

Video Encryption System

Brian Carter, Ari Kassin, and Tanja Magoc

November 20, 2007

1 Existing Problems with Content Distribution

Currently, there exists a major problem in the movie industry of mass piracy. Once content is able to be cracked by a user, it is distributed quickly on file-sharing networks throughout the world. It is a well-known fact that this costs the movie industry billions of dollars every year [1]. The system proposed below aims to solve this problem by providing a secure means for content distribution.

2 Typical Expected Use Scenario

Two parties will use the system: a provider and a user. The provider will operate all server-side aspects of the system while the user will deal with client-side aspects. The provider and the user will have some automated method of communication—the exact means (e.g., via Internet, satellite, or wireless connection) is irrelevant. The user will have some type of set-top box (STB) for processing received content.

In the typical scenario, the user will indicate to the provider that he or she wishes to purchase some specific video content from the provider. The provider will then prepare to transmit the content to the user. This consists of generating a unique key on both the client- and server-side ends of the system. Diffie-Hellman, or a similar algorithm, would be used to perform this operation. The user's STB will store this key securely and associate it with the content the user purchased. The STB will prepare to receive the purchased content from the provider.

The provider will transmit the content to the user while encrypting it using the unique key. The user's STB will receive the content. The entire encrypt-transmit-receive process must take less than real time (that is, the time length of the content being transmitted). At this point, the user may choose to begin playback of the content, or the user may opt to wait until a later time to play back the content.

When the user plays the content, it will be decrypted in memory and in real time using the unique key associated with the content. This decryption time must be faster than real time (that is, the decryption rate shall be higher than the frame rate of the video). It is important to note that the content will not ever be stored unencrypted on the STB; rather, it will be processed in real time and only stored in volatile memory.

There are advantages and disadvantages to this approach, summarized as follows:

- Pros:
 - Content distribution is highly secure—every time content is distributed, encryption is performed using a secure key
 - Fast distribution—content is transmitted, encrypted, and received faster than real time. On the user's STB, content is decrypted in faster than real time.
 - Storage is secure—content is never stored unencrypted on the user's STB
 - Potential for content expiration—once the STB no longer stores a content's key, the content is unusable for the user

- Cons:
 - Key management becomes a potential issue—there exists a need to securely manage multiple keys for each content and to associate each key with a specific content
 - Provider-side performance could be an issue—the provider needs to perform encryption and transmission of different content, with different keys, to different users simultaneously

3 Another Possible Scenario

We are interested in single video files that can be downloaded by an end user for playback. Consider a web application that provides video files to users. This web application and the content (all video files) are inside an intranet or the application is part of a military network. Assume that accessing the site requires authentication and authorization—this way the site can control the history and content that will be offered to each user. The way each video file is encrypted or decrypted is related to the credentials and information of each user. A “packet” shall be defined as a group of video files that a user has access to; there will be as many packets as users. Analyzing this configuration, the best case comes with a small-to-medium number of users with a large number of video files. The encryption or decryption process will use fewer keys than having every video file with its own decryption key. The worst case would be if there are many users and each packet has few videos creating more decryption keys than the total number of video files.

4 Possible Approaches to Encryption

Typically, video content is played back on a viewing device composed of red, green, and blue components. Therefore, video is often recorded in the same manner. This type of video is often referred to as RGB (red-green-blue). However, digital video is typically stored in a different format. It is possible to divide an image into two main components: the *luma*, or the brightness of an image, and the *chroma*, the color associated with an image. Luma is traditionally denoted as Y while gamma-corrected luma (which is almost always used in digital video) is denoted as Y' . Chroma is broken into two components: Cr, the red component, and Cb, the blue component. Digital video stored in this format is traditionally known as $Y'CbCr$. It is possible to convert video between RGB and $Y'CbCr$ format and vice versa without difficulty. [2, 3, 4]

The design of the human eye is the primary factor influencing the decision to prefer the $Y'CbCr$ format over the RGB format. The human eye is substantially more sensitive to light than to color; therefore, it is possible to vastly reduce the amount of signal associated with the chroma content while not affecting the way video content is perceived by humans. (Light is processed by the eye using structures known as *rods* within the eye while colors are perceived using structures known as *cones*.) [5, 6, 7]

As a result of the potential for reduction of chroma bandwidth, the chroma portion of a video signal is often substantially smaller (in terms of required storage space) than the luma portion—typically on the order of one-tenth of the entire video content. Furthermore, audio signals multiplexed into digital videos are also substantially smaller than the video signals (perhaps, again, approximately one-tenth the total video size).

