Program StudiTeknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Tugas III IF5054 Kriptografi

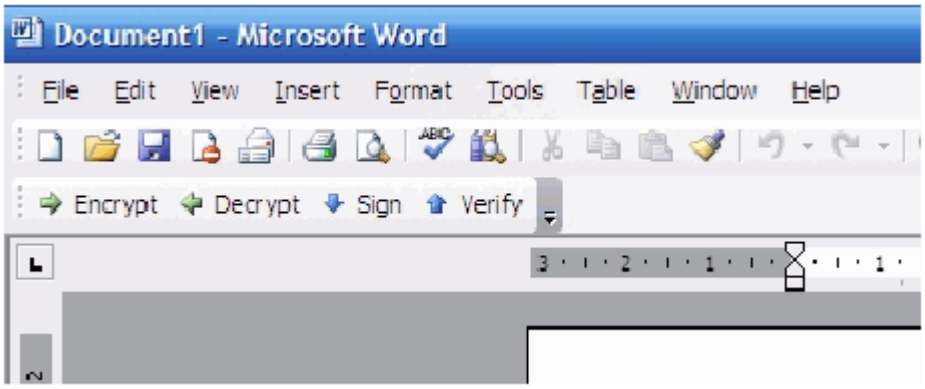
Program *Add-in* Tanda-tangan Digital pada Aplikasi Pengolah Kata dengan Menggunakan Algoritma RSA dan Fungsi *hash* MD5

Batas pengumpulan : 30 November 2007
Tempat pengumpulan : Lab IRK
Arsip pengumpulan :
- disket/cd berisi program, arsip *readme.txt*, laporan, arsip contoh, arsip parameter dan kunci.
- kertas A4 untuk laporan (*print 2up*)

Deskripsi tugas :

Tanda-tangan digital dapat digunakan untuk otentikasi data digital, seperti pesan yang dikirim melalui saluran komunikasi dan dokumen elektronis yang disimpan dalam komputer.

Pada tugas ke-3 ini, anda diminta mengimplementasikan algoritma *RSA* untuk memberi tanda-tangan digital pada dokumen (*file*) elektronis, dan algoritma MD5 untuk membangkitkan nilai *hash* dari dokumen. Dalam hal ini, anda sebagai pemilik dokumen mempunyai sepasang kunci, yaitu kunci publik dan kunci privat. Program tanda-tangan digital di-*add-in* (*plug-in*) ke dalam aplikasi pengolah kata (misalnya *NotePad*, *WordPat*, *Microsoft Word*, dll) sehingga setiap kali pengguna membuat dokumen dan ingin menandatanganinya, maka ia cukup meng-klik tombol *sign* yang ada di *speed icon* aplikasi. Begitu juga jika ia ingin memverifikasi tanda-tangan, maka ia cukup meng-klik tombol *verify*.
Contoh yang sudah pernah diekmbangkan oleh mahasiswa IF (Agus Hilman Majid, IF 2000) adalah program *add-in* tanda-tangan digital pada aplikasi *Microsoft Word* dengan algoritma ElGamal:



Tanda-tangan digital dapat dilekatkan (*embedded*) di awal atau di akhir dokumen, tetapi, pada tugas ini tanda-tangan digital dilekatkan di akhir dokumen. Tanda-tangan digital selanjutnya digunakan untuk membuktikan keaslian isi dokumen dan keaslian pemilik dokumen. Dokumen harus dapat diekstraksi kembali dari arsip yang sudah diberi tanda-

tangan digital sehingga dokumen dapat dibuka dan diproses oleh program aplikasi yang bersesuaian. Begitu juga tanda-tangan digital harus dapat diekstraksi dari dokumen.

Tanda tangan digital bergantung pada isi dokumen dan kunci. Tanda-tangan digital direpresentasikan sebagai karakter-karakter heksadesimal dan ditaruh pada awal dokumen. Untuk membedakan tanda-tangan digital dengan isi dokumen, maka tanda-tangan digital diawali dan diakhiri dengan *tag* `<ds>` dan `</ds>`, atau penandaan dengan cara lain (diserahkan kepada anda)
Contoh: `<ds>4EFA7B223CF901BAA58B991DEE5B7A</ds>`.

