Eksperimen Steganografi Audio Pada Openpuff Yang Dikirim Melalui Pesan Instan

Ayubi Wirara
Lembaga Sandi Negara
Jakarta Selatan, Indonesia
lionelkun@gmail.com

***Abstrak*—Kirim terima pesan secara *real time* bukan lagi menjadi suatu hal yang sulit dilakukan untuk saat ini. Kirim terima informasi bukan hanya sekedar chat teks biasa melainkan sudah dapat mengirim dalam bentuk file. File audio merupakan salah satu file yang dapat ditransmisikan dan menjadi salah satu media yang telah dikembangkan untuk menyembunyikan pesan. Salah satu aplikasi Steganografi yang mendukung penyisipan pada file audio (dalam format wave, MP3, AIFF, dan NEXT/SUN) adalah OpenPuff. Hasil Stego Audio akan dieksperimenkan untuk melihat file yang dapat bertahan dan dapat diekstrak saat file sudah ditransmisikan melalui aplikasi pengiriman pesan instan yang sedang menjadi *trend* saat ini (WhatsApp dan Telegram).**

Kata kunci—Steganografi; Pesan Instan; Openpuff; WhatsApp; Telegram

I. PENDAHULUAN

Steganografi yang dalam alih bahasa yunani berarti tulisan rahasia telah banyak digunakan dalam berbagai bentuk dan metode selama kurang lebih 2500 tahun [1]. Steganografi telah banyak digunakan dalam bidang militer, diplomasi, dan lainnya yang bertujuan untuk menyembunyikan pesan rahasia dalam sebuah objek/media, dimana orang lain tidak akan jelas melihat adanya pesan tersembunyi. Seiring dengan perkembangan teknologi, stegano klasik mulai beralih ke area dunia digital. Gambar, audio, video, dan bentuk file lainnya saat ini telah dikembangkan menjadi media/objek untuk tempat menyembunyikan pesan. Tragedi serangan 11 September 2001 mengindikasikan steganografi kembali digunakan dalam melakukan komunikasi rahasia. Pemerintah dan para peneliti dari U.S. menyatakan grup teroris menyembunyikan peta dan foto target teroris serta instruksi untuk aktifitas teroris pada *chat room* olahraga, *bulletin pornografi* dan *website*.

Sekarang ini sudah banyak aplikasi steganografi dikembangkan guna mendukung beragamnya media yang dapat digunakan sebagai *cover* serta dapat menyembunyikan beragam jenis data. Hasil stego yang mengandung informasi tersembunyi secara virtual akan identik dengan *cover* yang tidak mengandung informasi tersembunyi dan tidak nampak adanya perbedaan. Hal ini dikarenakan steganografi pada dasarnya mengeksploitasi persepsi manusia dan manusia tidak dilatih untuk dapat membedakan hal tersebut [2]. Salah satu media yang dikembangkan untuk menjadi *cover* dan akan digunakan dalam percobaan pada paper ini adalah file audio. Terdapat beberapa teknik penyisipan dengan media audio yang dikembangkan saat ini dengan segala keunggulan dan kelemahannya, beberapa diantaranya telah dipaparkan Jayaram,dkk [2] yaitu: LSB *coding*, *Phase coding*, *Phase coding*, *Spread Spectrum*, dan *echo coding*. Beberapa faktor dapat dijadikan dasar untuk membandingkan teknik steganografi audio dalam menyembunyikan data. Anirudh, dkk merangkumnya seperti tabel dibawah ini [8].

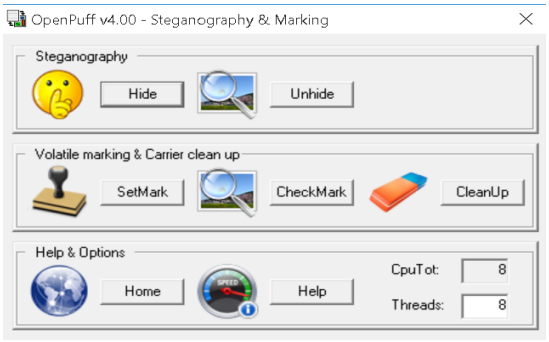
TABEL I. PERBANDINGAN METODE STEGANOGRAFI AUDIO

Metode	Teknik Penyembunyian	Kekuatan	Kelemahan
LSB	Setiap sample pada audio diganti oleh satu bit data tersembunyi	Sederhana dan mudah	Mudah untuk diekstraksi
Phase Coding	Memodulasi phase pada sinyal *cover*	Tahan melawan operasi pemrosesan sinyal	Kapasitas rendah
Spread Spectrum	Menyebarkan informasi ke semua sinyal frekuensi	Menyediakan ketahanan dan peningkatan tranparansi yang lebih baik	Lemah terhadap modifikasi *time scale* dan menggunakan lebih banyak *bandwidth*
Echo Coding	Menyembunyikan informasi dengan memasukan echo ke dalam sinyal *cover*	Menghindari permasalahan dengan *adaptive noise*	Keamanan data rendah dan kapasitas rendah

