Ekplorasi Penerapan Steganografi Dengan Eksploitasi Spesifikasi Format Media Container Populer

Andika Pratama. NIM 13507005
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
if17005@itb.ac.id

***Abstrak*— Dengan berkembang pesatnya teknologi untuk telekomunikasi, semakin banyak pesan dalam media digital, dan kebutuhan untuk steganografi data digitalpun semakin besar. Teknik Steganografi digital yang paling populer adalah dengan menyembunyikan pesan dalam content multimedia, namun terdapat juga teknik-teknik lain yang lebih jarang digunakan dan belum banyak dieksplorasi. Salah satu teknik steganografi yang kurang diperhatikan adalah steganografi dengan menyelipkan data pesan diantara data content multimedia dengan mengeksploitasi spesifikasi dari format media container yang digunakan untuk membungkus content. Untuk mendorong penelitian lebih jauh mengenai topik ini, penulis akan menganalisis dan mengksplorasi teknik steganografi jenis ini dengan menganalisis peluang penerapannya pada beberapa format media container yang umum digunakan.**

***KataKunci*— eksploitasi, spesifikasi format media container, steganografi**

I. PENDAHULUAN

Seiring dengan perkembangan teknologi informasi, internet menjadi salah satu kebutuhan penting di dalam kehidupan masyarakat sehari-hari. Pertukaran informasi pun kerap terjadi setiap saat di dunia maya. Sayangnya, pertukaran informasi melalui media ini tidaklah aman. Pesan penting yang kita kirimkan dapat saja disadap oleh orang lain. Bahkan transaksi yang penting pun dapat terganggu. Oleh karena itu, berkembanglah teknik-teknik kriptografi yang melindungi pesan yang dikirim. Dengan teknik-teknik kriptografi, walaupun makna suatu pesan menjadi hilang ataupun kacau, keberadaan pesan rahasia tersebut masih dapat diidentifikasi karena pesan akan terlihat ganjil atau aneh. Hal ini yang membuat penyadap informasi menyadari adanya pesan rahasia yang disembunyikan dan melakukan kriptanalis.

Berbeda dengan kriptografi, steganografi adalah salah satu metode yang dapat digunakan di dalam mengamankan akses terhadap suatu pesan tanpa menyebabkan kecurigaan. Penyembunyian ataupun penyamaran pesan ini dibuat sedemikian rupa sehingga pihak lain tidak curiga akan adanya pesan lain di dalam pesan yang dikirimkan. Hanya pihak penerima yang sah saja yang dapat mengetahui bahwa ada pesan lain yang disembunyikan di dalam pesan yang dikirim. Perbedaannya dengan kriptografi adalah kriptografi mengubah ataupun mengacak karakter pesan menjadi bentuk lain yang tidak bermakna, sedangkan steganografi hanya mengaburkan ataupun menyembunyikan penyampaian pesan dengan berbagai cara dan tetap mempertahankan pesan.

Dengan berkembang pesatnya teknologi dalam dunia telekomunikasi, semakin banyak pesan yang disampaikan dalam media digital, dan kebutuhan untuk stegongrafi data digitalpun semakin banyak. Steganografi digital biasanya menggunakan media digital sebagai wadah penampung, misalnya gambar, suara, teks, maupun video.

Teknik steganografi data pada media digital yang umum digunakan adalah dengan menyisipkan pesan pada *content*, seperti teknik LSB(Least Significant Bit) untuk media gambar, yang menyembunyikan data pada bit tidak signifikan, atau teknik *spread spectrum steganography* pada data audio. Kelemahan dari teknik kriptografi seperti ini adalah proses kriptografi akan menimbulkan degradasi pada *content*. Degradasi ini umumnya terlalu halus untuk dipersepsi manusia, namun akan mudah dideteksi dengan software kriptanalis, dengan membandingkan data content dengan data sumber yang masih bersih. Salah satu kelemahan lain adalah bahwa umumnya pada teknik ini ukuran pesan rahasia yang dapat disamarkan relatif kecil, karena semakin besar pesan rahasia yang disamarkan, degradasi akan semakin besar hingga bahkan bisa dikenali oleh manusia.