Berhubung algoritma *RSA* menggunakan parameter bilangan bulat yang panjang (besar), maka program anda harus mampu menggunakan bilangan yang besar dengan membuat tipe data khusus untuk bilangan bulat besar dan primitif-primitif operasi aritmetikanya. And adapat membuat sendiri tipe *BigInteger* (ldianjurkan) atau menggunakan fungsi-fungsi *BigInteger* yang sudah disediakan oleh kakas (seperti *.NET* atau *Java*) atau diambil dari situs-situs internet. Situs web ini misalnya,

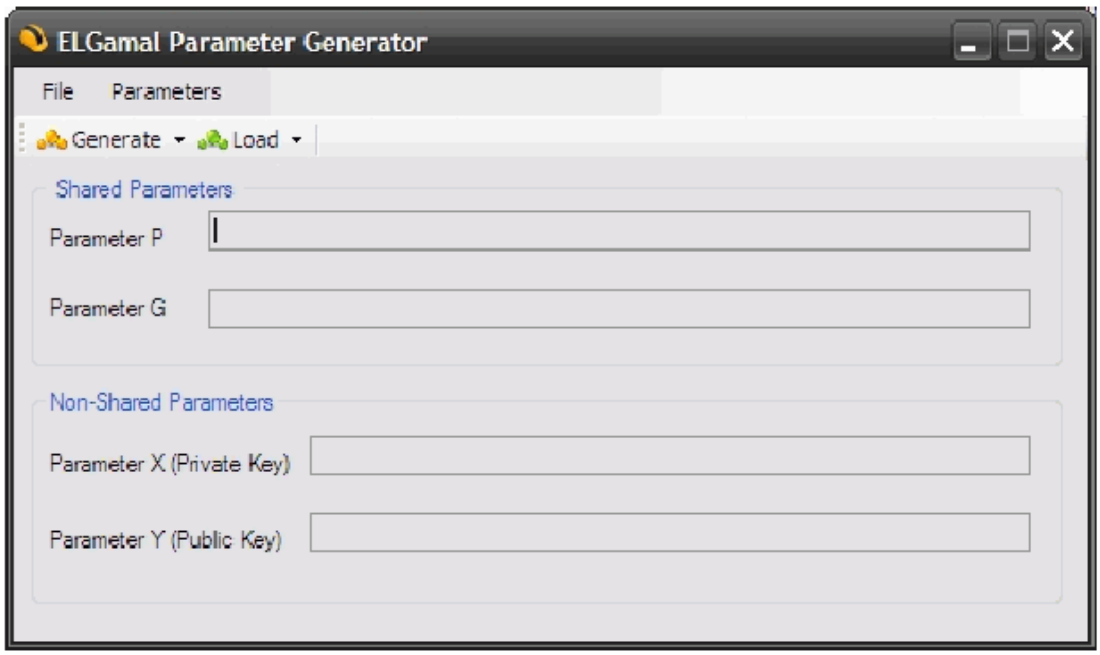
Bouncy Castle Cryptographic C# API (http://www.bouncycastle.org).

menyediakan pustaka eksternal (*dll*) khusus C# dalam bentuk *API*.

Spesifikasi program:

Yang anda buat adalah:

1. Pustaka RSA dan MD5 (*RSA-MD5Libary*) adalah pustaka aplikasi yang berisi operasi-operasi pembangkitan tanda-tangan digital dengan algoritma RSA (enkripsi, dekripsi) dan algoritma *hash MD5*. Pustaka RSA dan MD5 bisa juga dibuat terpisah.
2. Aplikasi *desktop KeyGenerator*, adalah aplikasi yang bertujuan untuk membangkitkan parameter-parameter di dalam algoritam RSA (bilangan prima *p* dan *q*, kunci publik, kunci privat).
   Contoh program *KeyGenerator* ElGamal yang dikembangkan oleh Agus Hilman Majid:



3. Aplikasi *Word Add-in*, berupa *add-in* di program pengolah kata teks (boleh dipilih: *NotePad*, *WordPad*, *UltraEdit*, dll). Sementara ini baru untuk editor teks karena strukturnya sederhana, tetapi tetap diperbolehkan menambahkannya pada aplikasi pengolah kata yang lebih kompleks seperti *Microsoft Word*.
   Ikon menu program *add-in* minimal dua: penandatanganan dan verifikasi. Silakan menambahkan fitur lain seperti enkripsi dan dekripsi.