Semakin cepatnya perkembangan di dunia internet mengakibatkan semakin banyak pula aplikasi kirim terima pesan melalui jaringan internet atau yang disebut dengan instant messaging. Bukan hanya teks, bahkan telah banyak aplikasi yang mendukung kirim terima beragam file. Oleh karena itu, setelah melakukan penyisipan pada file audio, percobaan dilanjutkan dengan mengirim file stego audio melalui aplikasi instant messenger. Beberapa sifat yang harus dipenuhi saat mengirim informasi menggunakan metode steganografi adalah jumlah data rahasia yang dapat dikirim per satuan waktu (bandwidth steganografi), tidak terdeteksi sebagai steganogram, dan perubahan steganogram tidak menghancurkan data rahasia yang disispkan (robustness) [9]. Oleh karena itu, pada paper ini akan diketahui file audio seperti apakah yang paling memungkinkan dan dapat bertahan saat dilakukan kirim terima file stego pada aplikasi pesan instan.

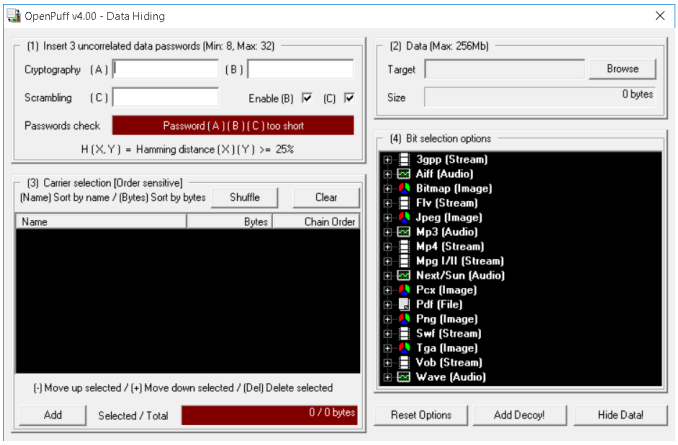
II. OPENPUFF

Aplikasi yang dipublikasikan secara gratis oleh EmbeddedSW pada tahun 2015 ini merupakan aplikasi penyembunyian data dan watermarking dengan menggunakan berbagai media sebagai *cover* penyisisipan. OpenPuff mengimplementasikan system kriptografi CSPRNG sebagai pembangkit bilangan acak dan AES-256 sebagai fungsi enkripsinya [3]. Pada prosesnya, data dibagi diantara beberapa *cover* media (yang support pada aplikasi ini) dan penyisipan bisa maksimal hingga 256 Mb.



Gambar 1. Tampilan awal openpuff

Aplikasi Openpuff merupakan aplikasi berbasis GUI yang cukup mudah digunakan dan dapat dijalankan di semua Operating System Windows. Openpuff mendukung untuk dilakukannya penyisipan dengan beberapa format media. Untuk media *cover* berupa file audio, aplikasi Openpuff mendukung empat format audio,yaitu Audio Interchange File Format (AIFF) dengan ekstensi file *.aif, file MP3 dengan ekstensi *.mp3, file NEXT/SUN dengan ekstensi *.AU/*.SND, dan file wave dengan ekstensi *.wav/*.wave.



Gambar 2. Tampilan penyembunyian pesan pada Openpuff

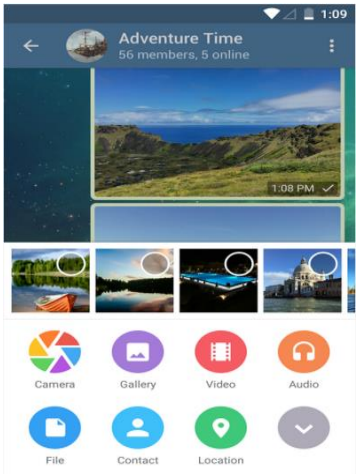
III. APLIKASI PENGIRIMAN PESAN INSTAN

Terdapat banyak aplikasi pengiriman pesan instan yang umum digunakan oleh banyak orang untuk mengirim pesan instan. Pesan instan merupakan jenis percakapan online secara *real-time* yang ditansmisikan melalui jaringan Internet [4]. Saat ini aplikasi pengiriman pesan instan tidak hanya mampu melakukan chat berupa teks saja akan tetapi sudah dapat melakukan kirim terima file termasuk mendukung untuk kirim terima file audio. Aplikasi yang mendukung hal tersebut dan digunakan pada eksperimen dalam paper ini adalah Whatsapp dan Telegram.