Teknik kriptografi digital lain yang jauh lebih jarang digunakan adalah dengan menyembunyikan pesan pada *format media container* dari *content* tersebut. Misalnya, pada sebuah *file* format *container* BMP, kita tidak menyembunyikan pesan pada *byte* data gambar yang disimpan dalam BMP, melainkan kita menyelipkan pesan diantara data gambar dengan mengeksploitasi spesifikasi format cara file BMP menyimpan data, tanpa merusak kemampuan file untuk dibuka oleh *image*

viewer. Keuntungan utama dari teknik ini adalah bahwa karena pesan rahasia tidak pernah menyentuh data content, tidak akan ada degradasi content untuk dideteksi.

II. DEFINISI FORMAT MEDIA CONTAINER SERTA STEGANOGRAFI PADA FORMAT MEDIA CONTAINER

Format media container adalah sebuah meta-file format, dimana spesifikasinya menjabarkan bagaimana berbagai elemen data dan meta-data disimpan secara bersamaan pada file digital. Secara teori, sebuah format container dapat menyimpan data apapun, tetapi kebanyakan format container dispesialisasikan untuk kebutuhan data yang spesifik, seperti file format PNG untuk menyimpan gambar, MP3 untuk menyimpan audio, dll. Penggunaan paling umum dari format container adalah untuk membungkius data multimedia. Walau data audio dan video dapat dikodekan dengan berbagai macam algoritma yang berbeda, semuanya dapat disajikan dalam satu format container yang dikenali user.

Steganografi pada format media container bekerja dengan mengeksploitasi spesifikasi dari format media container content multimedia, untuk menyelipkan pesan rahasia diantara content multimedia, bukan pada content itu sendiri. Karena data content multimedia tidak diproses tidak akan ada degradasi pada content multimedia.

Steganografi seperti ini juga dapat dibedakan menjadi dua metode paling umum. Yang pertama adalah steganografi dengan menyimpan pesan rahasia pada slot data dalam format yang bersifat opsional. Slot data semacam ini biasanya menyimpan data yang tidak relevan untuk membuka file, sehingga tidak akan dibaca oleh aplikasi media player. Metode ini tidak akan menambah ukuran file, tetapi besar pesan rahasia yang bisa disimpan relatif kecil dan terbatas pada data slot yang boleh dioverwrite.

Metode kedua Steganografi dengan menciptakan ruang kosong diantara data format yang valid dengan mengeksploitasi data header *container*. Hal ini dimungkinkan karena banyak spesifikasi format media container yang antara lain: menyimpan informasi panjang header, menyimpan tabel lokasi offset data content dalam header, atau menyimpan pointer – pointer struktur internal dalam file, dll. Spesifikasi semacam ini biasanya dibuat agar memberikan fleksibilitas dalam penyusunan data, namun kebebasan ini juga memungkinkan kita untuk membuat ruang kosong dengan memanipulasi data tersebut. Ruang kosong diciptakan lalu dapat diisi dengan pesan rahasia. File hasil steganografi akan tetap bisa dibuka secara normal karena pesan rahasia akan dilompati oleh media player yang mengikuti prosedur standar spesifikasi format untuk mengabaikan data yang tidak diakui. Metode ini akan menambah besar file, tetapi biasanya tidak ada batasan mengenai ukuran pesan rahasia yang bisa disembunyikan.

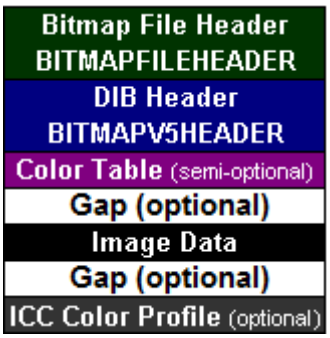
Selain itu, berbeda teknik steganografi pada content yang tidak memedulikan format file selama jenis content medianya sama, teknik ini akan berbeda penerapannya untuk tiap jenis format *container*, sesuai dengan apa dari spesifikasi format *container* yang dapat dimanfaatkan tanpa merusak kemampuan file tersebut untuk dibaca oleh media player.

