Penerapan Steganografi dan Vigenere pada File Terkompresi Berformat Zip

Sandy Gunawan Tanuwijaya/13510025
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13510025@itb.ac.id

***Abstract*— Steganografi merupakan sebuah ilmu dan seni menyembunyikan suatu informasi dengan cara menyisipkan informasi tersebut di dalam informasi/pesan lainnya. Pesan/informasi yang disembunyikan dapat berupa teks, audio, gambar maupun video, sedangkan media tempat penyembunyian dapat berupa gambar, audio, video, aplikasi/executable ataupun file terkompresi. Pada kasus ini, steganografi akan diterapkan untuk menyembunyikan sebuah file terkompresi dalam format Zip dengan cara membuat file header milik file yang ingin disembunyikan tidak terdaftar pada direktori pusat arsip.**

File Zip umumnya dapat dipasang kata sandi untuk melindungi keamanan dari isinya. Untuk perlindungan yang lebih lanjut, algoritma Vigenere akan digunakan untuk mengenkripsi kata sandi, sehingga untuk mengakses isi Zip, diperlukan kunci dan ciphertext yang akan diubah menjadi plaintext password yang sebenarnya.

Kata kunci: Steganografi, Vigenere, file terkompresi, Zip

I. PENDAHULUAN

Pengompresian/pengarsipan file komputer banyak dipakai untuk mengecilkan ukuran file-file yang dikompres maupun sekedar membuat suatu media penyimpan/kontainer file agar terlihat lebih teratur. Format *Zip* merupakan salah satu format pengarsipan *file* komputer yang sudah lama beredar dan sangat umum dipakai sebelum format pengarsipan yang lebih baru seperti *.rar* dan *.7z* mulai terkenal. *Zip* dapat diandalkan berkat keunggulannya dalam aspek kecepatan dan *support* langsung dari sistem operasi untuk membuka/mengekstraksi isi arsip.

Steganografi (dalam bahasa Yunani berarti tulisan tersembunyi) merupakan salah satu teknik kriptografi dengan cara menyembunyikan sebuah pesan di dalam sebuah pesan/media lain dengan tujuan untuk menghilangkan kecurigaan dari pihak tak berwenang.

Vigenere Cipher merupakan sebuah cipher substitusi *polyalphabetic* klasik. Cipher substitusi artinya setiap unit (bisa satu atau lebih kata) pada *plaintext* diganti oleh sebuah unit lain sehingga membentuk *ciphertext*.

II. DASAR TEORI

A. *Format Pengarsipan Zip*

Gambar 2.1 *Icon* yang biasa digunakan untuk merepresentasikan file berformat .zip

Format pengarsipan *Zip* pertama kali diciptakan oleh Phil Katz dari perusahaan PKWARE,pada tahun 1986 dan pertama kali diimplementasikan pada program PKZIP buatan mereka. Pengembangan format pengarsipan ini dianggap sebagai pengganti format pengarsipan *ARJ* yang umum digunakan pada jaman itu, format *Zip* dianggap lebih unggul karena operasi yang dilakukan lebih cepat daripada format pengarsipan lainnya pada jaman itu.

Sebuah file *Zip* dapat menyimpan beberapa file yang dikompresi secara *lossless*. Setiap file di dalam arsip disimpan secara terpisah sehingga file-file dapat dikompres menggunakan metode kompresi yang berbeda-beda. Hal ini juga membuat program pengarsipan tidak perlu melakukan kompresi/dekompresi ulang seluruh arsip ketika ingin menambah atau mengekstrak suatu file dari arsip *Zip*. Ukuran minimum sebuah file *Zip* adalah 22 *byte*, sedangkan ukuran maksimalnya adalah 4,294,967,295 *byte* (sekitar 4GB). Untuk mengatasi batas ukuran maksimal tersebut, maka PKWARE membuat variasi *Zip* yang bernama *ZIP64* yang memiliki batas ukuran maksimal sebesar 2^{64} *byte* (16 EiB/ExbiByte). File *Zip* mendukung

