Selective Encryption of JPEG Standard Baseline Compression Images

Tom Lookabaugh, Andrew T. Pouzeshi, Geoffrey L. Griffith, Joe B. Jarchow, Joseph Z. Kadhim, Shinya Daigaku
Department of Computer Science, University of Colorado, Campus Box 530, Boulder, CO, USA 80309-0530

Abstract

One of the ramifications of compressing a file is that vital data are localized in small, specific areas. Consequently, it is easy to exploit this property of compression to provide a high level of security as selective encryption focuses on encrypting only these vital portions of data to render a file unusable. Selective encryption results in a large savings in computationally intensive operations, while maintaining a reliable level of security. There have been a number of selective encryption methods proposed for the JPEG compressed image format. This paper describes a simple, yet secure method for selectively encrypting JPEG images that are compressed using the Baseline[1] standard. JPEG Selective Encryption has a high value for any application in which sensitive images may be at risk, from low power satellite imaging systems to securely transmitting images across the Internet.

Keywords: JPEG, image encryption, encryption, selective encryption, partial encryption, cryptography, cryptanalysis, compression, security.

1. Introduction

Selective Encryption is defined as applying encryption to a portion of a file's bit-stream with the assumption that the entire file will become useless without the proper decryptor. The attractiveness of selective encryption arises from the idea that a file can be securely encrypted and transmitted without spending the computational effort of encrypting the entire file. Selective encryption techniques range from encrypting a portion of the file, say a straight percentage of the data, to others that encrypt specific vital sections of a file. Selective encryption methods are never as secure as encrypting an entire file, because much of the data is not encrypted. The goal of selective encryption is to reduce the computational time of encryption, while maintaining a sufficient level of file protection.

The increase of multimedia applications and the transmission of data over public networks necessitate efficient methods of securing transmitted data. Because of the large size of multimedia files, Selective Encryption methods have been devised for various different types of multimedia compressions. The increasing of use of JPEG image encoding software and hardware and transmission across large public networks warrants a strong, yet simple, Selective Encryption scheme for JPEG images. The goal of the research behind this paper was the development of a simple yet secure method of Selective Encryption for the JPEG Baseline compression standard. Such a method would be applicable in situations ranging from encrypting transmitted satellite imagery to encrypting images generated by digital cameras to protecting images for transmission across the Internet.

---

[1] As defined by Pennebaker and Mitchell in "JPEG Still Image Data Compression Standard."

2. JPEG Image Selective Encryption Criteria

The goal of Selective Encryption for JPEG images is to minimize the amount of encryption applied to a file while maximizing the damage done to the image. As a bonus, this paper will define a method that is relatively fixed in size and will not require the amount of encryption to increase linearly as the image size increases linearly. Most of the file will remain unencrypted, which allows the retrieval of those data. However, the data that remains unencrypted will be useless without the encrypted data and the encrypted data will be reasonably difficult for a hacker to replace, reconstruct or calculate. The result will be a JPEG file structure, partially encrypted, that is impossible to use without the proper decryptor and key. The focus of this paper is to present the research and algorithm developed for selective encryption of standard Baseline compressed JPEG image files.

3. Goals and Criteria for Selective Encryption

Selective encryption can be measured in several different ways and optimized for many different purposes. Confusion may arise from reading literature about selective encryption, as the method is usually specific to the type of file being encrypted. Thus, this confusion can be avoided by having a clear idea of our selective encryption criteria. The criteria used for selective encryption include:

Security Criterion:

Selective encryption has been proposed for a number of different user scenarios. For the purposes of this paper, we will define the security criterion as encryption of data sufficient to render the image unusable to a standard JPEG image decoder. The data must be vital enough to render the image unusable to an attacker's reconstruction or replacement of data to the point that the attacker would be forced to use an expensive brute force method to decode the image. Although the attacker will still be able to retrieve most of the data from the selectively encrypted image file, the image itself cannot be easily reconstructed.

Security Validation:

Security of any given encryption can be validated in a number of different ways. Some researchers validate security by choosing the criteria and then feeding the selectively encrypted data into a standard decoder and observing resulting reconstructions. Other researchers take a cryptanalytic approach by acting as an attacker and working with a modified decoder and other available information to design a method of defeating the selective encryption. Still others make mathematical calculations, such as RMS (root mean squared) or PSNR (peak signal-to-noise ratio), to find differences between the encrypted and unencrypted data values. This paper considers all three of these methods as valid and all three have been considered for this cryptosystem.