III. ANALISIS PELUANG PENERAPAN STEGANOGRAFI PADA BEBERAPA FORMAT MEDIA CONTAINER POPULER

BMP

Bitmap, atau BMP adalah salah satu format file media container untuk gambar raster yang paling dasar dan umum. Gambar BMP umumnya berukuran relatif besar karena biasanya tidak terkompresi, sehingga penyisipan data tidak akan terlalu mencolok. Bedasarkan spesifikasi Microsoft, file berformat BMP memiliki struktur sebagai berikut:

Spesifikasi Format:



Gambar 1 Struktur File BMP

Data dalam file BMP disusun dalam beberapa struktur blok, yaitu Bitmap File Header, DIB Header, Image Data dan ICC Color Profile. Dari lima struktur tersebut, tiga bersifat wajib, satu opsional dan satu lagi semi-opsional(Color Table hanya wajib jika kedalaman warna kurang dari 8 bit). Diantara Image Data dan Color Table serta ICC Color Profile juga secara opsional bisa dibuat rongga kosong, dengan metode yang akan dijelaskan dibawah.

Proposal Metode Steganografi:

1. **Eksploitasi struktur Bitmap File Header**
   Terdapat dua rongga kosong yang dapat dibuat pada file bitmap, yaitu sebelum dan setelah

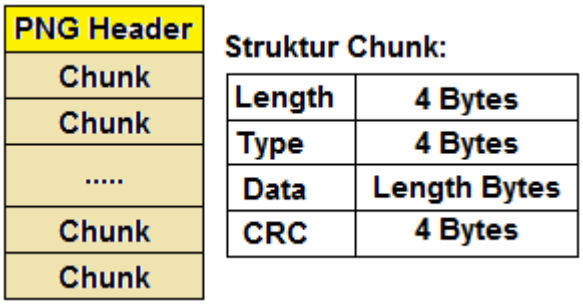
Image Data. Hal ini dilakukan dengan memanipulasi data integer 4 bytes pada offset 000A pada blok Bitmap File Header, yang menyimpan offset dimana Image data disimpan, sehingga rongga kosong dapat dibuat dengan menyimpan offset Image Data lebih besar dari panjang header. Besar rongga ini fleksibel, dapat dibuat hingga mencapai 4 GB. Rongga kosong ini lalu dapat dimanfaatkan untuk menyimpan pesan rahasia

2. **Eksploitasi *Padding* dalam Image Data**
   Image Datadalam BMP disimpan dalam bentuk array bytes 2 dimensi, dengan tinggi dan lebar sesuai dengan resolusi pixel gambar, dengan ketentuan lebar array harus kelipatan 4 bytes, jika kurang maka akan ditambah *padding*. Bytes *padding* ini tidak dianggap oleh image viewer, maka dapat dioverwrite dengan pesan rahasia. Metode ini tidak menambah besar file, tapi kuruannya relatif kecil dan hanya dapat digunakan apa bila array Image Data menggunakan *padding*. Metode ini juga hanya dapat dilakukan pada file yang lebar gambar nya dalam pixel dikali tiga bukanlah kelipatan 4.

PNG

Portable network Graphics, atau PNG adalah format container gambar raster yang mendukung lebih banyak fitur daripada file bitmap biasa, diantaranya dukungan kompresi *lossless*, alpha channel, filtering, dll. Strukturnya adalah sebagai berikut:

Spesifikasi format:



Gambar 2 Struktur File PNG

Struktur File berformat PNG diawali dengan header 8-byte signature, yang diikuti dengan serangkaian "chunk", dimana setiap chunk menyimpan suatu informasi dari gambar. Chunk juga dibagi dalam 2 jenis utama:

- **Critical Chunks:**
  Chunks ini menyimpan data penting gambar, sehingga apabila chunk ini rusak gambar tidak bisa dibuka oleh image viewer. Terdapat 4 jenis critical Chunks, yaitu :
  - IHDR : menyimpan header.
  - PLTE : menyimpan color palette.
  - IDAT : menyimpan data gambar.
  - IEND : menandai akhir gambar.

- **Anciliary Chunks:**
  Chunks ini menyimpan atribut lain gambar yang non krusial, seperti metadata, warna background, gamma, dll. Tidak seperti Critical chunks, apabila chunk ini rusak file gambar tetap bisa dibuka karena image viewer akan mengabaikannya.

