Program StudiTeknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Tugas III IF3058 Kriptografi

Implementasi Program Tanda-tangan Digital dengan Menggunakan Algoritma RSA dan Fungsi *hash* MD5

Batas pengumpulan : 8 Mei 2009
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*)

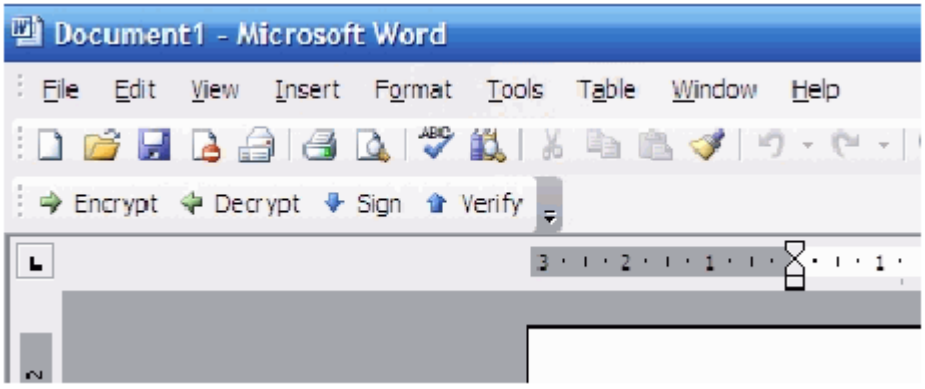
Deskripsi tugas :

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:

1. Membuat aplikasi desktop yang mengimplementasikan algoritma *RSA* + MD5 untuk memberi tanda-tangan digital pada dokumen (*file*) elektronis. Dalam hal ini, anda sebagai pemilik dokumen mempunyai sepasang kunci, yaitu kunci publik dan kunci privat.
2. Mengimplementasikan program tanda-tangan digital sebagai program *add-in* (*plug-in*) pada salah satu dari dalam aplikasi berikut:
    a) 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*.
    b) aplikasi *e-mail* seperti *Microsoft Outlook* atau yang lainnya sehingga setiap e-maild apat dibubuhi tanda-tangan digital.

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*) pada dokumen pesan atau disimpan di dalam dokumen terpisah, tetapi pada tuags ini tanda-tangan digital disatukan di dalam dokumen pesan. Tanda-tangan digital dapat diletakkan 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,

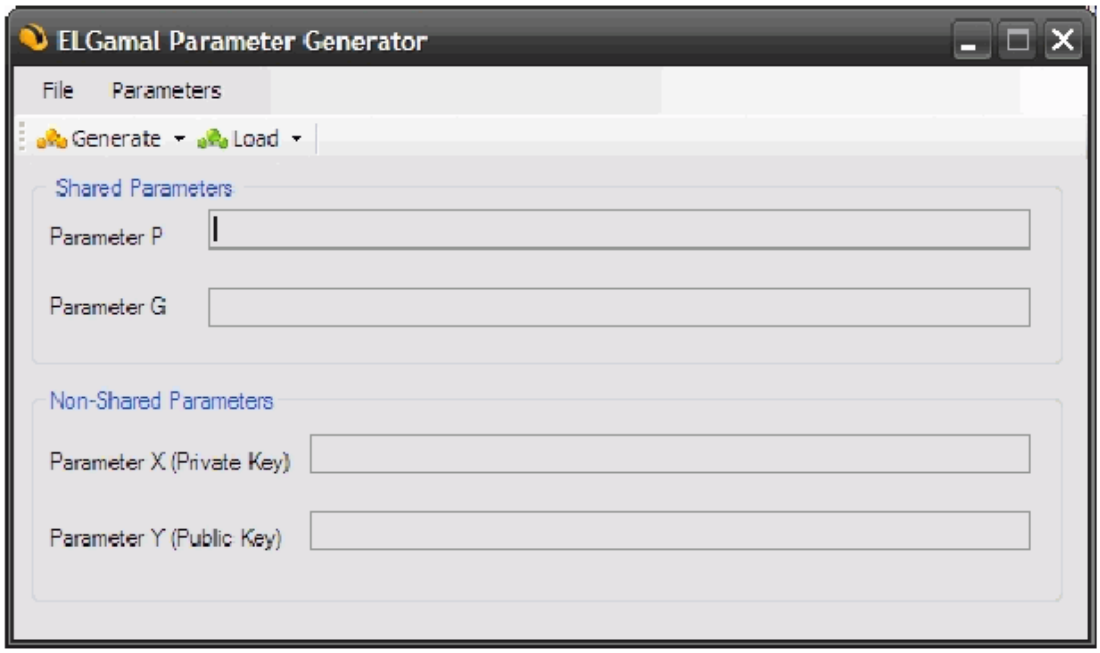
Bouncy Castle Cryptographic C# API (http://www.bouncycastle.org).

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

Spesifikasi program:

Yang anda buat adalah:

1. Aplikasi desktop tanda-tangan digital dengan algoritma RSA dan MD5 (*RSA-MD5Libary*).
2. Aplikasi *desktop KeyGenerator*, adalah aplikasi yang bertujuan untuk membangkitkan parameter-parameter di dalam algoritma RSA (bilangan prima *p* dan *q*, kunci publik, kunci privat).
   Contoh program *KeyGenerator* ElGamal yang dikembangkan oleh Agus Hilman Majid:



3. Program *add-in* tanda-tangan digital pada aplikasi *word processor* atau aplikasi *e-mail*. Program pengolah kata berbasis teks (boleh dipilih: *NotePad*, *WordPad*, *UltraEdit*, dll). Sementara ini baru untuk editor teks karena strukturnya sederhana, tetapi diperbolehkan pada aplikasi pengolah kata yang lebih kompleks seperti *Microsoft Word*.
   Ikon menu program *add-in* minimal dua: penandatanganan dan verifikasi.

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.