metode-metode kompresi antara lain: *store* (tanpa kompresi sama sekali), *shrunk*, *reduced*, *imploded*, *tokenizing*, *Deflated* (paling banyak dipakai), *Deflate64*, *bzip2*, *LZMA (EFS)*, *WavPack*, dan *PPMD*. Untuk enkripsi, *Zip* mendukung enkripsi simetris sederhana via *password* yang sangat mudah untuk dipecahkan. Oleh karena itu, PKWARE mengimplementasikan fitur enkripsi yang lebih aman pada versi *Zip* yang lebih baru, antara lain *AES*, enkripsi *header* arsip, *Triple DES*, *Digital Certificate*, dan lain-lain. Pengenkripsian nama file juga mulai didukung untuk mengenkripsi metadata pada direktori pusat arsip.

Selain pada aspek kecepatan, keunggulan lain dari format *Zip* antara lain, oleh karena popularitasnya, maka *tools* dan *library* yang berhubungan dengan file *Zip* sangat melimpah. Beberapa sistem operasi terutama sistem operasi berbasis *Microsoft Windows* sudah dapat membuka dan mengekstraksi file *Zip* tanpa perlu bantuan aplikasi eksternal sejak *Windows 98 Plus!* (fitur ini disebut *"Compressed Folders"*).

Hanya saja dengan pesatnya perkembangan format pengarsipan/pengkompresian yang lebih modern, terutama format-format yang memiliki rasio kompresi yang jauh lebih tinggi, popularitas *Zip* makin menurun. Kekurangan lainnya adalah, minimnya fitur-fitur yang tersedia (seperti *data recovery*, enkripsi *AES*, dll) dibandingkan dengan format-format yang lebih modern.

B. *Steganografi*

Steganografi dibagi menjadi dua jenis, yaitu steganografi teks di mana pesan yang tersembunyi disisipkan di dalam pesan lain, dan steganografi digital, di mana pesan/file yang tersembunyi disisipkan di dalam pesan/file lainnya.

Contoh Steganografi teks:

Gerakan orang-orang dari yogya enggan ambil resiko

Covertext:

erakan rang-rang ari ogya nggan mbil esiko

Hiddentext:

Good year

Stegotext:

Gerakan **o**rang-**o**rang **d**ari **y**ogya **e**nggan **a**mbil **r**esiko

Untuk steganografi digital, pesan/file yang disembunyikan biasanya disisipkan di dalam media-media seperti file gambar, audio, video, maupun file arsip terkompresi atau *executable*.

Steganografi terdiri atas empat komponen:

- *Embedded Message*/*hiddentext*: pesan yang akan disembunyikan, dapat berupa teks maupun media lain (gambar/video/audio/etc)
- *Cover-object*: pesan/media yang digunakan sebagai media penyisipan/penyembunyian *embedded message*
- *Stego-object*: pesan/media yang sudah diisi *embedded message*
- *Stego-key*: kunci yang digunakan untuk menyimpan dan mengekstrak *embedded message* dari *cover-object*

Walaupun masih termasuk sebagai sebuah teknik kriptografi, Steganografi tidak dapat digunakan sebagai pengganti kriptografi, karena fokus dari Steganografi adalah penyembunyian wujud pesan, bukan penyembunyian isi pesan seperti dalam konteks kriptografi.