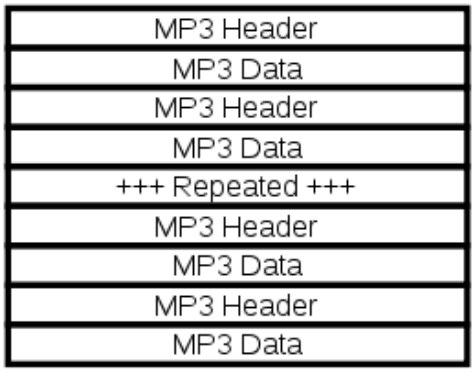
Proposal Metode Steganografi:

Struktur yang dapat dimanfaatkan adalah Anciliary Chunks. Anciliary Chunks tidak dibutuhkan oleh image viewer, sehingga field data-nya dapat di overwrite oleh pesan rahasia dengan aman. Kita juga bisa membuat Anciliary Chunks baru, dan menyimpan pesan rahasianya pada Anciliary Chunk tersebut. Apabila sebuah chunk tipenya tidak dikenali oleh image viewer, chunks ini akan diabaikan tanpa memengaruhi content gambar. Ditambah lagi jumlah Anciliary Chunks tidak dibatasi, sehingga tidak ada batasan sebesar apa pesan rahasia yang dapat disimpan.

MP3

MP3 adalah format file audio yang mengkompress data audio dengan menggunakan model psikoakustik untuk mengurangi atau menghilangkan data suara yang tidak dikenali oleh telinga manusia dan mengkompress sisanya secara efisien. MP3 merupakan format audio yang paling populer dan didukung oleh hampir semua media player.

Spesifikasi Format:



Gambar 3 Struktur File MP3

Format file MP3 terdiri dari serangkaian MP3 *frames*, yang terdiri dari sebuah header dan sebuah blok data. Struktur Header *frame* MP3 adalah sebagai berikut:

Panjang bits	Deskripsi
11	*Frame* sync
2	MPEG Audio version ID
2	Layer description
1	Protection bit

4	Bitrate Index
2	Sampling index
1	*Padding* bit
1	Private bit
2	Channel Mode
2	Mode extension
1	Copyright
1	Original
2	Emphasis

Sedangkan struktur Data MP3 pada setiap *frame* adlah blok data yang menyimpan kompresi data audio dalam frekuensi dan ampitudo. Selain itu, kebanyakan file MP3 sekarang menyimpan meta data ID3, yang disimpan sebelum *frame*s-*frame* MP3.

Proposal Metode Steganografi:

Walau tidak begitu besar, terdapat beberapa bit dalam header *frame* mp3 yang dapat dimanfaatkan yaitu : Private Bit, Copyright Bit, Original Bit, serta Emphasis bit. Para bit ini jarang digunakan serta tidak akan merusak playability file jika dioverwrite dengan pesan rahasia. Selain itu, kita bisa memanfaatkan juga fitur *padding* file MP3 dengan mengaktifkan *padding* bit.

Padding dalam file MP3 berfungsi agar MP3 data dalam *frame* tepat sebesar bitrate yang dispesifikasikan. Besar *padding* yang didapatkan tergantung pada mode Layer apa yang digunakan oleh file MP3, slot *padding* untuk Layer I memiliki panjang 32 bit, sedangkan untuk Layer II dan Layer II panjangnya 8 bit. Karena *padding* tidak dibaca oleh media player, data tersebut dapat jika dioverwrite dengan pesan rahasia.

Dengan menggabungkan kedua cara tersebut, kita bisa mendapatkan tempat 37 bit untuk tipe layer I, atau 13 bit untuk tipe Layer II dan Layer III untuk setiap *Frame* dari file MP3 yang dapat ditulis dengan cipher data. Menurut standar, setiap detik audio dalam file MP3 dikodekan dalam 38 *frame*, maka besar pesan rahasia yang dapat ditulis tergantung dari lamanya data audio. Sebagai contoh, untuk sebuah file MP3 yang lamanya 5 menit akan terdiri dari sekitar 11400 *frame*, yang berarti tersedia tempat sekitar 18 kB(untuk tipe Layer II/III) atau 51,5 kB(untuk tipe Layer I) untuk pesan rahasia.

FLV

Flash Video, atau FLV adalah format file container yang awalnya digunakan untuk menampilkan video di internet melalui flash player. Format FLV dengan cepat telah menjadi format pilihan untuk mengembed video di web. Penggunanya antara lain YouTube, Yahoo! Video, Hulu, dll.

Karena kebanyakan file video yang didownload dari web berformat FLV, Dari asalnya di web, kini file FLV juga mulai sering digunakan di komputer desktop dan sekarang hampir semua media player populer mendukung FLV.

Spesifikasi Format:

Gambar 4 Struktur File FLV

Format file FLV diawali dengan sebuah FLV Header, yang lalu diikuti dengan serangkaian Tag.