Complexity:

Often encryption can be complex and computationally expensive. The primary goal of selective encryption is to reduce the percentage of data that needs to be encrypted, while maintaining an acceptable level of security. This reduction of encryption operations must

be weighed against increased operations necessary to implement the selective encryption algorithm. If computing the data to encrypt and/or searching for those data is more expensive than simply encrypting the entire file, then the selective encryption system should not be considered as valid.

Compression Efficiency:

The primary goal of compression is to store a set of data in less space than the data representation requires by utilizing specialized algorithms. Image, video, and audio compression formats often exploit the fact that a human detects only a portion of the overall sensory input. Therefore, certain compression formats further reduce the number of bits needed to represent the data by approximating the data values such that the difference is relatively undetectable to human sense. Note that in these cases, the exact data are lost, but the compression efficiency will be much greater. However, the quality of the media is degraded exponentially by increasing the overall compression. For this reason, compressors often allow for varying degrees of data loss.

Some methods of selective encryption compromise compression efficiency by adding data overhead and/or by modifying the compression algorithm, causing a penalty in performance. For example, constantly searching a data stream for information about where to encrypt will add computational overhead. In addition, certain encryption algorithms greatly increase the size of the data, which is contrary to compression. Although there are newer encryptions that do not increase data size, circumstances may require use of less size efficient encryption algorithms. Any degradation in compression efficiency must be weighed against the constraints surrounding the particular need for selective encryption.

Interaction with Compressors:

There are methods of selective encryption that work with, and others that are independent, of the compression algorithm. It is important to be aware that there are potentially major differences in both performance and compression efficiency between these two methods. Ideally, the selective encryption algorithm would be implemented within the compression algorithm. This will minimize file parsing operations and reduce the overall number of operations needed to find the portion of data to encrypt.

Selective Encryption Attacks:

One final item to define and consider are the types of attacks on a selective encryption system. There is a clear difference between cracking a particular selective encryption system and cracking an encryption algorithm. If the encryption algorithm used to implement the selective encryption system is breakable, or found to be breakable in the future, we must assume the selective encryption system is invalidated and we must switch to a more secure encryption algorithm. For the purpose of this paper, we will assume the particular encryption algorithm used to implement JPEG selective encryption system is secure and we will discuss only attacks that pertain to selective encryption and not the various attacks on different encryption algorithms.

There are really three ways of attacking a selective encryption system. The first, and most well known, is a purely brute force attack. Since selective encryption systems only encrypt a portion of the file, usually the minimum possible to sufficiently protect the data, it will take much less time to either defeat the encrypted data or remove the encrypted data and try a systematic bit replacement of all possible values. For JPEG selective encryption, an attacker would work with a standard JPEG image decoder and would run each permutation of the replaced data through the decoder to try to rule out the most obvious, unviewable images. Then, once the automated process has produced a number of images that are viewable, the attacker would have to look at each one to find the correct, or at least understandable image. However, this will potentially take a large amount of time, depending on how many bits are encrypted. So, we will define the brute force attack as the most undesirable method.

The second method of attack is defined as a reconstruction attack. An expert in the particular file format that is selectively encrypted could devise a method for reconstructing the vital data that have been encrypted, given the unencrypted data in the file. In the case of JPEG selective encryption, the attacker would work with a modified JPEG image decoder that would compute the correct information, given the selectively encrypted file. Fortunately, the JPEG compression standards (along with many other compression standards) are designed specifically to decompose the original image into its vital components that allow the decoder to calculate each pixel value to reduce overall file size. Ironically, it is this reason that makes selective encryption of compression formats very attractive. Still, and in general, it is important to be extremely familiar with the data format of the selectively encrypted file, and measures must be taken to avoid this form of attack.

Finally, the third, and probably most effective, would be a hybrid attack. There are several possibilities for this type of attack, which would consist of doing research on real world instances of the particular file format to try to find consistency of vital data and having some understanding of how the data are structured. This could help the attacker by reducing the amount of "most likely" possibilities of that data. For JPEG selective encryption, this would probably entail a basic understanding of the JPEG image data components and some prior knowledge of a large number of real world instances of JPEG images and commonly used JPEG encoding schemes. Then, it could be determined if the encrypted data could be replaced by trial and error with a relatively small number of sets of real world data to try to reproduce (or even approximate) the original image to an acceptable level. Again, measures must be taken to ensure this type of attack will lead to failure.

