Selective Encryption and Watermarking of MPEG Video *
(Extended Abstract)

Tsung-Li Wu
E.C.E Department
North Carolina State University
Raleigh, NC 27695
twu2@eos.ncsu.edu

S. Felix Wu†
Computer Science Department
North Carolina State University
Raleigh, NC 27695
wu@csc.ncsu.edu

Feb 17, 1997

Abstract

MPEG (ISO Moving Picture Expert Group) is a compression standard for video processing and is widely used in multimedia application, e.g. *VOD (video on demand), HDTV (High Definition Television)*, and *DVD (Digital Video Disk)*. Confidentiality [MS95, AG96] and Copyright protection [HG96] are two security-related issues that have recently brought many attentions, especially when the MPEG streams are transmitted over the public Internet. In this paper, we present a research prototype which integrates selective encryption schemes with watermarking techniques in one system. In particular, we propose a simple and efficient scheme to securely distribute watermarked MPEG video streams over multicast/broadcast channels. We have experimented 7 different selective encryption schemes and one selective watermarking scheme on three well-known MPEG streams: *Bus*, *Flower*, and *Miss America*. Our result indicates that, for applications like video-on-demand or pay-per-view, we can selectively encrypt (< 1%) of the total MPEG stream, and (<3.7%) for each individual frame in the worst case.

* submitted to International Conference on Image Science, Systems, and Technology, CISST'97.

† Supported in part by the U.S. Department of Defense Advanced Research Projects Agency and the U.S. Air Force Rome Laboratory under contract F30602-96-C0325, and in part by the Center for Advance Communication and Computing under grant FAS #5-30183.

1 Introduction: MPEG

With the rapid growth of multimedia application in Internet, security and legal issues of copyright protection have become more important. The MPEG is the major compression algorithm to be used in video applications, and it has broad support from the consumer electronics, telecommunications, cable and computer industries. The high compression capability of MPEG translates into lower storage costs and less bandwidth needed for transmitting video over the network.

MPEG compression algorithm is a combination of a number of diverse tools, each of which exploit a particular data redundancy. MPEG has two classes of pictures/frames: intracoded and nonintracoded pictures. Intracoded picture is also called I-picture, and nonintracoded pictures are further divided into P-picture and B-picture. Each picture is divided into macroblock of 16 x 16 pixels for the purposes of motion estimation in MPEG compression and motion compensation in MPEG decompression. The basic idea of motion estimation is that if the location of a block in a picture can be predicted from the previous picture, only the displacement vector need to be coded and transmitted. If the error is too large (e.g. fast video motion in a sequence), then the block is encoded as an I-block. Blocks (8x8 pixels) of either an original pictures or the difference between a frame and the motion prediction are transformed using the *DCT (Discrete Cosine Transform)*.

2 Two Concerns: Confidentiality and Copyright Protection

Confidentiality of MPEG streams can be achieved by encryption. Cryptographic systems (either symmetric, e.g., DES, or asymmetric, e.g., RSA) permit only valid keyholders access to encrypted data. Once such data is decrypted, there is no way to track its *reproduction* or *retransmission*. Thus, *copyright protection* becomes another important concern. Furthermore, even the most efficient encryption scheme could introduce overhead. Especially, the decryption procedure in a secure MPEG2PLAY system could be the performance bottleneck if not being developed efficiently:

- The performance of a MPEG viewer can be affected significantly if the decryption and decoding modules are implemented purely in software [PSR93].
- In many MPEG applications, the encoding procedure (and thus the encryption procedure) only needs to be performed once (*e.g.,* a movie), while the decoding procedure is performed for each client.
- In MPEG, the encoding process runs much slower than the decoding process. On the other hand, in a symmetric encryption algorithm (*e.g.,*, DES or IDEA), the time to encrypt is equal to the time to decrypt. Thus, while the encoding process is slowed down 10% by the encryption task, the decoding process could have been slowed down 100% by the decryption task.

