A Survey of Recent Advances in Video Security

Jason Barnes, jason.barnes@wustl.edu (A project report written under the guidance of Prof. Raj Jain)

Download 

Abstract

Video security must be handled in a different way than that of generic data. Traditional security methods can be used to aid in video security, but fall short because of the demanding requirements of streaming video. Methods in encryption, authentication, and steganography have been created to ensure security in video. Many of these methods are created and used in order to prevent theft of copyrighted material.

Keywords

Streaming, video, encryption, steganography, authentication, watermark, fingerprint, signature, encoding, VEA, video encryption algorithm, DCT, discrete cosine transform, AVC, advanced video coding, perception, steganalysis, SVM, support vector machine, distortions, digital rights management, copyright protection

Table of Contents

1 Introduction

Data security is a necessary part of network transmissions in the world today. In order to ensure that all communications maximize security while minimizing losses in latency and quality, special methods are used which take advantage of the specific attributes of the transmitted data. Multimedia security is a field of security that focuses specifically on the secure transmission of multimedia data. More than ever before, Internet traffic has been dominated by multimedia transmissions. As of September 2011, streaming multimedia applications generated over 53.6% of North American traffic [Sandvine11]. Since multimedia data currently comprises most of the Internet, ensuring multimedia security is important for both the senders and

receivers of content.

While there are definitely security methods that are directly suited to multimedia applications, traditional security methods can be used to some degree as well. Although multimedia data can be made up of any combination of images, video, and audio, at a lower level, it is still made up of individual bits. For this reason, traditional methods of security can be applied to the data in the same way as they can be applied for any kind of data. However, additional methods can be employed to ensure that every aspect of security is maintained without sacrificing the quality of the data.

The classification of multimedia data covers a wide variety of information types. However, many of the challenges that arise from ensuring multimedia security come from making sure videos are resistant to attack. Many of the problems in creating secure methods for video come from either the high information density of video files or the fact that streaming video files are frequently compressed into a lossy format [Lian07]. Every aspect of security must be approached differently for video: If a video is completely secure, but hard to view because of the enhanced security, then users will not want to go through the hassle required to watch the video [Lian09].

The most important aspects of video security are encryption, authentication, and steganography. To ensure that video transmissions are confidential when necessary, encrypting the information is a necessity. When transmitting complete video files, it is not difficult to protect a video because there is no demand for the data to be sent in real time. Most of the advances in video encryption are designed for streaming video, since it is difficult to give high levels of security without lowering the quality of the video in some way. Encryption also implicitly provides some degree of user authentication, since only authorized users can successfully decrypt the video. Authentication methods also exist outside of encryption. Many of these methods are based on watermarking, a method that embeds authentication information into the video. Steganography refers to a system similar to watermarking that can embed information into a video or image. Recent advances in steganography are mainly concerned with steganalysis, the process of extracting hidden information.

2 Encryption

The encryption of video is necessary to protect the confidentiality of sensitive information. Additionally, strongly encrypted video also implies authentication and integrity between the client and the server [Lian09]. The goal in encrypting video information is to maximize security while minimizing losses in quality. There are many kinds of video encryption algorithms that can be used depending on the situation that the video is being transmitted in. If both the sender and recipient have a large amount of computational power, then even high-quality video can be encrypted with the full protection of AES, but the amount of power necessary makes this kind of encryption impossible for most users to employ while streaming [Raju08]. When less powerful computers are being used, other methods that are less secure than AES can be used to give a level of security that is still relatively high.

2.1 Complete Encryption

The algorithms that are normally used for encrypting data cannot normally be applied to streaming video data as well. Block ciphers such as AES and DES can be used on normal quality video only if there is extreme computing power available [Raju08]. However, even though they cannot be used for high-quality video, less powerful computers can handle the application of block ciphers on low quality video [Shah11], but this scheme is only acceptable if high video quality is not a requirement of the end users. As they are less computationally intensive, stream ciphers like RC4 can be applied to the transmission bitstream for security as well, but they provide significantly less security than AES.

In extremely high security applications such as the government or military, the full encoding of every bit of video data may be necessary. In this case, the only way to fully encrypt streaming video is to either greatly reduce the quality or to use extraordinary computing power that is normally infeasible [Raju08]. Unfortunately, these limits mean that in order to send high quality video with maximum security, live streaming is not an option which can implemented without great cost. In most situations, there is not a need for such high security. In the entertainment industry, there is not much concern if only a few frames or fragments of video are taken, as long as thieves cannot steal the complete video [Raju08].