4. Previous Selective Encryption Attempts

There is currently a small amount of existing research on the topic of Selective Encryption available for various multimedia formats. Much of the research pertinent to this paper is based on previous MPEG and JPEG Selective Encryption techniques and research. Through out this literature there is much indecision as to which file components are the best target(s) for Selective Encryption. This paper attempts to

evaluate all possible targets and previous attempts of Selective Encryption for JPEG image formats and any possible attacker's counter measures.

In their paper "Selective Encryption of the JPEG2000 Bitstream," Norcen and Uhl [8] outline a selective encryption method for the JPEG2000 compressed file format. The proposed method uses an AES block cipher to encrypt 20% of the visual information in JPEG2000 files, providing relatively secure file transmission without the computational costs of encrypting the entire file. While this method is more efficient than encrypting the entire file, the algorithm fails to exploit the relationship between compression and isolation of vital data. Moreover, the amount of data that is encrypted in the file increases linearly as the JPEG file size increases. Ideally, the amount of data encrypted would be relatively fixed and would include only vital components that would render the image unusable.

By carefully selecting vital components of the file to encrypt, it is possible to provide security while encrypting an even smaller, and ideally fixed, portion of the file. Several other research papers (mostly concerning MPEG Selective Encryption) suggest targeting the DCT (Discrete Cosine Transform) Quantizer tables found in many compressed multimedia file formats, including JPEG formats. The DCT is a mathematical technique used for decomposing wavelengths into elementary frequency components. For a JPEG image, these coefficients are stored in the Quantizer table. Encrypting the Quantizer tables are an attractive target because there is no variance in table size and the number of tables allowed is small, yet the minimum amount of Quantizer data is not so small that it could easily be permuted or guessed. Each Quantizer table must be exactly 64 bytes, and there are no less than 1 and no more than 4 allowed. Thus, there is a minimum of 2^{512} possibilities and up to 2^{2048} possibilities to guess the exact Quantizer table(s) encrypted. This target is also large enough that a non-intelligent brute force attack of simply substituting values for these tables would take a considerable amount of time to reproduce the original image. Even though the Quantizer table looks promising at first glance, it proves to be an extremely weak target for JPEG selective encryption, as we'll see in section 5 of this paper.

In their paper "Secure Compression using Adaptive Huffman Encoding," Kailasanathan, Naini, and Ogunbona [4] propose the Huffman encoding tables, found in the Baseline JPEG format, as a viable target for selective encryption. This selective encryption algorithm offers two possible solutions. The first involves removing the compression tables from the image, securely transmitting the tables separately, and then reintegrating the tables when received. The second, more appealing solution, is to encrypt the compression table and send it along with the file and then securely transmitting a key to decrypt on the other side. As with the Quantizer tables, the Huffman tables appear to be a good target for selective encryption, because these tables have a relatively small variance in size, yet the minimum size is sufficient to repel brute force attacks. After further research discussed in section 5 of this paper, the Huffman tables prove to be a more valuable a target for selective encryption. Unlike the Quantizer table values, it is not as easy to produce an image by replacing the Huffman values of an optimized JPEG image. However, because many JPEG compression applications use default Huffman

tables, an attacker may have success by trying a series of popular default tables used by the more common graphical editing applications, digital cameras (JPEG encoding chips) or the example tables in the JPEG standard. Still, both Quantizer and Huffman seemed to have potential, and in the end, the research finally yielded a solid solution.

5. Cryptanalytic Approach to JPEG Selective Encryption

To devise an algorithm for selectively encrypting JPEG images effectively, the team researched the feasibility of this project from several different angles. Since there is no universal method for selective encryption, the team thought it appropriate to examine previous research on subject for multiple compression formats, review the JPEG baseline compression standard[2], research of common implementations of the JPEG encoders/decoders, and collect a large sample of real world JPEG images to be used for statistical analysis. By the end, the team was able to devise a method of selective encryption that will sufficiently protect JPEG images against any of the possible attacks mentioned in this paper.

The team began by researching the Baseline standard compression for JPEG images. Although there is a large amount of data included in the format, much of it is not vital to the image, or can be replaced, or even calculated. The team narrowed the possible targets for selective encryption to three pieces: the Encoded Data stream, the Quantizer tables, and the Huffman tables (which coincided with previous research available).