Both confidentiality and copyright protection for MPEG have been studied. Very little works have been published in techniques to handle both concerns at the same time. A naive way to solve this problem is to encrypt the whole watermarked MPEG stream. This approach introduces a new problem:

> The video server S would like to send the movie M to N different clients, $C_i, 1 \leq i \leq N$. S watermarks and encrypts N copies, $ENC_{K_i}[WM_i[M]], 1 \leq i \leq N$ (each such copy will contain a hidden mark associated with the receiving client's identity.). Since these N cipher copies are different, they can only be *unicasted* to their clients.

In applications like pay-per-view, the most efficient way to deliver the same movie M is through a multicast channel such that all customers can receive the same video stream. However, because of the watermarks, the streams are different for different clients such that we can not benefit from the multicast/broadcast network services.

3 Security and Performance Requirements

We consider the situation that a video source will deliver a movie or a teleconference video stream to a set of distributed clients in the Internet. The security requirements are:

1. An outsider on the Internet should not be able to view the video.
2. An insider (*i.e.,* one of the clients) should be discouraged to redistribute the video stream he receives.

Please note that, in order to achieve the second security requirement, a robust watermarking scheme is necessary to resist many different attacks. For example, the scheme needs to support *collusion-secure* [BS95] and *asymmetric* fingerprinting [PS96]. However, in the current stage, our implementation is neither collusion-secure nor asymmetric. We consider to support these features in near future.

The performance requirements are:

1. The security mechanism should not introduce too much computational overhead. We assume that special hardware for encryption might not be available everywhere and an efficient software solution is necessary.
2. The network bandwidth spent on delivering the video should be proportional to the size of the original video sequence. Please note that if we "unicast" the same movie but with different watermarks to N different client, the network bandwidth required is proportional to the size of the movie multiplied by the number of clients, which is undesirable.

4 Our Approach

"*Selective*" is the key to our approach. We choose small segments of bits from a MPEG video stream. Depending on the specific selection scheme we use, the chosen segments totally could be from 90% down to less than 1% of the original movie. And, we only watermarked and encrypted these small chosen segments. The result is that the rest of the MPEG stream is unchanged, and thus can be broadcasted or multicasted. We only need to unicast the chosen segments.

More specifically speaking: the movie $M = \{M_{plain}, M_{selected}\}$. For each client C_i, we generate $ENC_{K_i}[WM_i[M_{selected}]]$, which will be unicasted to C_i. The M_{plain} is multicasted to all clients, $C_1, ...C_N$. We define $SelectRatio_M = M_{selected}/(M_{plain} + M_{selected})$. If M_{plain} is very large comparing to $M_{selected}$ (or $SelectRatio_M$ is very small), then the required network bandwidth is proportional only to the movie size itself because most of the movie can be multicasted.

Although we only watermark and encrypt a small portion of the total stream, we need to guarantee:

1. Without properly decrypting the chosen segments, the whole stream is unviewable or very low quality.
2. Through the MPEG decoding process, the watermarks on the chosen segments will be automatically replicated to the entire video.

Secret key management is simpler when we only encrypt the unicast part but not the multicast part. If we want to encrypt M_{plain} and send it through the broadcast channel, then we need to worry about the *group key exhange problem* [HMR94], which is more complicated than simple key exchange [HC96].

5 Design and Implementation

In our experiment, we took the ISO/IEC MPEG-2 software video codec and mpeg2play developed by Stefan Eckart and Chad Fogg [EF95] as the basis. We modified the package so it will support both selective encryption and watermarking. The mpeg2play program has also been enhanced to decrypt the encrypted MPEG streams.

5.1 Selective Watermarking

A digital watermark is a visible, or preferably invisible, identification code that is permanently embedded in the video signal, it remains present within the signal after any decryption process. Add a watermark that authenticates the legal copyright holder and that cannot be manipulated or removed without degrading the image quality.