The true usefulness of the traditional encryption algorithms is their partial uses within some of the algorithms specifically designed for streaming video encryption. Different methods such as selective encryption and perceptual encryption use block ciphers to only partially encrypt the data [Shah11]. While partial encryption is not as secure as full encryption, when done correctly, it can offer a level of security that is acceptable for most streaming video applications [Shah11]. In the case where the encryption does not need to guarantee complete security but merely act as a deterrent for unauthorized access, the primary result of the encryption only has to ensure that the video is degraded enough that a watcher cannot visually discern the content [Li07]. Many of these perceptual encryption schemes are not cryptographically secure, but they can prevent users without any security knowledge from being able to view the content.

2.2 Partial Encryption

The primary goal in most video encryption algorithms is not the complete protection of the data. To fully protect video, all frames must be encoded, which is computationally infeasible [Raju08]. Instead, as long as unauthorized viewers cannot see the video data and attackers cannot recover it, then a video encryption algorithm is successful. Because of the encoding process, certain video frames carry image data while intermediate frames carry motion data. In order to give an acceptable level of security, only some of the attributes have to be encoded. As long as the partially encrypted portions both provide a high level of distortion and are cryptographically secure, then the video is reasonably secure for most streaming applications [Raju08].

The video encryption schemes that most closely follow the methodology of using block ciphers for complete encryption are the ones that feature partial encryption, where only a selection of the video data is encrypted [Shah11]. Most video encryption algorithms take advantage of the encoding process to perform most of the work [RoyS11]. Since nearly all video is compressed and encoded before being sent across the Internet, modifying the encoding process to encrypt as well as compress both saves time and ensures that the compression rate is not compromised by the encryption of the video [Shah11].

One of the most common sets of partial video encryption algorithms is the video encryption algorithm (VEA) and its updated versions. Modified VEA and real-time VEA (RVEA) are two improvements upon VEA that operate in a similar manner to VEA. All three algorithms work by encrypting some of the parameters used during the encoding process, particularly the sign bits of the discrete cosine transform (DCT) coefficients [RoyM11]. DCT is used in many lossy compression algorithms, and is one of the ways that MPEG (Motion Pictures Expert Group) video is compressed. By encrypting the DCT coefficients, the video is distorted enough that a viewer cannot see the real video without decoding the DCT with the correct VEA key. Modified VEA adds to the strength of the distortion by also encrypting the sign bits of the motion vectors in the intermediary frames [RoyM11]. RVEA again increases the security by using DES to encrypt 64 DCT and motion vector sign bits per frame of video [RoyM11].

Although the VEA set of encryption algorithms can sufficiently distort video so that it is not possible to determine the content of the video by simply watching the encrypted video, the original specifications for the

VEA algorithms are weak against cryptographic attacks. Both VEA and modified VEA simply use an XOR operation for their encryption, so known plaintext and ciphertext attacks can be used to find the secret key [RoyM11]. RVEA uses DES, which is not nearly as secure as the much better block cipher AES. Additionally, RVEA only encrypts 64 bits per frame, so videos with larger resolutions are not sufficiently distorted, allowing attackers to use visual analysis to recover the original video [RoyM11].

RVEA can be modified to have a high level of security regardless of video resolution. One simple, yet effective, modification is to replace the use of DES with AES. This eliminates the problem of an attacker breaking the encryption through cryptanalysis since it gives RVEA the same protections that AES has [RoyM11]. The problem with large resolutions being easy to break with visual analysis can be solved by making the number of bits encrypted per frame into a variable number: The larger the video resolution, the more sign bits will be encrypted [RoyM11]. Additionally, the use of more bits and AES instead of DES does not significantly increase the computation time on modern computers, meaning that this modified version of the RVEA algorithm can be used for real-time streaming video [RoyM11].

2.3 Perceptual Encryption

In applications where confidentiality is not a high priority, perceptual encryption can be used. Perceptual encryption algorithms do not necessarily have to have high cryptographic strength. Instead, the video merely has to be encrypted well enough to distort it so that it is unusable to users without any cryptographic knowledge [RoyS11]. These methods can also be resilient to attacks from hackers, but the main goal in perceptual encryption is to make sure that the video cannot be deciphered through human perception.