As mentioned above, a previous attempt at the Selective Encryption of JPEG images was to encrypt a percentage of the entire Encoded Data. While this method will definitely work, it was ruled out for two reasons. The first, and the most important reason, is that non-intelligently encrypting a percentage of the Encoded Data fails to exploit the relationship between a compression format and the concentration of vital data. Second, the amount of encryption needed will linearly increase as the size of the file increases. The Encoded Data makes up the largest percentage of the file size (on the order of 96% for JPEG images under 20 KB and 99%+ for files of 200 KB or more). The goal is to have a relatively fixed amount of data that needs to be encrypted and ideally that size will not be dependent upon the image size. Thus, the Encoded Data was ruled out as a viable target.

Another possible target found from both analyzing the JPEG standard and reviewing previous research is the Quantizer tables. There was a considerable amount of selective encryption research available for methods that utilize the Quantizer tables, but much of it was for other compression formats. However, there were at least two research papers on the topic of selective encryption for JPEG images that suggested the Quantizer tables are good targets. With this in mind, the team decided to try working with this Quantizer to see what effect, if any, altering these values had on various images. During the course of the research, over 2500 random JPEG images were gathered from the Internet and over 200 were tested directly. Unfortunately, it was determined that this target was neither vital enough nor unique enough to provide ample security. Altering the DCT coefficients only distorts the resolution, brightness, or color. Even a completely random table would

[2] As defined by Pennebaker and Mitchell in "JPEG Still Image Data Compression Standard."

yield a viewable image of the original only slightly degraded. In many cases, the team was able to reconstruct most images by simply replacing the entire table with a single value for each of the DCT coefficients, allowing the image to decode with a negligible degradation of quality. Although the images were often slightly discolored and/or the resolution was distorted, these images were certainly not damaged enough to render them incomprehensible. For this reason, the Quantizer tables were ruled out as a viable target.

Finally, the team focused on the Huffman (compression) tables as a target for selective encryption. The image was found to be extremely sensitive to minor changes in the Huffman tables, as these tables are used to generate/decode the Encoded Data stream. If even one encoding value is altered, then the resulting image will be considerably damaged. Furthermore, it will be impossible to reconstruct images by replacing Huffman tables with random values or even different Huffman tables from other images. Unlike the encoded data stream, the size of these tables is relatively fixed, as the Baseline standard dictates that there can be a maximum of four of these tables. So, on the surface, and as other research pointed out, the Huffman tables seem to be the most attractive target for JPEG selective encryption. However, it is necessary to look more into JPEG compressors and common instances of JPEG images to validate the security of a selective encryption method that targets the Huffman tables.

There are a wide variety of different JPEG encoders available, such as the IJG[3] JPEG encoding/decoding classes, Adobe Photoshop (a professional image editing application) or even the common Microsoft Paint (included with every copy of Microsoft Windows). While each encoder provides a different level of features, they all work with the JPEG Baseline compression standard. The main differences among these encoders can be measured by how they actually encode the image itself. While some encoders will actually calculate an optimized Huffman table, others use a series of default tables that are pre-calculated. Although these pre-calculated tables reduce computation, they pose a problem to security, because if an attacker had "inside information" on which JPEG encoder was used, they might be able replace the encrypted compression table. Due to the existence of default compression tables, a selective encryption method that only encrypted the Huffman tables would be insecure.

A remedy to solve the problem with default Huffman tables would be to optimize the compression of every JPEG image, before selectively encrypting. However, there are two potential problems with this remedy. First, using the IJG compressor with a flag to optimize images, the team produced approximately 470 optimized JPEG images. These images were randomly collected from the Internet. Even after optimization, there were still a large number of duplicate Huffman tables. Of these non-optimized images, 76.3% contained duplicate Huffman data. After optimizing these same images, 39.6% contained duplicate Huffman data. Thus, even after optimization, a considerable number of duplicate tables still existed, meaning that even if images are optimized, attackers may still be able to replace these values. Secondly, a goal of selective encryption is to reduce the amount of computation necessary to protect the file. However, by optimizing JPEG

[3] The IJG Organization is one of the most common providers of a C++ API for encoding and decoding JPEG images.

images (i.e. not making use of pre-calculated tables), there is an increase in the amount of computation needed to assure security of the image. Moreover, many of the JPEG compression chips used in digital cameras or satellite systems do not have the capability of calculating an optimized table. So, although the Huffman tables seem like the perfect target, they alone do not provide the level of security selective encryption hopes to achieve.

After spending a considerable amount of time researching, it became increasingly apparent that just encrypting one or two frames of data in the image wasn't going to solve all of the problems. The attacker could know at least the size of the table and the number of tables for both the Quantizer and the Huffman tables by counting encrypted frames in the image. Moreover, the Huffman tables have an ordering which greatly reduces the number of possible permutations and the Quantizer tables by themselves are much too weak because even a randomized table will often produce a degraded image, but not damaged enough to make it completely unusable. The team realized that we needed to hide the exact size and number of the compression tables.