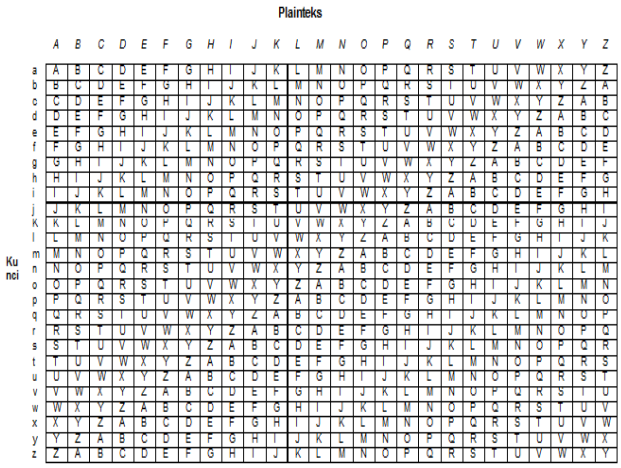
C. *Vigenere Cipher*

Vigenere Cipher merupakan sebuah algoritma kriptografi klasik yang pertama kali dipublikasikan pada tahun 1586 oleh seorang diplomat Perancis bernama Blaise de Vigenere. Sebelum dipublikasikan, sebenarnya algoritma ini sudah terdapat pada buku La Cifra del Sig oleh Giovan Batista Belaso tiga puluh tahun sebelumnya, tapi algoritma ini masih belum dikenal sampai Blaise de Vigenere mempublikasikannya. Algoritma kriptografi ini menggunakan sebuah table yang disebut Bujursangkar Vigenere atau *tabula rectal* yang berisi huruf-huruf ciphertext yang diperoleh dengan *Caesar Cipher*. Untuk menghasilkan *ciphertext*, huruf *plaintext* digeser sesuai dengan posisi huruf pada kunci. Contoh:

Plainteks : THIS PLAINTEXT
Kunci : sony sonysonys
Cipherteks : **LVVQ HZNGFHRVL**

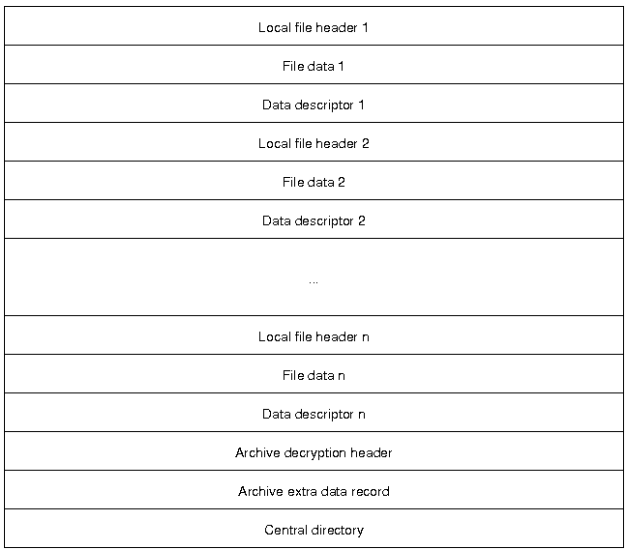
Dengan (T + s) mod 26 = L, (H + o) mod 26 = V, dan seterusnya. Sehingga tidak seperti *Caesar cipher* atau *cipher* monoalfabetik lainnya, pada Vigenere huruf yang sama tidak selalu dienkripsi menjadi huruf cipheteks yang sama.

Vigenere Cipher berhasil dipecahkan oleh Babbage dan Kasiski. Metode yang digunakan bernama metode Kasiski, metode ini menggunakan kelemahan *Vigenere*



Gambar 2.2 Bujursangkar Vigenere

III. PERANCANGAN



Gambar 3.1 Struktur detil sebuah file .zip

Sebuah file *Zip* terdiri dari *local file header*, file local, dan direktori pusat. Pada saat sebuah program membuka sebuah arsip, yang dibaca pertama kali adalah direktori. Di dalam direktori pusat terdapat pointer-pointer yang menunjuk ke file-file yang ada di dalam arsip. *File header* yang tidak terdaftar pada direktori tidak akan terbaca oleh program.

Gambar 3.2 Struktur normal sebuah arsip *Zip*

Gambar di atas menggambarkan struktur sebuah arsip *Zip* yang normal, dengan semua file yang berada di dalam arsip *Zip* benar-benar terdaftar pada direktori, sehingga program Zip dapat menampilkan semua file secara semestinya.

Gambar 3.3 Struktur arsip *Zip* dengan sebuah file tersembunyi

Pada contoh di atas, sebuah *file* bernama *yodawgyo.exe* tidak terdaftar pada direktori, sehingga program Zip tidak dapat menampilkan maupun mengekstraksi *file yodawgyo.exe*.