5.1.1 Spread-Spectrum

We use a simplify scheme for watermarking of MPEG video presented in [HG96]. The watermark is embedded into the uncoded video, and can be retrieved from the decoded video. The idea of watermarking for video is addition of a pseudo-random signal to the video that is below the threshold of perception and that cannot be identified and thus removed without knowledge of the parameters of the watermarking algorithm. The scheme to accomplish this is a direct extension of ideas from direct-sequence spread spectrum communications [BGM95, CKLS95]. In spread spectrum communications, one transmits a narrow band signal over a much lager bandwidth such that the signal energy present in any single frequency is imperceptible. Similarly, the watermark is spread over very many frequency bins so that the energy in any one bin is very small and certainly undetectable. The advantages of security against unintentional or intentional attack:

1. The location of the watermark is not obvious.
2. Frequency regions should be selected in a fashion that ensure severe degradation of the original data following any attack on the watermark.

$A_j \in \{-1, 1\}$ is a sequence of information bits we want to hide in the video stream. We then spread this discrete signal by a large factor C_r, called *chip-rate*, and obtain the spread sequence $B_i = A_j, (j \times C_r) \leq i < ((j+1) \times C_r)$.

This watermark vector will be embedded to the MPEG video. Even if the receiver knows the basic scheme, it cannot recover the information without knowledge of the pseudo-noise sequence. In our design, the same watermarks will be embedded ONLY in each I-picture of the video sequence. The advantage of this selective watermarking is big saving for computation time. Since I-picture is the most important (the index picture) information in the whole sequence. An attacker who try to destroy the watermarks will mess up the quality of the video at the same time. The idea of embedding the same watermark in each I-picture is to prevent the attacks like cutting out some clips or corrupting some parts of video sequence in order to destroy the whole watermark's integrity. In such cases, we still can retrieve the watermark from the other uncorrupted I-pictures.

5.1.2 Example

For a sequence with resolution 352 x 288 each picture. We embed watermarks in each I-picture (Luminance signal only):

1. Set chip rate $C_r = 1000$, which means we can embed 101 bits of information into the 101376 pixels. There are $((352 \times 288)/(8 \times 8)) = 1584$ Y blocks associated 1584 DC-cofficients, one for each Y block.

2. We have $A_j, 0 \leq j < 100$ and spread information $B_i, 0 \leq i < 101376$. Arrange the one-dimension array vector b to a 352x288 two-dimension matrix in raster-scan order.

3. Obtain a DC-coefficient for each block by applying DCT algorithm.

4. Add the watermark DC-coefficient to the prediction error of MPEG video in the order of raster-sam corresponding to the macroblock structure.

5. Repeat the same procedure for each I-picture.

5.2 Selective Encryption

Selective encryption and decryption approach is proposed to avoid encrypting the entire MPEG bitstream. The main motivation is to reduce the computation time in the MPEG decoding process without compromising the security of the transmission too much. Two selective encryption schemes proposed in [MS95, AG96] are I-picture encryption only and I-picture plus I-block (in P and B pictures) encryption. Theoretically, encryption of the I-picture alone would render the information in the P- and B-pictures useless. However, previous experiments indicated that those I-blocks embedded in the P and B pictures can reveal certain amount of information.

We consider 7 different selective encryption schemes for MPEG:

1. I-Frame only, DC/Luminance only (If-DCLum).

2. I-Frame only, DC/Luminance/Chrominance only (If-DCLumChr).

3. I-Frame only, DC/Luminance/Chrominance/AC (If-DCAC).

4. I-Frame plus I-Block, DC/Luminance only (IfIb-DCLum).

5. I-Frame plus I-Block, DC/Luminance/Chrominance only (IfIb-DCLumChr).

6. I-Frame plus I-Block, DC/Luminance/Chrominance/AC (IfIb-DCAC).

7. I/P/B Frame, DC/Luminance/Chrominance/AC (IPBf-DCAC).

For all 7 encryption schemes, we do not encrypt various headers in the MPEG streams. In our implementation, we used DES with a modified version of the output-feedback mode to generate the cipher stream.

6 Performance Evaluation

Our experiments are performed on Pentium PCs running Linux. The source code is available from http://shang.csc.ncsu.edu. The package, however, does not include the DES code itself because of the US export control rule. A LinuxDES package obtained from somewhere else is needed to link with the secure MPEG package. All the mpeg files we generated are also available.