Struktur FLV Header:

Field	Deskripsi
Signature	Penanda ini file FLV
Version	Versi file
TypeFlagsReserved	Kosong, disimpan untuk ekspansi
TypeFlagsAudio	Menandai adanya Audio Tags
TypeFlagsReserved	Kosong, disimpan untuk ekspansi
TypeFlagsVideo	Menandai adanya Video Tags
DataOffset	Panjang header ini dalam bytes

Struktur FLV Tag:

Field	Deskripsi
Reserved	Direservasi untuk FMS
Filter	Menandai jika data dienkripsi
TagType	Tipe content tag. Ada tiga jenis: Audio, Video, dan ScriptData
DataSize	Panjang tag setelah StreamID
Timestamp	Menandai kapan tag dimainkan
StreamID	Selalu 0
TagHeader	Tag Header sesuai content tag
EncryptionHeader	Header enkripsi jika Filter=1
FilterParams	Parameter Filter jika Filter=1
Data	Data sesuai content tag

Proposal Metode Steganografi:

1. **Eksploitasi struktur FLV Header**
   Field DataOffset menyimpan panjang header dai file FLV. Nilai dari DataOffset ini dirancang agar dapat diubah untuk mendukung kemungkinan header yang lebih besar pada versi format kedepan. Fleksibilitas ini juga dapat dimanfaatkan untuk menghasilkan rongga kosong yang dapat dibuat dengan mengeset nilai DataOffset lebih besar dari panjang header. Besar rongga ini fleksibel, dapat dibuat hingga

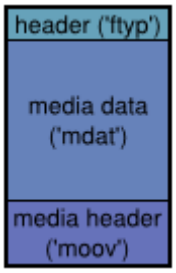
mencapai 4 GB. Rongga kosong ini lalu dapat dimanfaatkan untuk menyimpan pesan rahasia.

2. **Menyimpan Pesan rahasia sebagai ScriptData**
   FLV menndukung penyimpanan Content Tag ScriptData untuk berinterkasi dengan player, namun fitur ini hanya digunakan oleh Adobe Flash Player di internet browser, sementara untuk penggunaan oleh media player didesktop Tag ini umumnya diabaikan, sehingga dapat dioverwrite dengan pesan rahasia jika file ini tidak akan disajikan melalaui web.
   Kita juga bisa membuat dan menempelkan Tag ScriptData baru kepada file, yang ukurannya dapat dibuat sebesar kebutuhan. Karena tidak ada batasan jumlah tag dalam format FLV, besar pesan rahasia yang dapat disisipi tidak ada batasnya.

MP4

MPEG-4 Part 14, atau MP4 adalah standar format file container yang dibuat oleh ISO MPEG untuk membungkus stream video dan audio. Format ini didukung oleh banyak vendor dan dijadika sebagai format file studio standar di kebanyakan *smartphone* dan *gadget*.

Spesifikasi format:



Gambar 5 Struktur File MP4

MPEG-4 file format didasarkan pada spesifikasi ISO 14496-14, dimana format ini dibentuk dari elemen-elemen yang disebut 'atom'. Atom adalah potongan data yang terdiri dari sebuah 4-byte nilai panjang atom, sebuah 4-byte identifier tipe ASCII dan payload sebesar nailai panjang sebelumnya. Atom ini dapat bersifat rekursif, dimana payload menyimpan lebih banyak atom lagi.

Dalam spesifikasi file format MP4, terdapat tiga atom utama yang wajib ada: that are required in every proper QuickTime or MP4 file:

- Atom file header (atom ID '`ftyp`') : atom ini hanya berukuran beberapa bytes dan tujuannya adalah menunjukkan bahwa ini adalah file MP4.
- Atom media header (atom ID '`moov`') : atom ini menyimpan infomasi header dari video yang disimpan dan lokasi data video tersebut dalam atom mdat. Informasi metadata juga disimpan disini.
- Atom media data (atom ID '`mdat`') : Atom ini menyimpan data dari media video dan audio.

Mengenai susunan atom sendiri, satu-satunya aturan adalah bahwa atom `ftyp` harus disimpan didepan. Untuk tipe atom lain tidak harus berurutan.