Penerapan Steganografi dalam kasus ini adalah dengan membuat agar *local file header* tidak terdaftar pada direktori sehingga program tidak dapat membaca adanya file tersembunyi tersebut ketika program membuka arsip. Dengan kata lain, *cover object* adalah file arsip *Zip* itu sendiri, yang menjadi *embedded object* adalah isi dari arsip yang akan disembunyikan, sedangkan file arsip yang memiliki file tersembunyi merupakan *stego-object-nya*.

Untuk meningkatkan keamanan, arsip *Zip* dapat diberi *password* untuk mengakses arsip tersebut. Vigenere Cipher digunakan untuk mengenkripsi *password* yang diinput, tetapi karena secara teknis program pengkompresi hanya dapat men-*support* satu *password* enkripsi simetris saja, maka program yang dibuat akan secara otomatis membuat *Cipherkey* sebagai *password* dari arsip.

IV. IMPLEMENTASI

Aplikasi dibuat dalam bahasa C#, menggunakan IDE *Microsoft Visual Studio 2010*. *Library* yang digunakan untuk melakukan operasi-operasi yang melibatkan file Zip (seperti membuka file *Zip*, menambah file baru ke dalam *Zip*, dan lain-lain) adalah *#ZipLib* (dibaca *SharpZipLib*). *#ZipLib* juga digunakan untuk meng-*generate* direktori entri untuk setiap entri Zip secara otomatis.

Aplikasi dibagi menjadi 4 bagian, antara lain:

- Bagian pertama adalah bagian di mana pengguna dapat membuka arsip *Zip* dengan menentukan *path* arsip *Zip* yang ingin dibuka terlebih dahulu. Jika arsip ternyata diberi *password*, maka terdapat pilihan untuk mendekripsi *password*.
- Bagian kedua menampilkan isi arsip yang sebenarnya, termasuk file yang disembunyikan dengan

isi arsip dibaca oleh program. Pengguna dapat menyembunyikan/menampilkan suatu *file* dengan menggunakan tanda centang pada daftar *file* yang ada, dan melakukan ekstraksi *file* dengan mengklik file tersebut dua kali.

- Bagian ketiga untuk menambah *file* ke dalam arsip, disediakan juga opsi untuk menampilkan/menyembunyikan *file* yang akan dimasukkan ke dalam arsip.
- Bagian keempat adalah opsi untuk memberi password pada arsip, dan menyimpan arsip *Zip*.

Untuk membuka *file* arsip *Zip*, berikut potongan *source code*-nya:

```
private void Open()
{
       this.HasNewFiles = false;
       VisibleList.Items.Clear();
       ActualList.Items.Clear();

       if (browseZip.Text.Length > 0)
       {
              ICSharpCode.SharpZipLibZipZipEntry
zipEntry;
              ICSharpCode.SharpZipLibZipZipFile
zipFile = new
ICSharpCode.SharpZipLibZipZipFile(zipPath.Text);

              if (chkDecrypt.Checked)
              { //decrypt password jika ada
                  zipFile.Password =
txtOpenPassword.Text;
              }

              // tampilkan file yang tidak hidden
saja
              for (int n = 0; n < zipFile.Size;
n++)
                {
                    zipEntry = zipFile[n];
                    AddListViewItem(zipFile[n],
VisibleList);
                }


                // tampilkan semua file pada
listbox, termasuk yang di-hidden
                zipEntry = zipFile[0];
                AddListViewItem(zipEntry,
ActualList);
                int entryIndex = 0;
                while
(zipFile.HasSuccessor(zipEntry))
                {
                    zipEntry =
zipFile.GetAttachedEntry(zipEntry);
                    AddListViewItem(zipEntry,
ActualList);
                    entryIndex++;
                }

                zipFile.Close();
            }
        }
```

Program menggunakan fungsi yang sudah ada dari *library SharpZipLib* untuk mengakses file *Zip* serta entri direktorinya. Ketika membuka sebuah arsip *Zip*, pertama-tama program memeriksa apakah arsip diproteksi *password* atau tidak. Jika ada, maka program akan mencoba untuk mendekripsi *password* menggunakan *password* yang diinput. Setelah berhasil mendekripsi, fungsi akan mempopulasikan kedua *listbox* yang ada dengan daftar dari file-file di dalam arsip. Satu *listbox* berisi file yang terdaftar pada entri, dengan kata lain file yang terlihat oleh program, sedangkan *listbox* lain berisi seluruh file yang terdapat di dalam *Zip*, termasuk file-file yang disembunyikan.