Although secure versions of VEA that use AES can be used on many modern computers, less powerful machines also have a need for secure streaming video. Encryption algorithms that run no faster than the encoding and decoding process are best on platforms with limited power. One method to create secure video transmissions is to fully combine the processes of encoding and encryption, creating a joint encryptor-encoder [RoyS11]. The algorithm works by encoding a slice of video, then encrypting it with a dual RC4 stream cipher, then performing post-encoding processes to ensure video format compliance [RoyS11]. The dual RC4 stream cipher is split into a temporal key and a spatial key. The temporal key is a global key stream that is used across the entire video, while the spatial key stream is reset for each encoding slice. The initial spatial key in each key stream is based on the current temporal key. This setup allows the encrypted video to be resistant to frame loss since the spatial keys are recalculated with each slice [RoyS11].

Partial encryption algorithms also exist for MPEG4, which is normally referred to as Advanced Video Coding (AVC). AVC contains more information than its predecessors, MPEG1 and MPEG2. The AVC compression process is different enough that merely encrypting the DCT and motion vectors will not provide the same level of security that it does in older video versions [Lian08]. When encoding AVC videos, modifications must be made to account for the intra-prediction mode used during the encoding process, as the extra mode information that intra-prediction mode provides can be used to reconstruct video if not encrypted properly [Lian08].

3 Authentication

Part of the field of security is ensuring that only authorized users have access to information. Authenticating video is no different, and in a similar way to video encryption, many of the same principles and algorithms used in normal authentication can be used when authenticating video. Authentication can be provided to some degree through encryption regardless of the data type: If information is encrypted with a secret shared only between authorized servers and clients, then both the clients and the servers are authenticated to each

other as long as the secret is not compromised. However, encryption is a computationally expensive process, and if the content of the video is not secret, then there is much less motivation to use encryption for authentication.

Authentication is especially useful in the realm of digital rights management. Generally, in the entertainment industry, the authentication is one-way. The client must authenticate to the provider, but the provider does not have to show its authenticity to the user through the video [Lian09]. The authentication methods provide two different securities against content theft: They prevent unauthorized users from gaining access to the videos directly from the provider, and they can also embed the userâ€™s information into the video itself in the form of a digital fingerprint. This way, if a user decides to perform a copyright violation by distributing the video, authorities can prove who leaked the video by checking the fingerprint information against the information held by the video provider [Lian09].

Methods exist which are specifically tailored for authenticating video communications. Much of the authentication methods for video are based on watermarks, which alter the video data in such a way that the source can be authenticated. Signature-based authentication also exists, which uses the video information but does not integrate it into the video transmission itself [Wang09]. Both methods allow videos to be authenticated without the computational overhead required by encryption.

3.1 Authentication Through Watermarks

There are many different kinds of watermarking algorithms for video, but all of them share the general purpose of being used to authenticate video. In general, watermarks are either generated based on the content of the video, or independently generated [Wang09]. Both types take advantage of the compressed nature of video. By introducing altered values into the encoding process, the video can be altered in a way that does not produce a discernable change to the viewer of the video, but the alterations can be picked up by client authentication methods that can determine if the watermark is valid [Wang09]. Since data is being hidden to some degree, watermarking can be seen as a special case of steganography, but the data is not hidden to the same degree as it is with both fingerprinting and normal steganographic methods [Paul11].

Watermarks that are generated independently of the video data are easier to generate and apply to the video data, but offer the least amount of security. Since the watermarks do not rely on the content of the video, they are easier to detect than those that adapt to the information [Wang09]. If the watermark can be easily extracted, it can be analyzed and used to create forgeries using the broken watermark. The content-dependent watermarking algorithms do not carry a huge computational cost, so whenever possible, content-independent watermarks should be avoided [Wang09].

Watermarking algorithms that rely on the content of the video provide better security. Some methods exist that use intermediate motion frames as well as image frames, but by watermarking only the image frames, an entire video can be authenticated [Vashistha10]. In addition to authentication, watermarking in this manner also preserves the integrity of the video, allowing the end users to know if an attacker has tampered with the data [Vashistha10]. This allows video to be authenticated with the same methods used for images alone. Most watermarking methods work in a similar way to the video encryption algorithms: The DCT or the discrete wavelet transform information is altered slightly to produce a distortion [Parameswaran08]. In encryption, the goal is to produce an extreme distortion, but in watermarking, the distortion should distort the video as little as possible.