For evaluation, we have tested three popular MPEG streams: *Bus*, *Flower* and *Miss America*. The first result is the total number of bits being encrypted in all 7 different selective schemes:

Movies	*Bus*		*Flower*		*Missa*	
Bits	$M_{total} = 6899392$		$M_{total} = 6906512$		$M_{total} = 6479440$	
Scheme	$M_{selected}$	$SelectRatio$	$M_{selected}$	$SelectRatio$	$M_{selected}$	$SelectRatio$
If-DCLum	63183	0.0092	67812	0.0098	46324	0.0071
If-DCLumChr	73045	0.0106	87653	0.0127	65173	0.0101
If-DCAC	890029	0.1290	1246591	0.1805	1324104	0.2044
IfIb-DCLum	142008	0.0206	80195	0.0116	46589	0.0072
IfIb-DCLumChr	169777	0.0246	105698	0.0153	65639	0.0101
IfIb-DCAC	1577432	0.2286	1417669	0.2053	1331192	0.2054
IPBf-DCAC	5286256	0.7662	5529772	0.8007	5088292	0.7853

From the above results, we can clearly observe a trade-off between security and overhead. If confidentiality for an application is extremely important, then the IPBf-DCAC scheme (*i.e.,* encrypting I/P/B frames) should be chosen. On the other hand, for applications like pay-per-view, If-DCLum or If-DCLumChr is very attractive as the $SelectRatio$ is extremely low. This latter approach will not only reduce the overhead in the decryption/decoding process but also require much less network bandwidth in a multicast/broadcast environment.

7 Jitters in Transmitting/Multicasting MPEG Streams

So far we have only considered the total amount of encrypted bits in MPEG streams. We are also interested in how these encrypted bits are distributed in the whole MPEG stream. We would like to understand the *jitters* in the selective encryption process.

Jitter is an important concern in delivering MPEG stream over a real-time network environment [LCY94, OLS95, IR95, JLS96]. The numbers of bits per frame for all three MPEG files are depicted in Figure 1 and we can clearly observe the jitters introduced mainly by I-frames. In our approach, because of watermarking and encryption, some selected stream segments need to be unicasted (instead of multicasted). If the selected segments are within the I-frames, the jitters will be even greater because we need more bandwidth to transmit those selected and unicasted MPEG segments. In Figures 2,3,4, the jitters injected by the selective encryption scheme are illustrated. We can clearly observe that the If-DCLum approach introduces very small jitters.

In fact, we have calculated the amount of jitters introduced by If-DCLum frame by frame. We found that, if an I-frame contains K bits, then in the worst case If-DCLum will watermark and encrypt less than 3.7% of those K bits (*i.e.,* $< (0.0037 \times K)$). If we have only one client, then the required network bandwidth per frame is K. If we have N clients, the worst case bandwidth is $(1 + 0.0037 \times (N-1)) \times K$, which is close to K even if N is large. In the IPBf-ACDC scheme, the worst case ratio is around 90% and the required network bandwidth: $(1 + 0.9 \times (N-1)) \times K$, which is not scalable well with a big number of clients, N. Therefore, we conclude that, considering both security and performance, If-DCLum and If-DCLumChr are very attractive in a broadcast/multicast environment.

8 Remarks

In this paper, we present a prototype system to support both confidentiality and copyright protection for MPEG video streams. We have implemented different selective encryption schemes and evaluated