Karena file *Zip* memiliki karakteristik di mana jika entri direktori tidak dimulai dari *offset* dari *file header* pertama (0), program pengarsip tidak dapat memproses file *Zip* tersebut, maka *file* pertama yang terdaftar pada direktori pusat tidak dapat disembunyikan.

Berikut adalah potongan *source code* untuk memeriksa apakah entri sebuah direktori pada arsip *Zip* kosong atau tidak. Jika kosong, *file* yang bersangkutan tidak akan ditampilkan oleh program.

```
foreach (ListViewItem viewItem in
ActualList.Items)
{
ICSharpCode.SharpZipLibZipZipEntry zipEntry =
viewItem.Tag as
ICSharpCode.SharpZipLibZipZipEntry;
Stream inputStream;
if (zipEntry == null)
{
   inputStream = new FileStream(viewItem.Text,
FileMode.Open);
   zipEntry = new
ICSharpCode.SharpZipLibZipZipEntry(Path.GetFileNa
me(viewItem.Text));
}
else
{
   inputStream =
zipFile.GetInputStream(zipEntry);
}
```

Untuk menyembunyikan sebuah *file*, program harus menyimpan dahulu arsip yang sudah dimodifikasi. File pertama yang terdaftar pada *listbox* tidak dapat disembunyikan karena file bertindak sebagai *anchor* untuk direktori pusat. Pengkompresian *Zip* baru selalu dilakukan dengan metode kompresi *Deflated*.

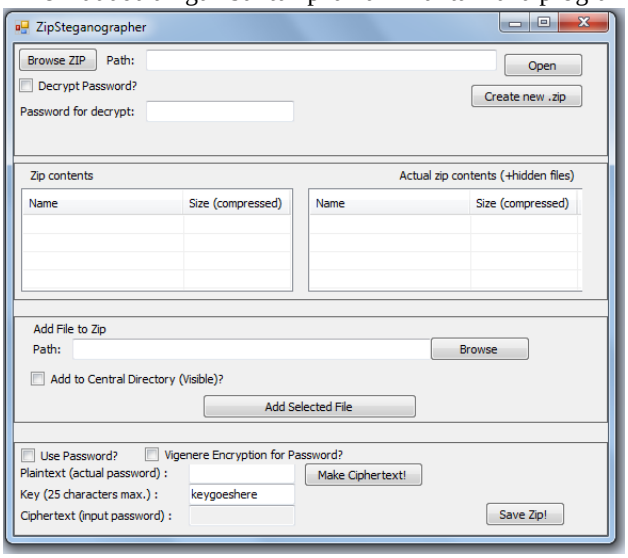
```
SaveFileDialog dlg = new SaveFileDialog();
dlg.Filter = "Zip-Archive (*.zip)|*.zip";
dlg.RestoreDirectory = true;
if (dlg.ShowDialog(this) == DialogResult.OK)
        {
        ZipFiles(dlg.FileName, (isPassword.Checked ?
txtSavePassword.Text : null));
        zipPath.Text = dlg.FileName;
        txtOpenPassword.Text = txtSavePassword.Text;
        chkDecrypt.Checked = isPassword.Checked;
        Open();
        }
```

Untuk bagian *password*, program dapat meng-*generate ciphertext* dari password/*plaintext* yang dimasukkan jika opsi Vigenere dipilih. Otomatis *ciphertext* pula menjadi *password* bagi file arsip.

```
if (isVigenere.Checked)
{
       if ((keyText.Text.Length < 25) && (keyText !=
```

