UAS Kriptografi ini.

Bandung, 17 Mei 2010

Aqsath Rasyid Naradhipa