Capitalizing on the human boundaries of visual perception, one algorithm attempts to maximize the amount of distortion that can be applied to a video without it being noticed by a human observer [Koz08]. Without taking human vision into account, the only way to ensure that a watermark does not create a visible distortion

is to perform the smallest necessary transformations necessary. The primary value that must be respected in this method is the temporal contrast threshold, the smallest change across two frames of a video that can be physically detected [Koz08]. By creating boundaries for what kinds of distortions are actually visible to the human eye, more space can be granted for the watermark, making it cryptographically harder for an unauthorized user to detect and forge [Koz08]. Individual component analysis can also be used to watermark video. By comparing temporal frames to detect the motion in a video, a watermark can be created and applied before encoding takes place [Sun09]. If the watermark is placed only in the parts of the video where there is the highest motion, then it is unlikely that human vision will be able to determine any distortion [Sun09]. Although this system does not actively consider the thresholds of human vision, by placing the watermark in a high motion area, any distortions that take place are much less likely to be seen by human eyes [Koz08]. A similar method using individual component analysis also exists for applying the watermark after encoding has taken place [Liang08]. Instead of directly applying the watermark to the uncompressed data, this method changes the DCT coefficients in the luminance section of the data. Again, the transformation takes place only in the highest motion areas of each frame, leaving no noticeable distortions [Liang08]. Although both methods focus on the temporal domain of the video, they both show resistance to spatial and temporal attacks [Liang08, Sun09].

One possible attack against many watermarking systems converts the information into analog data, and then converts it back to digital data. Known as a camcorder attack, it can be combined with geometric distortions to form a good attack against watermarks [Oh10]. This kind of attack causes the watermark data to become spatially asynchronous with the correct watermark pattern even though the watermark has only been geometrically transformed in some way. To defeat such geometric and analog based attacks, the transformed watermark is generated from the transformed data [Oh10]. Then, by comparing the transformed watermark to the geometrically altered video, the watermark can be verified without the end user needing any knowledge of how the video looked before it was distorted. This way, as long as both legitimate parties know the watermarkâ€™s secret key, the video can be authenticated despite the camcorder attack [Oh10].

3.2 Fingerprinting for Leak Detection

When watermarks are used specifically used to determine which user has distributed a video, they are referred to as fingerprints [Lian09]. Conceptually, fingerprinting works in a nearly identical way to watermarking, but instead of containing information for immediate authentication, fingerprints contain information for authenticating the user after the video has been leaked. If the distributer recovers a leaked video, then it can prove exactly which user downloaded the video and illegally sent it across the Internet [Lian09]. Removing the fingerprint from a video is the goal in attacking. Like watermarks, fingerprints are designed to be hard to detect and minimally distorting [Lu09]. If the fingerprint is removed, then the video cannot be easily traced back to who leaked it. For this reason, fingerprinting is closer to steganography than general watermarking is [Paul11].

In addition to simply finding videos that have been distributed in violation of copyright, fingerprints can also be used as part of an automated system to prevent others from gaining access to the compromised information. By sending databases of compromised videos and their fingerprints to video hosting sites such as YouTube, filters can be applied to the stolen videos that prevent them from being played or even showing up in searches [Lu09]. Recently, it has become the legal responsibility of video hosting sites to do everything in their power to remove illicit copyrighted materials due in the United States, especially since fingerprinting algorithms allow the hosting sites to easily filter out illegal content [Lu09].

While authentication in general can be carried out with either watermarking or signatures, fingerprints should not be sent as signatures alongside the video data. If a signature is used to send fingerprint data, then attackers do not have to go through the extra trouble of analyzing the content of the video to detect the location of the

fingerprint [Lian09]. While watermark data can be recovered by a client application as part of the process of verifying a clientâ€™s identity, the steps to recovering a fingerprint should be kept as secret as possible. This way, the potential for an attacker to be able to remove the fingerprint is diminished [Paul11].

3.3 Authentication through Signatures

Video transmissions can be authenticated without watermarks. While watermarks rely on being contained within the video data while not distorting it too much, signatures are sent separately from the visual information. Additionally, signatures can verify both the authenticity of the source and the integrity of the data just as watermarks can [Atrey07]. There are a variety of signature-based authentication methods, and they mainly differ in whether they are applied before the encoding process or afterwards. Unfortunately, even though signatures do not modify the original video data, they do have the downside of requiring additional bandwidth since they are sent alongside the video stream. In situations where network speeds are limited, watermarking is the superior choice [Saadi10].