Because the luma content is by far the largest component of a digital video, it is proposed that only the chroma and audio content be encrypted. This would vastly reduce the overhead required by encryption (due to the substantial reduction in content requiring encryption) while still providing some level of security. It is our belief that the unencrypted portion of the content, namely black-and-white video with no audio, would be considered completely useless by most file-sharing releasers and users. This nicely solves the problem of mass video piracy.

It is important to note that this idea is purely theoretical at this point—it may prove to be perfectly reasonable to encrypt and decrypt the entire video (i.e., luma, chroma, and audio) well within the time constraints discussed above.

5 Possible Encryption Algorithms

With so many different cryptosystems already existing, there are two main theoretical approaches that we are considering for this particular application.

The first possible approach is based on the fact that one of the main satellite providers, Dish Network, is currently using Triple-DES (TDES) encryption as the tool to provide security for its programming [8]. TDES has been proven as a secure algorithm, but its encryption and decryption are relatively slow. On the other hand, the predecessor of TDES, the original DES algorithm, is faster but has low security due to the short key used in the cryptosystem. We believe that by using the basic idea of DES, and tripling the size of its key, we can build a novel cryptosystem that will achieve the security of TDES but will perform much faster than TDES. The main challenge in this approach is to extend the S-boxes that will serve the purpose for the larger length of the key. On the other hand, the advantages of using this approach are our familiarity with these cryptosystems and the current practical usage of them by Dish Network.

The second approach, which would possibly open more unexplored opportunities, would be to perform a detailed study of algebraic number theory and try to apply some of these concepts to build a completely novel cryptosystem (possibly based on ideas behind RSA). However, even though this method might provide the opportunities to gain improvement over all existing cryptosystems, considering both security and speed, the main disadvantage of this approach is the uncertainty of what exactly it offers and our current unfamiliarity with these concepts.

Besides the two stated approaches, we also considered the other two mainly used cryptosystems: AES and RSA. We thought of testing the speed of each part of the AES algorithm to detect its weaknesses and try to improve them. However, with numerous implementations already existing that use numerous optimization techniques, it does not seem as reasonable to aim for such a goal. Building a new cryptosystem that will rely on basic field properties also does not seem promising to improve the performance of the already existing AES algorithm. On the other hand, a differently based RSA algorithm uses “nice properties” of prime numbers, but since there are not so many of these “nice properties,” this approach, in the same setting, does not seem to allow for any new possibilities, unless we can exploit similar characteristics in the area of algebraic number theory.

6 Conclusion

With growing piracy in the movie industry, encryption of movies is an important factor in secure transmitting movies through different media. To allow the speed of real-time encryption and decryption, we propose enciphering only chroma arrays and the audio portion of the video files, which, together, do not count for more than a quarter of the entire movie file. This way, the content will not be sufficiently recognizable to be watched and the time will allow real-time procedures.

We considered different encryption techniques and decided to explore what algebraic number theory can offer for this project. On the other hand, we consider an existing algorithm, namely DES, to improve its security by increasing the key size.

References

- [1] “Who Piracy Hurts,” *Motion Picture Association of America*,
http://www.mpa.org/piracy_WhoPiracyHurts.asp
- [2] “Luma (video),” *Wikipedia*,
http://en.wikipedia.org/wiki/Luma_%28video%29
- [3] “Chrominance,” *Wikipedia*,
<http://en.wikipedia.org/wiki/Chrominance>
- [4] “Chroma subsampling,” *Wikipedia*,
http://en.wikipedia.org/wiki/Chroma_subsampling
- [5] “Photoreceptor cell,” *Wikipedia*,
http://en.wikipedia.org/wiki/Photoreceptor_cell
- [6] “Cone cell,” *Wikipedia*,
http://en.wikipedia.org/wiki/Cone_cell
- [7] “Rod cell,” *Wikipedia*,
http://en.wikipedia.org/wiki/Rod_cell
- [8] “Broadband Satellite Access,” *Xilinx*,
<http://www.xilinx.com/esp/wired/optical/collateral/Satellite.pdf>