To overcome all of these drawbacks, Team ISE devised an algorithm that encrypts not only the compression data frames, but also all the data between the compression tables and the beginning of the Encoded Data stream. The Team ISE algorithm can be implemented in cooperation with compression or independent of compression, as well as in software or in hardware. The algorithm is as follows:

1. Choose a block size of some number of bytes (for example, 32 bytes work well with the AES block cipher encryption system).
2. Write the file as normal until the FFC0 (SOF0 frame) or FFC4 (DHT frame) marker (whichever is written first for the particular encoder).
3. Write this 2 byte marker and then start encrypting in blocks of the pre-chosen block size until the FFDA marker (SOS frame) is to be written.
4. Encrypt the FFDA marker and fill out the rest of the current block and write it to file.
5. Encrypt one final block and write it to file.
6. Write the rest of the Encoded Data stream and file as normal.

This effectively hides the size of the Huffman tables within the file. This causes the encryption to run directly into the Encoded Data stream. Since both the encryption and the encoded data stream appear to be random values, it is now impossible to tell where the Huffman tables end and the Encoded Data begins. Thus we have overcome the problem of direct table replacement. Furthermore, a brute force attack would be extremely expensive, because the average size of these tables for a small image would yield about 2^{2400} possibilities! This leaves only the problem of the Hybrid attack with (1) "inside information" of a compressor that (2) uses pre-calculated or default compression tables that are unchanging. In this case, an attacker could replace the encrypted table and recalculation of the Scan header frame. Any data that was encrypted at the beginning of the Encoded Data stream could be systematically substituted until the correct solution is found. At a minimum, the Hybrid attack method would have (assuming a 32 byte block

size) at least 2^{256} possibilities and at most, there would be at most 2^{512} possibilities. Thus, this particular Hybrid attack would still be very expensive and take quite a bit of time and effort by the attacker. However, the key to overcoming this attack is to use an optimized compression algorithm for the table. Moreover, this cryptosystem encrypts only about 3% of the JPEG image data for a very small image around 20 KB and for the case of a image produced by a digital camera (of about 1 MB in size), this selective encryption algorithm will encrypt only about 0.001% of the file.

6. Conclusion

After researching previous attempts at JPEG selective encryption, we found that although previous researchers were definitely on the right track, there are many weaknesses in the other approaches. The algorithm developed by Team ISE overcomes these weaknesses while adhering to the original goals of selective encryption defined in this paper. The algorithm performs in such way that the number of computational operations needed to encrypt the data does not increase as file size increases. Furthermore, the algorithm is simple enough that it can be easily implemented in both software and hardware, in cooperation or independent of the compressor, thereby lending itself to provide high flexibility for many different applications. The Team ISE selective encryption algorithm will only be vulnerable to a brute force attack. The algorithm defined here has met all of the goals set out in this paper and finally, but most importantly, the algorithm is secure.

Bibliography

1. Chang, H. and Li, X. *On the Application of Image Decomposition to Image Compression and Encryption*. 1996.

2. Chang, H. and Li, X. *Partial Encryption of Compressed Images and Videos*. 2000.

3. Droogenbroek, M. and Benedett, R. *Techniques for Selective Encryption of Uncompressed and Compressed Images*. 2002.

4. Kailasanathan, C. and Naini, R. *Compression Performance of JPEG Encryption Scheme*. 2003.

5. Li, X., Knipe, J. and Cheng, H. *Image Compression and Encryption Using Tree Structures*. 1997.

6. Lookabaugh, T., Sicker, D., Keaton, D., Guoand, W. and Vedula, I. *Security Analysis of Selectively Encrypted MPEG-e Streams*. 2003.

7. Miano, J. *Compressed Image File Formats*. Addison Wesley Longman, Inc., Reading, Massachusetts, 1999.

8. Norcen, R. and Uhl, A. *Selective Encryption of the JPEG2000 Bitstream*. 2003.

9. Pennebaker, W. and Mitchell J. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, New York, 1993.

10. Podesser, M., Schmidt, H. and Uhl, A. *Selective Bitplane Encryption for Secure Transmission of Image Data in Mobile Environments*. 2002.

11. Seo, Y., Kim, D., Yoo, J., Dey, S., Agrawal, A. *Wavelet Domain Image Encryption by Subband Selection and Data Bit Selection*. 2003.