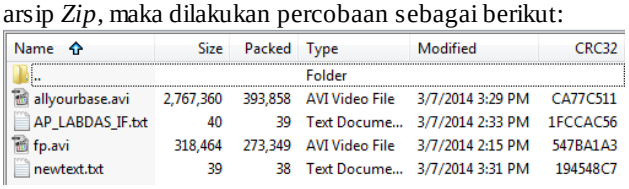
```
null))
        {
                String pass =
Vigenere.Encipher(txtSavePassword.Text,
keyText.Text);
                txtSavePassword.Text = pass;
        }
}
```

Berikut adalah gambar tampilan akhir antarmuka program



V. PERCOBAAN

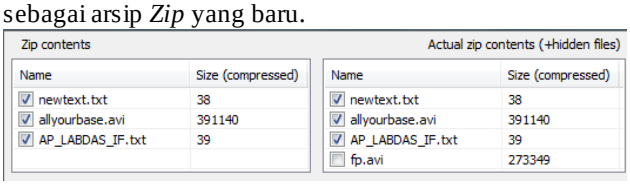
Untuk membuktikan apakah dengan menghilangkan entri *local file header* pada direktori utama dapat membuat program tidak dapat membaca keberadaan suatu file dalam arsip *Zip*, maka dilakukan percobaan sebagai berikut:



Name	Size	Packed	Type	Modified	CRC32
..			Folder		
allyourbase.avi	2,767,360	393,858	AVI Video File	3/7/2014 3:29 PM	CA77C511
AP_LABDAS_IF.txt	40	39	Text Docume...	3/7/2014 2:33 PM	1FCCAC56
fp.avi	318,464	273,349	AVI Video File	3/7/2014 2:15 PM	547BA1A3
newtext.txt	39	38	Text Docume...	3/7/2014 3:31 PM	194548C7

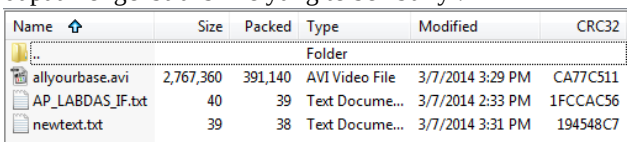
Gambar 5.1 Isi *Zip* sebelum ditambah file baru

File *Zip* awalnya berisi empat file, yaitu newtext.txt, allyourbase.avi, AP_LABDAS_IF.txt, dan fp.avi. Misalnya, file yang akan disembunyikan adalah fp.avi, maka tinggal menghapus centang yang sesuai, lalu simpan sebagai arsip *Zip* yang baru.



Gambar 5.2 Isi Zip dengan file tersembunyi

Seperti yang dapat dilihat pada gambar 5.2, pada *listbox* kiri, file *fp.avi* yang disembunyikan tidak terdaftar, sedangkan pada *listbox* kanan yang berisi isi file yang sesungguhnya, file fp.avi yang disembunyikan ternyata masih ada. Lalu, apa yang terjadi jika file *Zip* dibuka menggunakan program eksternal? Ternyata file yang disembunyikan juga tidak terbaca oleh program eksternal, yang dalam kasus ini, program yang digunakan adalah *WinRAR*. Ketika isi arsip diekstraksi juga, *WinRAR* tidak dapat mengekstraksi file yang tersembunyi.



Name	Size	Packed	Type	Modified	CRC32
..			Folder		
allyourbase.avi	2,767,360	391,140	AVI Video File	3/7/2014 3:29 PM	CA77C511
AP_LABDAS_IF.txt	40	39	Text Docume...	3/7/2014 2:33 PM	1FCCAC56
newtext.txt	39	38	Text Docume...	3/7/2014 3:31 PM	194548C7

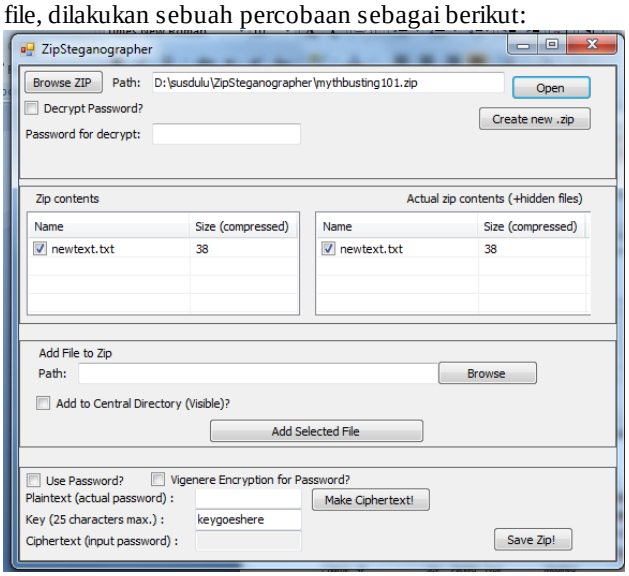
Gambar 5.3 Win*RAR* tidak dapat membaca file tersembunyi