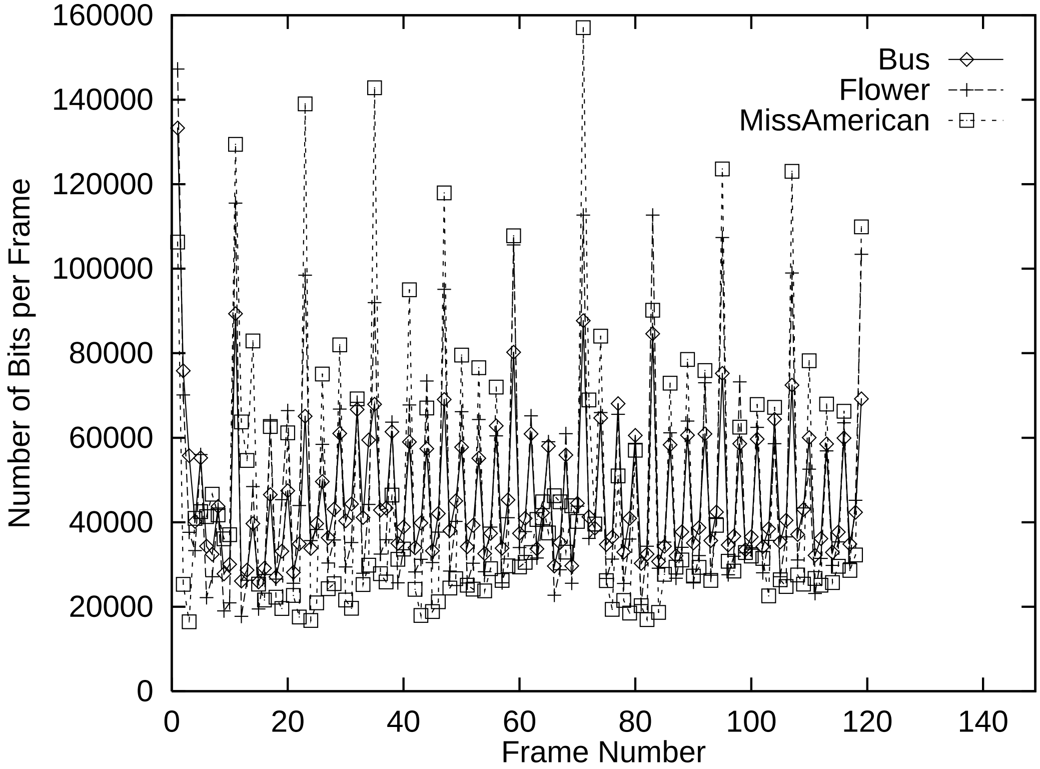

Figure 1: Per-Frame Bit Rates for Bus, Flower, and Miss America

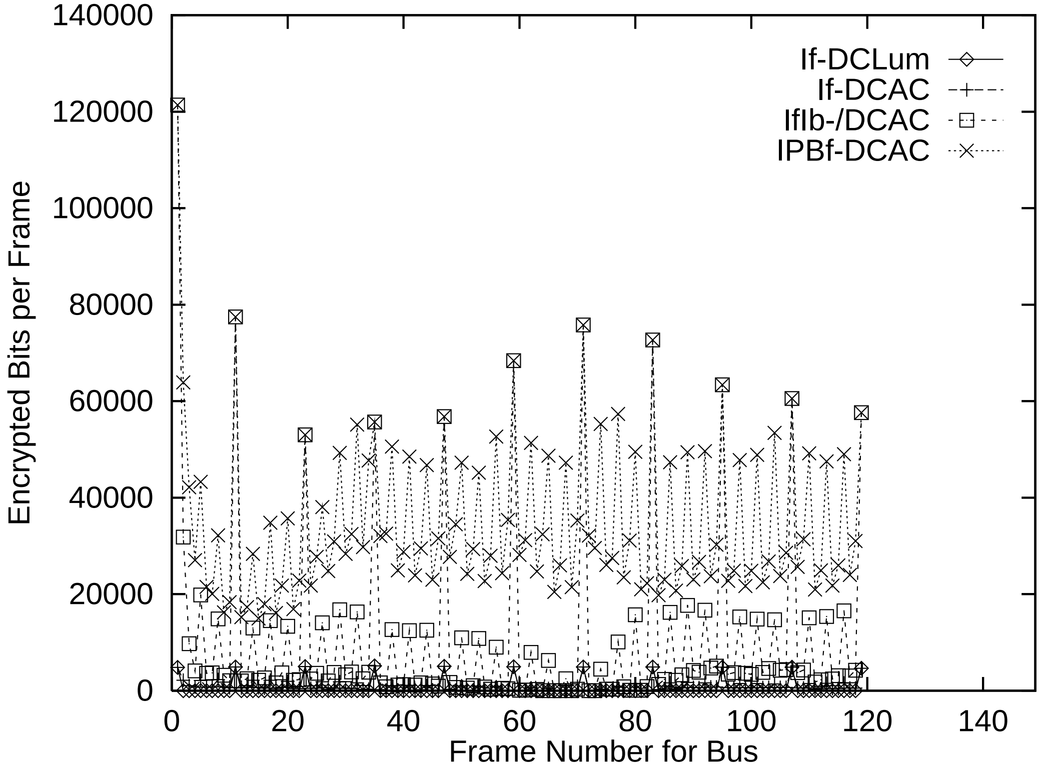