When working with uncompressed video, signatures can be calculated without regards to the video format [Atrey07]. In one algorithm, the signatures can be scaled to fit multiple partitions of the video. The individual signatures are created by hashing together a secret value with the properties of the portion of the video being used. By allowing signatures to be calculated for either each individual key frame, a set of related frames, or the entire video, the signature can maximize the security it provides depending on the situation the video is being transmitted in [Atrey07]. While a signature based on the entire video ensures that the entire video is authentic, the end user can only prove the videoâ€™s authenticity after all of the information has been transmitted. In real-time applications, the signature must be based on either small sets of frames or individual key frames for the user to know the video is authentic as it is received [Atrey07].

Using uncompressed video data is not necessary to calculate a strong signature. Videos that have already been encoded can be signed in a similar way to their non-encoded counterparts. One method for authenticating encoded video is to calculate the signature based on a secret key and the features extracted from the DCT encoded frames [Ahmed07]. Since the hash is applied to each frame separately, the algorithm can be used to sign video in real time without encountering any problems [Ahmed07]. The main weakness of this particular algorithm is that it only relies on spatial data instead of temporal data, so although the algorithm is good for authentication, it may not completely protect against tampering in the temporal domain.

In order to increase the security of a video authentication method, watermarks and signatures can be combined. Since two different methods are being used in tandem, the computational costs are higher. However, the increase in computation time is low enough that a combined algorithm can be feasibly implemented on most modern platforms [Saadi09]. The inclusion of a signature does still incur extra costs in bandwidth, and in low-bandwidth settings, watermark-only authentication should be used instead of a combined method [Saadi10]. In combined algorithms, both methods are applied nearly independently of each other and then compared for accuracy [Saadi09]. One combined algorithm operates on AVC by using feature extraction. The main features that are extracted for use by this method are high motion areas, and the watermarking portion of the algorithm functions in a similar way to the algorithm described above that also uses higher motion [Saadi09, Liang08]. The signature is generating by performing an MD5 hash of the extracted feature information. If the hash matches the value of the watermark, then the video is authenticated.

4 Steganography

While data hiding itself does not necessarily improve the security of video, steganography must be included in

any multimedia security overview. Video data is both high-volume and lossy, definitely making it a suitable hiding place for data. Like watermarking, video steganography must not create visual distortions that are great enough to be discovered with human eyes. Most data hiding is done with JPEG since JPEG images are the most widely used image format [Fridrich07]. However, key frames in videos are encoded in a very similar manner to JPEG: Both use DCT. So, since encoded videos contain frames of JPEG-style images, the methods used to hide data within JPEGs can be extended to video. Furthermore, watermarks can be seen as an extension of steganography since they must also be hidden within the video data. Much like a needle in a haystack, if a minute number of bits are encoded across a large image, then the hidden data is statistically undetectable [Fridrich07]. When the boundaries of distortion in an image are pushed, steganalysis has a much higher chance of being able to recover hidden data.

4.1 Hiding data with Steganography

Successful data hiding requires that the container for the data, in this case videos or images, is not distorted in a way that can be determined by either human eyes or analytic algorithms. As long as the amount of data encrypted is very small, steganalysis is hard to perform accurately. For this reason, most of the recent research in steganography has been in creating new analysis algorithms instead of new hiding methods.

In order to ensure that data hidden within a JPEG image (or by extension, a key video frame) is undetectable by statistical analysis, the highest possible rate that can be used is 0.05 bits per non-zero DCT coefficient [Fridrich07]. The steganography algorithm capable of this level of hiding data is texture-adaptive perturbed quantization (PQ). This algorithm is a content-adaptive method that determines how to hide the data based on an analysis of the image and a secret key. As long as the boundary for detection is observed, data hidden with the method should be undetectable [Fridrich07]. However, in situations where more data needs to be hidden, image analysis has a greater chance of being able to uncover the hidden data.