   Berikut ini dikutip dari laporan Tugas Akhir Agus Hilman Majid:

   Untuk implementasi *add-in*, anda dapat menggunakan *Visual Studio Tools for Office* (*VISTO*). *VISTO* adalah *framework* aplikasi berbasis *.NET* untuk membangun aplikasi di atas *Office*, dalam hal ini menjadi add-i aplikasi di dalam program pengolah kata. Untuk bisa menggunakan *VISTO* hasrus diinstalasi lebih dahulu kemudian akan ada tipe *project* baru pada *Visual Studio* yaitu jenis *Office Project*.

   Untuk ElGamal *Add-in* ini, yang termasuk kategori *Word Add-in Project*, *VSTO* meng-*generate* kelas `ThisAddin` yang berisi dua *method* utama, yaitu:

   1. `ThisAddIn_Startup`

      Method yang dijalankan ketika aplikasi *Microsoft Word* dibuka.

   2. `ThisAddIn_Shutdown`

      Method yang dijalankan ketika aplikasi *Microsoft Word* ditutup.

   Berikut ini adalah contoh isi kelas ThisAddin *default* yang di-*generate* oleh *VSTO* :

```
using System;
using System.Windows.Forms;
using Microsoft.VisualStudio.Tools.Applications.Runtime;
using Word = Microsoft.Office.Interop.Word;
using Office = Microsoft.Office.Core;

namespace WordAddIn2
{
    public partial class ThisAddIn
    {
        private void ThisAddIn_Startup(object sender, System.EventArgs e)
        {
         // isi disini kode yang akan dijalankan ketika MS Word dibuka
        }

        private void ThisAddIn_Shutdown(object sender, System.EventArgs e)
        {
         // isi disini kode yang akan dijalankan ketika MS Word ditutup
        }

        #region VSTO generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InternalStartup()
        {
            this.Startup += new System.EventHandler(ThisAddIn_Startup);
            this.Shutdown += new System.EventHandler(ThisAddIn_Shutdown);
        }

        #endregion
    }
}
```

Untuk implementasi ElGamal Word Add-in ini, pada *method* `ThisAddIn_Startup` dilakukan pembuatan *toolbar* baru yang 'ditempelkan' pada *toolbar Microsoft Word* dengan menggunakan *method* `CreateToolBar`. *Method* `CreateToolBar` ini juga melakukan *assign* terhadap masing-masing *button* yang ada pada *toolbar* tersebut untuk melakukan *method* sesuai dengan yang telah ditentukan. Untuk jelasnya bisa dilihat di lampiran B.

Lain-lain

1. Program diberi nama yang singkat, menarik, dan memiliki makna.
2. Program harus mengandung komentar yang jelas.
3. Sertakan juga program setup untuk memg-instalasi dan me-*remove* progam *add-in* ke dalam aplikasi pengolah kata.
3. Lampirkan di dalam disket program anda arsip contoh dan arsip parameter & kunci.
4. Program MD5 sangat dianjurkan dibuat sendiri (lebih memberi tantangan). Jika anda mengambil kode program MD5 dari internet, anda harus menyebutkan *URL* yang mengandung program MD5 tersebut.

Isi laporan :

1. Deskripsi masalah.
2. Dasar teori.
3. Strategi penyelesaian masalah (lingkungan implementasi dan trik khusus).
4. Struktur data dan spesifikasi subrutin.
5. Pengujian dan analisis hasil. Pengujian menggunakaan arsip contoh yang disertakan di dalam teks.
   Pengujian meliputi otentikasi dengan kasus-kasus berikut:
   - karakter di dalam teks diubah (dihapus, ditambah)
   - karakter di dalam tanda-tangan digital diubah
   - kunci privat yang digunakan tidak berpadanan dengan pasangan kunci publiknya.
   - tanda-tangan digital dihapus dari dokumen
6. Lampiran yang berisi:
   - antarmuka program
   - contoh arsip masukan
   - contoh arsip keluaran yang sudah diberi tanda-tangan digital.
   - contoh nilai-nilai paramater *RSA* yang digunakan
7. Kesimpulan dan saran.