Figure 2: Per-Frame Encrypted Bit Rates for Bus (4 Selective Schemes)

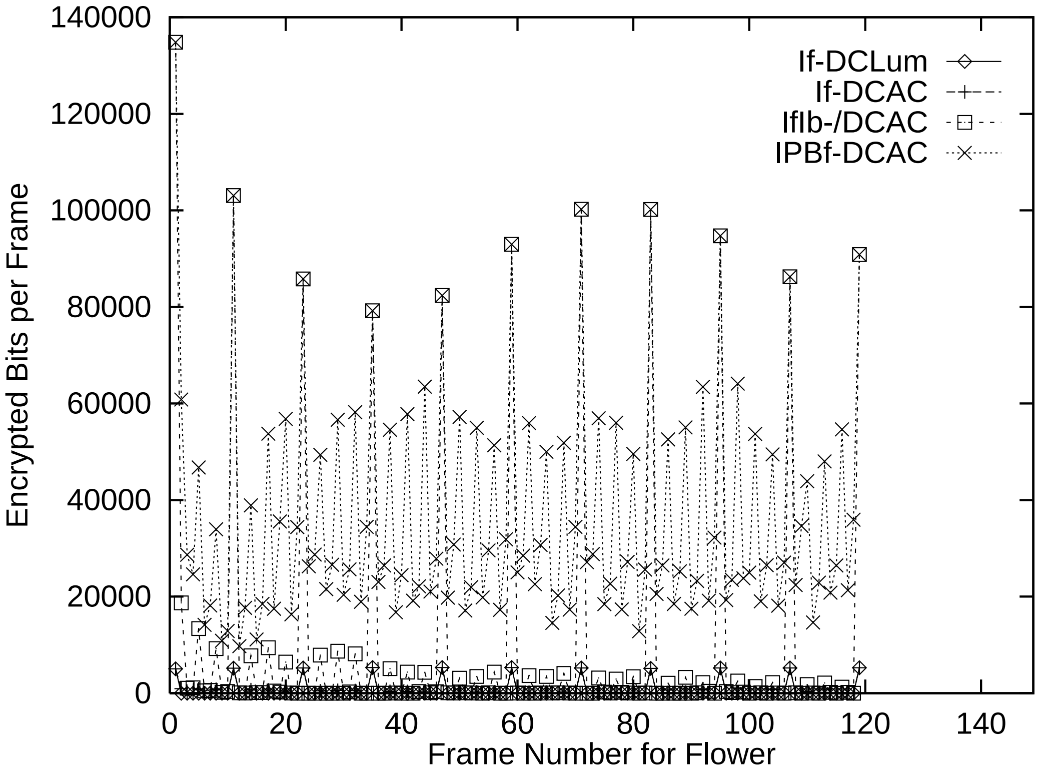

Figure 3: Per-Frame Encrypted Bit Rates for Flower (4 Selective Schemes)