PQ is based on spatial data along when performed on static images, but it can be adapted for use on the temporal data in video as well [Cao11]. It operates on a similar principle to the texture-adaptive PQ method used for JPEG data, but since it operates on video files, the storage capacity is increased due to the much larger file size [Cao11]. The time taken to encode video is increased when this form of steganography is used, but if the amount of hidden data is not too large, then it has the potential to be used in real-time [Cao11]. As with the spatial version of PQ, if miniscule amounts of data are encoded into the video, then steganalysis is statistically infeasible [Cao11]. With more data, image distortions increase, leading to correlations in the data that can be found with analysis.

4.2 Discovering data with Steganalysis

In many situations, more data than a handful of bits needs to be encoded in an image. Whenever the maximum undetectable data-hiding rate is broken, steganalysis has a chance at recovering hidden data. Multiple methods exist for analyzing images for hidden data, but the success of all of them depends on the density of the data that is hidden within the image [Fridrich07].

One way to analyze hidden data is to use Support Vector Machines (SVM) for recognizing patterns within images [Shi07]. As long as there is a statistically observable change in the image, the pattern recognition algorithm has a probability of being able to accurately find the hidden data. In particular, since most steganography methods hide the data within the DCT coefficients, this algorithm checks for patterns within that information. Another method that could be similarly used to check for patterns within the data would be to use neural networks to detect patterns [Shi07]. However, since neural networks are harder to use than SVMs, and SVMs have similar strength in this application of pattern finding, there is no reason to use neural

networks [Shi07]. As long as either learning structure is trained correctly, then they can be used to efficiently search for hidden data.

A more specific use of SVMs in steganalysis is to determine if an image has been double-compressed. Some steganography methods use multiple compressions to help hide data, and in order to recover hidden data, the number of compressions used normally has to be known [Fridrich07]. Regardless of the presence of hidden data, images that have been compressed multiple times with the same algorithm have distinct statistical correlations to each other [Pevny08]. Since SVMs have the capability of finding patterns within data, they are perfectly suited to classifying compressed images. In nearly the same way that double-compression is detected, SVMs can also be used to determine the features of an image most likely used during the data hiding process [Pevny08]. Since these measurements of the qualities of an image are used in some steganographic algorithms, finding them can help to recover hidden data [Pevny08].

5 Conclusion

While video security is definitely based on normal information security, the unique properties of video files increase the difficulty of ensuring security. The primary areas that are important to video security are encryption, authentication, and steganography. However, each of these fields of security is related to each other: Encryption provides a level of user authentication since only authorized users can unencrypt the video files, and all authentication methods must use steganography to a degree in order to prevent attackers from modifying the information.

Encryption is difficult to perform in real-time applications without harming the quality of video in some way. When dealing with static video files with no time constraints, the full security of AES can be employed as it can be with any other kind of information, and for now, the current levels of computing power available to most people will prevent the complete encryption of streaming video files [Raju08]. An acceptable level of confidentiality can be provided with either partial encryption or lightweight perceptual encryption. The stronger methods of partial encryption such as real-time VEA with AES do not completely protect the video, but severely limit an attackerâ€™s capability to break the encryption [RoyM11]. Perceptual encryption is mainly concerned with providing enough encryption to distort the video beyond a humanâ€™s capacity to determine what happens in the video. Some of these methods are cryptographically weak, but since they simply act as deterrents instead of guarantors of security, they serve their purpose well [Raju08].

Watermarks and signatures are better suited to user authentication than encryption is. Watermarks are hidden within the data of the video, and are normally used to authenticate the user to the content provider [Lian09]. In order for a watermark to be useful, the authentication information should be available even if the video has been distorted in some way [Oh10]. Signatures are an alternative to watermarks that does not alter the video information, and they can be combined with watermarks to further increase the security of the authentication [Saadi09]. Fingerprints are special watermarks that store user authentication information within a video file. If a video is leaked in violation of copyright, content providers can use the fingerprint to both stop the video from being viewed on distribution sites such as YouTube and prosecute the individual who leaked the video originally [Lu09].

Steganography is used to some degree in hiding watermarks and fingerprints, but the primary focus of steganography is hiding any arbitrary data. Most steganography is carried out through JPEG images, but since DCT is used in video for compression as well [Lian08], many of the methods used for JPEG steganography could be adapted for use in videos. Hiding data within videos is superior to hiding it in images because the large nature of video files gives more space to securely hide data [Cao11]. If a minute amount of information is hidden in an image or video, then the distortions will be so limited that analysis will have a low probability of recovering the data [Fridrich07]. In cases where there are larger amounts of secret data, then steganalysis

can be used to uncover the hidden information [Shi07].