WhatsApp merupakan aplikasi yang dikembangkan oleh WhatsApp Inc pada tahun 2009 dan dapat dijalankan pada beberapa dan lintas platform (iphone, andoid, blackberry, dan symbian) [5]. WhatsApp merupakan aplikasi yang mengizinkan seseorang untuk melakukan pertukaran pesan (termasuk chat, grup chat, gambar, video, audio, dan file) dan melakukan panggilan WhatsApp ke seluruh dunia [6]. WhatsApp yang digunakan dan terinstall pada eksperimen ini adalah WhatsApp versi 2.17.107.

Gambar 3. Fitur *attach* pada Whatsapp

Sementara itu, Telegram merupakan alternatif terbaru dari WhatsApp, yang pertama kali diluncurkan pada pertengahan agustus 2013 [7]. Hal utama yang terbaru adalah mengkombinasikan dari sisi *user-friendly* dengan keamanan di level tertinggi daripada kompetitornya tersebut. Sama halnya dengan Whatsapp, Telegram mengizinkan pertukaran pesan tidak hanya dalam bentuk chat tapi juga mendukung kirim terima file gambar, video, audio dan dokumen. Aplikasi Telegram yang diinstall dan digunakan pada eksperimen ini adalah Telegram v3.18.1.



Gambar 4. Fitur *attach* pada Telegram

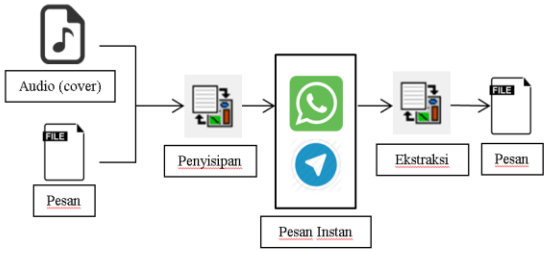
IV. HASIL EKSPERIMEN

Pada paper ini dilakukan eksperimen dengan menggunakan semua format file audio yang didukung untuk dilakukan penyisipan dengan aplikasi Openpuff. Oleh karena itu disiapkan empat jenis format ekstensi file (*.wav, *.MP3, *.aif, dan *.au). Masing-masing media *cover* selanjutnya disisipkan pesan yang sama. Pesan yang akan disisipkan tersebut berupa file *.txt berukuran 505 bytes.

TABEL II. FILE AUDIO YANG AKAN DISISIPI

File ekstensi	Ukuran file	Kapasitas *cover* yang digunakan (Sesuai Ukuran Openpuff)
*.wav	188 Kb	High (25%)
*.MP3	501 Kb	Very High (33%)
*.aif	6290 Kb	High (25%)
*.au	92 Kb	Very High (33%)

Setelah berhasil dilakukan penyisipan, ke empat file stego selanjutnya dikirim dengan menggunakan fitur pengiriman file audio pada aplikasi WhatsApp dan Telegram. Platform yang digunakan pada eksperimen ini menggunakan android versi 4.2.2 Jelly Bean. Setelah file stego berhasil diterima, dilakukan ekstraksi untuk memperoleh kembali isi pesan yang disisipkan tersebut. Berikut adalah mekanisme eksperimen yang dilakukan pada paper ini:



Gambar 5. Mekanisme eksperimen yang dilakukan

Berikut adalah hasil eksperimen yang dilakukan pada masing-masing aplikasi pesan instan:

A. *WhatsApp*

File audio yang dapat dikirim terimakan dan pesan yang disisipi berhasil diekstraksi melalui WhatsApp adalah file audio dengan format MP3. Sementara file dengan format wav dapat dikirim terimakan melalui WhatsApp akan tetapi file tersebut akan terkompresi menjadi format AAC (*Advanced Audio Coding*) dengan ekstensi *.aac sehingga menyebabkan file akan rusak dan gagal terekstraksi.

Hasil tersebut menunjukkan WhatsApp melakukan pemilihan kompresi berdasarkan format ekstensi dari file audio yang mau dikirim. Seperti yang diketahui file wave merupakan file audio yang menggunakan teknik *Pulse-Code Modulation* (PCM) yang tidak dikompres sehingga diasumsikan ukuran file yang akan ditransfer berukuran besar meskipun file yang dikirim tidak berukuran besar. Hasilnya file wave akan terkompresi menjadi file berformat AAC yang bersifat *lossy compression* yang berarti file tidak dapat kembali ke file sebelumnya setelah dikompresi. Sementara untuk MP3 merupakan file audio yang sudah menggunakan *lossy compression*, sehingga WhatsApp mengirim tanpa adanya proses kompresi lagi. Hasil eksperimen pada WhatsApp dapat dilihat pada tabel 3 dibawah ini.

