Introduction

Every day, we encounter encryption. Whether it be securing the information on your mobile phone or computer or making sure no one sees the information you send to a website when you log in; you rely on encryption to protect your data and identity [1]. Steganography, the act of hiding a piece of data within another piece of data [2], is used when a user wants to add an additional level of protection to the data they're transmitting. Steganography is very powerful because it's incredibly challenging to detect using traditional means, however, it is not thought to be widely used [3].

Advanced Encryption Standard

The Advanced Encryption Standard (AES) is one of the most widely used encryption method and was introduced by the National Institute of Standards and Technology (NIST) in 2001 as a part of the Federal Information Processing Standards (FIPS) publication 197 [1]. AES is a type of cipher known as a block cipher. In a block cipher the blocks of bits are encrypted separately. For example, with 128-bit AES, each 128-bit[1] chunk of data (128 1s and 0s) is encrypted to output 128 bits of encrypted data, and then they're all stitched together [1].

AES works by manipulating the input data in rounds. The number of rounds is determined by the type of AES being used. In each round one of several transformations can be used. The transformations are `AddRoundKey`, `SubBytes`, `ShiftRows`, `MixColumns`, and `Key Expansion`. [4]

Each of the 128-bit blocks are treated as a 4x4 matrix where each of the 16 elements is a single byte[2]. By treating it as a matrix, the data can be manipulated in rows and columns. [4]

`SubBytes`, `ShiftRows`, and `MixColumns` all manipulate bytes directly. `SubBytes` looks up each byte in the Rijndael S-box (see Appendix A) and replaces it with its pair. `ShiftRows` and `MixColumns` both manipulate the rows and columns of the matrix based on pre-defined algorithms in FIPS 197. [4]

`Key Expansion` uses the provided key to create $n + 1$ keys where n is the number of rounds; one to be used per round.

`AddRoundKey` uses the key generated for the round and manipulates each block using the `XOR`[3] operand. This transformation occurs before `SubByte`, `ShiftRows`, or `MixColums` in each round. [4]

[1] A bit is the smallest piece of data a computer can store. Represented in a 1 or 0. Bits are stored on magnetic hard drives as magnetic polarities, there are other methods of communicating a bit on physical hardware as well.
[2] There are 8 bits in a byte
[3] XOR known as exclusive or, is an operand used in computing that only returns true when both inputs are not the same.

Image Files

Most bitmap images are one of four types, 8-bit colour, 8-bit grayscale, 24-bit RGB, or 46-bit RGB [5]. The number of bits in the type represents the number of bits per pixel. For the purposes of this project, 8-bit colour will be ignored as the index may result in steganography being too obvious if not managed correctly. However, the remaining three types all function fundamentally the same.

In an 8-bit grayscale image, each pixel consists of 8 bites, or 8 1s and 0s, mapped to a single colour channel, shades of grey. A 24-bit RGB[4] image will have 3 bytes mapped per pixel or 24 1s and 0s. Each byte will be mapped to a colour channel: red, green, or blue. In a 46-bit RGB image, each pixel gets allocated 6 bytes, two per colour channel [5].

There is a third type of colour image commonly used for printing, as it uses the CMYK[5] standard. It works the same way as 24-bit RGB. However, instead of using 24 bits it uses 32 (4 bytes), 1 byte per colour channel: cyan, magenta, yellow, and black [5].

Audio Files

Audio files function very similarly to image files. However, instead of colour channels, there are samples. Each sample is a sound a specific to a moment in time. For example, an audio file with a sample rate of 64 kbps[6] and a bit-depth[7] of 8 bits would have 8,000 samples per second [6]. To calculate the number of samples per second, divide the bit-rate by the bit-depth.

Analogue Audio

Humans can hear sounds between approximately 20Hz and 20kHz [7]. However, most microphones can exceed that and detect sound with frequencies as high as 24kHz [8]. Because of the additional 4kHz that microphones can detect, it is technically possible to transmit data in those frequency ranges.

MATLAB Project

The project will use a mix of steganography and the Advanced Encryption Standard to create a secure method a transferring highly sensitive data. The project will support the most basic of the AES standards, 128-bit blocks. For the application of steganography, the project will support image, and data-based audio steganography.

Extension

As an extension to the encryption part of the project; the project should also implement the AES 192-bit and 256-bit standards and potentially other algorithms. As an extension to the steganography, humans are unable to hear sound above 20kHz, utilizing the fact that computers and microphones can process signals at frequencies greater than 20kHz, converting the digital stream of binary data to an analogue signal and then processing the analogue signal after transmission.

[4] RGB stands for red, green and blue. RGB is how screens emit light and show you nearly any colour.
[5] CMYK stands for cyan, magenta, yellow, and black. These colours are used in printing to make nearly any colour.
[6] kbps stands for kilobits per second or thousand bits per second.
[7] bit-depth is the number of bits allocated to a single sample

Approach

The first step of the project is to implement AES in MATLAB. Implementing the standard will require a strong understanding of the algorithm as well as MATLAB. After AES is implemented, a method for imbedding the encrypted data into image files will need to be determined. Most notably is the spacing between pixels that are changed. If the imbedded pixels are too close together, the alterations to the file will become noticeable to the human eye. Additionally, if the pixels line up vertically or horizontally in the image, streaks will become noticeable, so the location of altered pixels will need to appear random.

After image steganography is implemented, audio steganography will need to be as well. Audio steganography will follow the same principles as image steganography, with one key difference, the number of original samples there are between each imbedded sample in order for the audio alternations not to be noticed. High bitrate and high bit-depth audio should be good for steganography.

Finally, in order to implement an analogue method of transporting data, it is necessary to determine what frequencies to use, as well as what amplitudes will be used to symbolize a bit, and how the data itself will be transferred. Additionally, whether the data being imbedded should be a continuous stream of bits, or a stream of bytes where different frequencies determine different bits within a byte, needs to be determined.

References

[1] J. C. Villanueva, “What AES Encryption Is And How It's Used To Secure File Transfers,” 19 May 2015. [Online]. Available: www.jscape.com/blog/aes-encryption. [Accessed 14 February 2018].

[2] Merriam-Webster, “Steganography,” [Online]. Available: www.merriam-webster.com/dictionary/steganography. [Accessed 14 February 2018].

[3] D. Radcliff, “Steganography: Hidden Data,” 10 June 2002. [Online]. Available: www.computerworld.com/article/2576708/security0/steganography--hidden-data.html. [Accessed 14 February 2018].

[4] National Institute of Standards and Technology, *Federal Information Processing Standards Publication 197,* Washington DC: United States Department of Comerce, 2001.

[5] W. Fulton, “Color Bit-Depth, & Memory Cost of Images,” [Online]. Available: www.scantips.com/basics1d.html. [Accessed 14 February 2018].

[6] BBC, “GCSE Computer Science - Encoding Audio and Video - Revision 1,” [Online]. Available: www.bbc.co.uk/education/guides/z7vc7ty/revision/. [Accessed 14 February 2018].

[7] C. D'Ambrose , “Frequency Range Of Human Hearing,” 2003. [Online]. Available: https://hypertextbook.com/facts/2003/ChrisDAmbrose.shtml. [Accessed 14 February 2018].

[8] Curiosity, “Sounds That Mics Can Hear And Humans Can't Could Protect You From Electronic Eavesdropping,” 30 July 2017. [Online]. Available: https://curiosity.com/topics/sounds-that-mics-can-hear-and-humans-cant-could-protect-you-from-electronic-eavesdropping-curiosity/. [Accessed 14 February 2018].

Appendix

Appendix A

Rijndael S-box in human readable hexadecimal form [4]

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16