Proposal Metode Steganografi:

1. **Menyimpan pesan rahasia sebagai '`free`' atom**
   Terdapat beberapa tipe atom yang cukup umum seperti tipe '`free`', namun tipe atom ini hanyalah *placeholder*, dan menyimpan data yang dibutuhkan oleh media player untuk memainkan file, sehingga dapat kita overwrite dengan pesan rahasia. Karena tidak ada aturan mengenai jumlah atom, kita juga bisa membuat atom baru untuk menyimpan pesan rahasia sebesar dibutuhkan.

2. **Menyelipkan pesan rahasia dalam atom mdat**
   Salah satu hal menarik dari atom `mdat` adalah bahwa atom ini tidak memiliki struktur yang jelas. Dari sisi file format MP4, `mdat` hanylah blob data yang menyimpan media samples (*frame* audio dan video). Lokasi-lokasi samples disimpan dalam tabel chunk offset tables(atom '`stco`') yang disimpan dalam atom `moov`. Table ini mencatat file offset untuk setiap potongan sample dari video. Keuntungan utama adalah bahwa bagian dile yang berada di atom `mdat` tetapi tidak disebutkan dik `stcos` akan diabaikan oleh media player, sehingga kita dapat menyelipkan pesan rahasia dengan membuat rongga kosong melalui manipulasi `stcos`.

MKV

Matroska Multimedia Container, atau MKV adalah format standar gratis dan *open source* yang dirancang sebebas mungkin, bisa menyimpan track multimedia dalam *codec* apapun tanpa batasan jumlah semuanya hanya dalam sebuah file. Format ini populer di lingkungan *open source* seperti Linux.

Spesifikasi format:

Gambar 6 Struktur File MKV

Matroska Multimedia Container file format didasarkan pada prinsip *Extensible Binary Meta Language*(EBML). EBML menerapkan prinsip XML dalam konteks data biner, dan seperti XML, EBML dapat menyimpan data jenis apapun. File MKV adalah subset dari EBML yang menspesifikasikan node-node khusus yang diperlukan untuk menyimpan data audio dan video. Node-node ini adalah:

Node	Deskripsi
Header	menyimpan informasi versi dan tipe EBML
Metaseek	menyimpan informasai index dimana semua node lain disimpan.
Segment Information	menyimpan informasi dasar file secara keseluruhan, ini termasuk judul file, Id unik dari file, dll.
Track	menyimpan informasi mengenai semua track multimedia.
Chapters	menyimpan daftar chapter dalam video.
Clusters	menyimpan semua data video *frame*s dan audio untuk setiap track.
Cueing Data	menyimpan cues, atau index dari masing-masing track.
Attachment	untuk menyimpan file attachment. Node ini bebas dapat diisi file jenis apa saja
Tagging	menyimpan semua data tag pada file.

Proposal Metode Steganografi yang bisa digunakan:

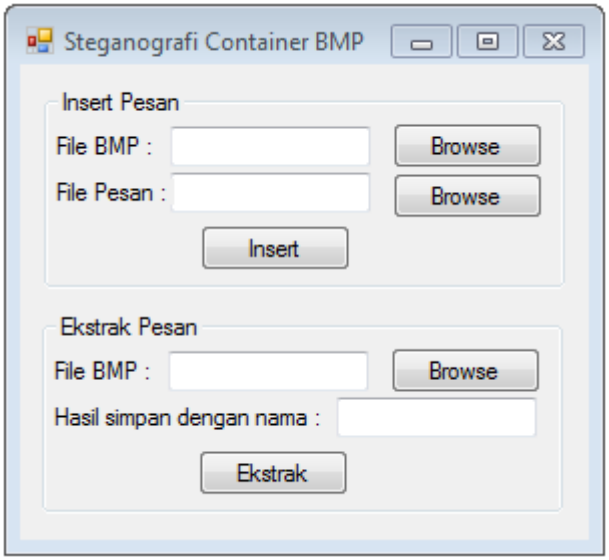
1. **Menyimpan pesan rahasia dalam Node Attachment**
   Karena MKV mendukung file lampiran, cara yang paling mudah untuk menulis pesan rahasia adalah dengan melampirkannya sebagai file pada Node Attachment. Karena ukuran Node attachment tidak dibatasi, maka besarnya pesan rahasia juga tidak terbatas. Kelemahan metode ini adalah, bahwa pesan rahasia akan mudah dilihat oleh aplikasi yang bisa mengekstrak attachment, tetapi umumnya media player tidak memiliki kemampuan ini.