TABEL III. HASIL IMPLEMENTASI FILE STEGO PADA WHATSAPP

Jenis File	Ukuran File	WhatsApp	
		Sent	***Ekstraksi***
*.wav	188kb	file terkompresi menjadi format *.aac	gagal
*.MP3	501kb	berhasil	berhasil
*.aif	6290kb	gagal	gagal
*.au	92kb	gagal	gagal

Sementara untuk file berformat *.aif dan *.au, file gagal dikirim terimakan karena WhatsApp tidak mendukung untuk format file tersebut yang mungkin disebabkan handphone android tidak mengenalnya sebagai file audio. Meskipun selanjutnya dicoba kirim terimakan menggunakan fitur document akan tetapi file tetap tidak dapat terdeteksi dan dikirim.

B. *Telegram*

Hasil implementasi pada aplikasi Telegram pada android diketahui format file yang didukung dalam menggunakan fitur audio adalah wave dan MP3. Tidak ada perubahan ekstensi/terjadi proses kompresi saat file dikirimterimakan sehingga file Stego audio berhasil diekstraksi. Akan tetapi file audio yang diterima/downlad masuk ke dalam file Telegram Documents bukan pada file Telegram Audio. Sementara untuk file dengan ekstensi *.aif dan *.au, Telegram tidak mendukung kirim terima file dalam bentuk audio tersebut hal ini mungkin dikarenakan android itu sendiri tidak mengenali file tersebut sebagai file audio, akan tetapi masih dapat dikirimterimakan melalui fitur pengiriman dokumen dan sama halnya dengan wave dan MP3 file tersimpan di file Telegram Documents serta file dapat diekstraksi. Hasil eksperimen pada Telegram dapat dilihat pada tabel 4.

TABEL IV. HASIL IMPLEMENTASI FILE STEGO PADA TELEGRAM

Jenis File	Ukuran File	Telegram	
		Sent	***Ekstraksi***
*.wav	188kb	berhasil	berhasil
*.MP3	501kb	berhasil	berhasil
*.aif	6290kb	berhasil**	berhasil
*.au	92kb	berhasil**	berhasil

** file dikirim melalui fitur Document bukan melalui Audio

V. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Hasil dari eksperimen yang dilakukan dapat diambil kesimpulan bahwa tidak semua file stego audio Openpuff dapat bertahan dan bisa diekstrak kembali saat file stego audio dikirim melalui aplikasi pesan instan. Hanya file MP3 yang berhasil dengan baik saat diimplementasikan pada WhatsApp dan Telegram. File wave berhasil hingga dapat diekstraksi

kembali hanya pada aplikasi Telegram. Sementara untuk file format AIFF dan NEXT/SUN berhasil pada aplikasi Telegram hanya saja pada saat pengiriman tidak menggunakan fitur pengiriman audio melainkan fitur pengiriman dokumen.

B. Saran

Beberapa saran yang selanjutnya perlu dilakukan penelitian lebih lanjut adalah sebagai berikut:

1. Eksperimen dilanjutkan dengan menggunakan platform yang berbeda dan menggunakan *tools* Steganografi lainnya yang mendukung penyembunyian pada media audio.
2. Perlu dilakukan penelitian lebih lanjut mungkin seperti dilakukan reverse engineering pada aplikasi Telegram untuk mengetahui mengapa file audio yang dikirim tersimpan pada file dokumen.

DAFTAR PUSTAKA

[1] James C. Judge, ”Steganography: Past, Present, Future”, 2001.

[2] Jayaram P., Ranagantha HR., dan Anupama HS, “Information Hiding Using Audio Steganography – A Survey”. The International Journal of Multimedia & Its Aplications (IJMA), 2011.

[3] EmbeddedSW, “OpenPuff v4.00 Steganography & Watermarking”, 2015.

[4] Anonim, “Instant Message”, 22 januari 2017, Diperoleh pada 30 April 2017, dari https://id.wikipedia.org/wiki/Pesan_instan.

[5] Anonim, “WhatsApp”, 3 Mei 2017, Diperoleh pada 17 Mei 2017, dari https://id.wikipedia.org/wiki/WhatsApp.

[6] WhatsApp Inc, “WhatsApp Encryption Overview”, 2016.

[7] Jesus Diaz Vico, “Telegram Bypassing the authentication protocol”. Instituto Nacional de Tecnologias de la Comunicacion, 2014.

[8] Anirudh Pachori, Adya, Sanjay Mange, dan Harshad Bhanushali, “Steganography Techniques – Data Security Using Audio, Video, and Image”, International Journal of Emerging Technology and Computer Science, vol.2, 2 june 2017.

[9] Rohith V, Yathiraj GR, Akshaya George, Aswathi M V, Athul Jithendran, dan Navanya M A, “Technology for Hiding Information using Steganography in Smartphones”, International Journal of Innovative Research in Computer and Communication Engineering, vol.4, Issue 5, May 2016.