Tidak hanya itu, bahkan ukuran file arsip sesudah penyembunyian file terlihat lebih kecil dibandingkan dengan ukuran file arsip yang aslinya seolah-olah file tersembunyi tidak ada di dalam file arsip tersebut.

test.zip	3/18/2014 7:21 PM	WinRAR ZIP archive	653 KB
test2.zip	3/18/2014 7:25 PM	WinRAR ZIP archive	650 KB

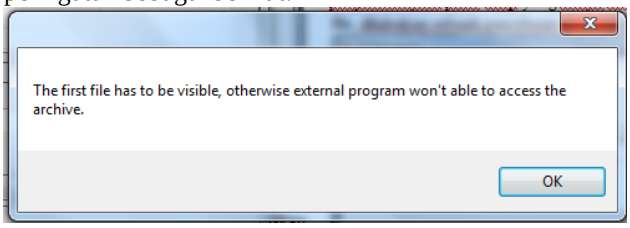
Gambar 5.4 Perbedaan ukuran arsip sebelum dan sesudah penyembunyian file

Untuk menguji apakah program dapat melakukan penyembunyian pada arsip yang hanya memiliki sebuah file, dilakukan sebuah percobaan sebagai berikut:



Gambar 5.5 Isi dari arsip yang dijadikan percobaan

Di dalam file *Zip* hanya terdapat satu file saja. Jika satu-satunya file tersebut disembunyikan, lalu arsip disimpan sebagai arsip baru maka muncul pesan *error* yang berisi peringatan sebagai berikut.



Gambar 5.6 Pesan *error* yang muncul ketika menyimpan arsip dengan file pertama disembunyikan

VI. KESIMPULAN

Steganografi pada arsip *Zip* dapat diterapkan dengan cara membuat agar *local file header* tidak terdaftar pada direktori pusat arsip. Dengan cara tersebut, program pengarsip pada umumnya tidak dapat membaca maupun mengekstraksi file-file yang tersembunyi.

Oleh karena kemudahannya untuk menyembunyikan sebuah file tersebut, maka arsip *Zip* dapat dijadikan sebuah media yang layak dipakai untuk Steganografi, hanya saja isi dari *Zip* harus lebih dari satu file karena program pengarsip umumnya tidak dapat mengakses file arsip *Zip* yang *offset file entry*-nya pada direktori utama tidak dimulai dari 0.

REFERENSI

[1] http://www.pkware.com/support/zip-app-note/ diakses tanggal 15 Maret 2014 pukul 19.45

[2] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2012-2013/Algoritma%20Kriptografi%20Klasik_bag2%20(2013).ppt diakses tanggal 15 Maret 2014 pukul 20.00

[3] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2012-2013/Steganografi%20(2013).ppt diakses tanggal 15 Maret 2014 pukul 20.00

[4] http://www.icsharpcode.net/OpenSource/SharpZipLib/ diakses tanggal 16 Maret 2014 pukul 19.50

[5] https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip.html diakses tanggal 18 Maret 2014 pukul 17.30

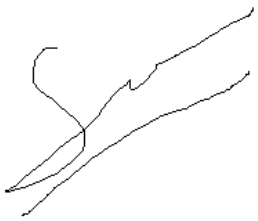
[6] https://www.pkware.com/documents/casestudies/APPNOTE.TXT diakses tanggal 18 Maret 2014 pukul 17.30

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 18 Maret 2014

ttd



Sandy Gunawan Tanuwijaya