Since video has become a larger part of consumersâ€™ lives in the Internet, the entertainment industry will continue to expand their efforts into online distribution of content [Sandvine11]. However, since piracy is definitely a problem online, companies must use security methods to ensure that profits from video sales are not lost [Lian09]. Even outside of corporate video distribution, end users who wish to communicate with video have a need for security as well. As long as video is widely used as a means of communication, methods must exist to protect the privacy of people and the rights of copyright holders.

6 References

[Lian09] S. Lian, D. Kanellopoulos, G. Ruffo, "Recent Advances in Multimedia Information System Security" Informatica, Vol. 33, No. 3-24, Pg. 3-24, 2009
http://www.informatica.si/PDF/33-1
/11_Lian%20-%20Recent%20Advances%20in%20Multimedia%20Information%20Syst.pdf

[Raju08] C. N. Raju, G. Umadevi, K. Srinathan, C. V. Jawahar, "Fast and Secure Real-Time Video Encyrption" Sixth Indian Conference on Computer Vision, Graphics and Image Processing 2008, pg. 257-264, Dec. 2008
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4756080

[Atrey07] K. Atrey, W. Yan, M. Kankanhalli, "A scalable signature scheme for video authentication", Multimedia tools and applications, Vol. 34, No. 1, pg. 107-135, 2007
http://www.acs.uwinnipeg.ca/pkatrey/papers/2007_MMTA_VideoAuth.pdf

[Wang09] J. Wang, J. Lu, S. Lian, G. Liu, "On the Design of Secure Multimedia Authentication", Journal of Universal Computer Science, Vol. 14, No. 2, pg. 426-443, February 2009
http://www.akademik.unsri.ac.id/download/journal/files/jucs/jucs_15_02_0426_0443_wang_oaj_unsri.pdf

[RoyM11] M. Roy, C. Pradhan, "Secured selective encryption algorithm for MPEG-2 video" International Conference on Electronics Computer Technology, pg. 420-423, April 2011
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5941730

[Shah11] J. Shah, V. Saxena, "Video Encryption: A Survey" International Journal of Computer Science Issues, Vol. 8, No. 2, pg. 525-534, March 2011
http://arxiv.org/pdf/1104.0800

[Fridrich07] J. Fridrich, T. Pevny, J. Kodovsky, "Statistically undetectable jpeg steganography: Dead ends, challenges, and opportunities", Proceedings of the 9th workshop on Multimedia and Security, Pg. 3-13, 2007
http://dl.acm.org/citation.cfm?id=1288872

[Shi07] Y. Shi, C. Chen, W. Chen, "A Markov Process Based Approach to Effective Attacking JPEG Steganography", Information Hiding, Vol. 4437/2007, pg. 249-264, 2007
http://www.springerlink.com/content/c308w1674110v420/

[Paul11] R. Paul, "Review of Robust Video Watermarking Techniques", IJCA Special Issue on Computational Science-New Dimensions and Perspectives, pg. 90-95, 2011
http://www.ijcaonline.org/nccse/number3/SPE169T.pdf

[Lu09] J. Lu, "Video fingerprinting for copy identification: from research to industry applications",

Proceedings of SPIE-Media Forensics and Security XI, Vol. 7254, pg. 1-15, January 2009
http://idm.pku.edu.cn/jiaoxue-MMF/2009/VideoFingerprinting_SPIE-MFS09.pdf

[Cao11] Y. Cao, X. Zhao, D. Feng, R. Sheng "Video Steganography with Perturbed Motion Estimation" Information Hiding, Vol 6958, pg. 193-207, 2011
http://www.springerlink.com/content/9r27706p222725x2/

[Koz08] A. Koz, A. Alatan, "Oblivious Spatio-Temporal Watermarking of Digital Video by Exploiting the Human Visual System", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 18, No. 3, pg. 326-337, March 2008
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4449114

[Li07] S. Li, G. Chen, A. Cheung, B. Bhargava, K. Lo, "On the Design of Perceptual MPEG-Video Encryption Algorithms" IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17. No. 2, pg. 214-223, February 2007
http://arxiv.org/pdf/cs.MM/0501014