2. **Menyimpan pesan rahasia dalam Node Baru.**
   Dalam file MKV tidak ada aturan urutan dan jumlah node yang harus dipatuhi, dan apabilai media player menmukan node yang tidak dikenalinya, node tersebut akan diabaikan. Maka, jika kita membuat sebuah node baru dengan tipe yang tidak ada di spesifikasi MKV, kita dapat menuliskan pesan rahasia kedalam node tersebut. Karena fleksibilitas dari EBML, ukuran pesan rahasia yang bisa ditulis juga tidak terbatas.

III. PENGUJIAN PROPOSAL PENERAPAN METODE STEGANOGRAFI UNTUK FILE BMP

Dari enam jenis format media container yang telah penulis bahas, penulis akan mencoba menguji analisis yang dibuat untuk salah satu jenis format, bagaimana hasilnya dan membandingkannya dengan Steganografi biasa.

Untuk itu, penulis membuat program sederhana untuk melakukan mencobakan penerapan metode steganografi pada file container berformat BMP. Programnya adalah sebagai berikut:



Gambar 7 Screenshot Program

Cara bekerja program ini mengikuti metode steganografi BMP nomor 1, seperti yang dibahas diatas. Lengkapnya adalah:

Proses penyamaran pesan:

1. Membuat rongga kosong antara header dan Image Data dengan menggeser file BMP kebelakang.

2. Memanipulasi header BMP sehingga offset ImageData tetap merujuk ke tempat yang benar
3. Menuliskan pesan rahasia pada rongga kosong yang telah dibuat.
4. Menyimpan file hasil modifikasi.

Sedangkan untuk proses ekstraksi pesan:

1. Membaca lokasi offset Data Image
2. Membaca semua data mulai dari berakhirnya header hingga offset gambar
3. Data yang dibaca tersebut disimpan sebagai file hasil ekstraksi.

File BMP yang telah disisipi oleh program ini tetap bisa dibuka oleh program Windows Photo Viewer tanpa masalah.

Kelebihan metode ini yang tampak jelas ada dua: Pertama, tidak seperti metode LSB umum yang besar pesan terbatas oleh besarnya gambar, dalam metode ini ukuran relatif tidak terbatas. Kedua, karena data content gambarnya sendiri tidak berubah, ada tidaknya pesan rahasia pada file ini tidak bisa dideteksi oleh program yang mengecek adanya steganografi berdasarkan PSNR.

Sedangkan kelemahan metode ini adalah, bahwa sebetulnya jika sang kriptanalis mengetahui metode yang kita pakai, maka untuk mengekstraksinya juga sama mudahnya dengan metode LSB. Karena itu, untuk mengurangi kecurigaan, penulis sarankan agar jangan menyisipkan pesan yang besar pada file container yang berukuran kecil, sebab akan tampak ganjil apa bila gambar yang kecil ukuran filenya besar. Sebelum disisipkan, pesan juga sebaiknya dienkripsi dan dipenuhi data dummy agar terlihat sebagai data sampah.

IV. KESIMPULAN

Teknik steganografi dengan mengeksploitasi spesifikasi format container media masih jarang dibahas, walau teknik ini memilki beberapa kelebihan dibandingkan teknik biasa. Karena itu sebaiknya lebih banyak penelitian dan pembahasan mengenai teknik ini.

Penulis telah mengeksplorasi enam jenis format media container yang populer, dan memproposalkan metode steganografi yang dapat diterapkan. Penulis juga menuji salah satu proposal metodenya yaitu steganografi di file BMP.

DAFTAR PUSTAKA

[1] http://en.wikipedia.org/wiki/Container_format(digital)
[2] http://en.wikipedia.org/wiki/BMP_file_format
[3] PNG (Portable Network Graphics) Specification, Version 1.2
[4] http://www.mp3-tech.org/programmer/*frame*_header.html.
[5] http://www.mp3-converter.com/mp3codec/*frame*s.htm.
[6] http://matroska.org/technical/diagram/index.html
[7] http://matroska.org/technical/specs/index.html.
[8] Munir, Rinaldi. 2004. Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.
[9] http://atomicparsley.sourceforge.net/mpeg-4files.html
[10] http://keyj.s2000.at/?p=458
[11] Noé, Alexander. 2009. Matroska File Format. *http://www.matroska.org/files/matroska.pdf*
[12] Adobe Flash Video File Format Specification Version 10.1 http://www.adobe.com/devnet/f4v.html

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, 22 Maret 2011

Andika Pratama / 13507005