Embry-Riddle Aeronautical University: CS426 - Digital Forensics
Troy Neubauer
Juan Ortiz Couder
March 7th, 2024

For my Digital Forensics final project I propose a mix of options 1C (steganography in images), option 2 (my own project), and option 3 (research). I am interested in creating a library prototype which utilizes steganography to hide data inside an AV1 video stream. Some theoretical approaches for hiding data inside AV1 exist [2], however there are currently no open source implementations. I propose implementing the approach discussed in “Introducing av1 codec-level video steganography”[2] in Rust and open sourcing the result.

Additionally, steganography in images (or in my case, video) provides a one way mechanism for transmitting hidden information. Assuming that parties want to encrypt their hidden communication, and pre-shared keys are not viable, it is desirable to perform a hidden key exchange. I believe video streaming control protocols like RTSP (a container format for live streaming video between devices and servers) could be used to hide this key exchange since bidirectional communication is allowed. With RTSP, clients can send control messages to the server to start/stop streaming, change video parameters, etc. I believe it would be possible to perform a hidden key exchange between the client and the server, and use the newly shared symmetric key to encrypt the data being sent to the client. This is a stretch goal and depends on having a working implementation of one way steganography inside AV1.

Work Plan:

I will start by implementing a minimal version of the steganography encoding approach in [2]. This will be followed by an implementation of the decoder discussed in the paper, which I will use to verify the retrieval of the hidden data. Then, I will perform analysis centered around:

1. How hidden is the data? Are there certain obvious indicators?
2. Is the video visually affected?
3. How much information can we store in the video stream as a percentage of the host video’s bitrate before 1 and 2 are affected?

Assuming I still have time, I will then experiment with RTSP (using the excellent rtsp-types Rust crate) to implement a hidden key exchange. If this goes well, I will add encryption to the hidden message, with the client/server using the shared symmetric key to decrypt/encrypt the steganographic payload.

References:

[1] Y. Liu, S. Liu, Y. Wang, H. Zhao, and S. Liu, “Video steganography: A review,” Neurocomputing, vol. 335, pp. 238–250, Mar. 2019, doi: 10.1016/j.neucom.2018.09.091.

Available: https://www.sciencedirect.com/science/article/pii/S0925231218312608. [Accessed: Mar. 07, 2024]

[2] L. Catania, D. Allegra, O. Giudice, F. Stanco, and S. Battiato, “Introducing av1 codec-level video steganography,” in Image Analysis and Processing – ICIAP 2022, S. Sclaroff, C. Distante, M. Leo, G. M. Farinella, and F. Tombari, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022, pp. 284–294. doi: 10.1007/978-3-031-06427-2_24