[Lian07] S. Lian, Z. Liu, Z. Ren, H. Wang, "Commutative Encryption and Watermarking in Video Compression", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 6, pg. 774-778, June 2007
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4220724

[Liang08] C. Liang, H. Wu, A. Li, "Video Content Authentication Technique Based on Invariant Feature Detection and Cloud Watermark", Eighth International Conference on Intelligent Systems Design and Applications, pg. 602-607, November 2008
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4696400

[Pevny08] T. Pevny, J. Fridrich, "Detection of Double-Compression in JPEG Images for Applications in Steganography" IEEE Transactions on Information Forensics and Security, Vol. 3, No. 2, pg. 247-258, June 2008
http://ws2.binghamton.edu/fridrich/Research/dc_7_dc.pdf

[RoyS11] S. Roy, J. Tian, H. Yu, W. Zeng, "A Multi-Layer Key Stream Based Approach for Join Encryption and Compression of H.264 Video" IEEE International Conference on Multimedia and Expo, pg. 1-6, September 2011
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=6012179

[Saadi09] K. Saadi, A. Bouridane, A. Guessoum, "Combined Fragile Watermark and Digital Signature for H.264/AVC Video Authentication", 17th European Signal Processing Conference, pg. 1799-1803, August 2009
http://www.eusipco2009.org/papers/1569192266.pdf

[Saadi10] K. Saadi, A. Bouridane, A. Guessoum, "H.264/AVC video authentication based video content", 2010 5th International Symposium on I/V Communications and Mobile Network, pg. 1-4, December 2010
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5656206&tag=1

[Lian08] S. Lian, J. Sun, G. Liu, Z. Wang, "Efficient video encryption scheme based on advanced video coding", Multimedia Tools and Applications, Vol. 38, No. 1, pg. 75-89, 2008
http://www.springerlink.com/content/a2062g5t7k802324/

[Oh10] T. Oh, M. Lee, K. Kim, H. K. Lee, H. Y. Lee, "Robust high-definition video watermarking based on

self-synchronizing signals against composite distortions", Optical Engineering, Vol. 49, No. 9, pg. 097006-1 to 097006-14, September 2010
http://dx.doi.org/10.1117/1.3488053

[Parameswaran08] L. Parameswaran, K. Anbumani, "Content-based watermarking for image authentication using independent component analysis", Informatica, Vol. 32, pg. 299-306, 2008
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.159.4129&rep=rep1&type=pdf

[Ahmed07] F. Ahmed, M. Y. Siyal, "A Robust and Secure Signature Scheme for Video Authentication", 2007 IEEE International Conference on Multimedia and Expo, pg. 2126-2129, July 2007
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4285103

[Vashistha10] A. Vashistha, R. Nallusamy, A. Das, S. Paul, "Watermarking video content using visual cryptography and scene averaged image", 2010 IEEE International Conference on Multimedia and Expo, pg. 1641-1646, September 2010
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5583256

[Sandvine11] Sandvine Incorporated ULC, "Global Internet Phenomena Report Fall 2011", October 2011, pg. 5-6
http://www.sandvine.com/downloads/documents/10-26-2011_phenomena/Sandvine%20Global%20Internet%20Phenomena%20Report%20-%20Fall%202011.pdf

[Sun09] Z. Sun, J. Liu, J. Sun, X. Sun, J. Ling, "A motion location based video watermarking scheme using ICA to extract dynamic frames", Neural Computing and Applications, Vol. 18, No. 5, pg. 507-514, March 09
http://www.springerlink.com/content/p6078323142p8830/fulltext.html

7 List of Acronyms

AES: Advanced Encryption Standard
AVC: Advanced Video Coding (Also known as H.264 and MPEG-4)
DCT: Discrete Cosine Transform
DES: Data Encryption Standard
JPEG: Joint Photographic Experts Group
MD5: Message Digest Algorithm 5
MPEG1: Motion Pictures Experts Group 1
MPEG2: Motion Pictures Experts Group 2
MVEA: Modified Video Encryption Algorithm
PQ: Perturbed Quantization
RC4: Rivest Cipher 4
RC5: Rivest Cipher 5
RVEA: Real-time Video Encryption Algorithm
VEA: Video Encryption Algorithm

Last Modified: November 27, 2011
This and other papers on latest advances in network security are available on line at
http://www1.cse.wustl.edu/~jain/cse571-11/index.html
Back to Raj Jain's Home Page