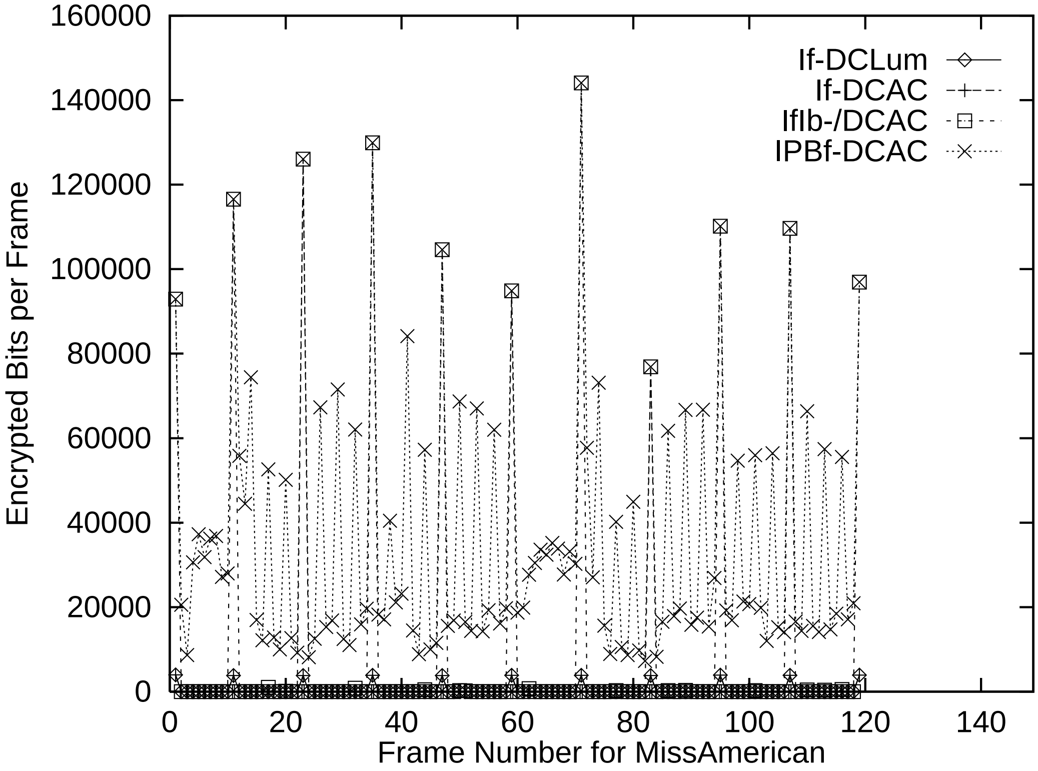

Figure 4: Per-Frame Encrypted Bit Rates for MissA (4 Selective Schemes)

their performance against three MPEG files: *Bus*, *Flower* and *Miss America*.

It is still a very open (but extremely important) research problem to *quantitatively* compare the "security" level achieved by these seven selective encryption schemes. Different applications might have different security requirements. The information revealed in one particular encrypted MPEG stream depends on not only the MPEG file itself but also the background knowledge of the human viewer. In this paper, we only focus on quantitative measures for the overhead introduced by the security enhancements.

One important result from our experiments is that selective schemes like If-DCLum or If-DCLumChr are extremely attractive in supporting applications like video-on-demand or pay-per-view. In these applications, the required confidentiality level is not high but the issue of quality of service is strongly demanded by the customers. Using If-DCLum, the outsider (who did not pay for the movie) can only view a corrupted movie. The client (who did pay) will enjoy a high-performance MPEG delivery but be discouraged to redistribute because of the embedded watermarks.

We did not cover in this paper how exactly our approach will map to network layer services like MBone and RSVP. We are currently looking at this issue. In the near future, we would like to extend our prototype to run on top of various network services and different scheduling policies.

References

[AG96] I. Agi and L. Gong. An Empirical Study of Secure MPEG Transmission. In *ISOC Symposium on Network and Distributed System Security*, San Diego, CA, February 1996.

[BGM95] W. Bender, D. Gruhl, and N. Morimoto. Techniques for Data Hiding. Technical report, MIT, Media Laboratory, 1995.

[BS95] D. Boneh and J. Shaw. Collusion-Secure Fingerprinting for Digital Data. In *Advances in Cryptology - CRYPTO 95*, pages 452–465, 1995.

[CKLS95] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure Spread Spectrum Watermarking for Multimedia. Technical report, NEC Research Institute, 1995.

[EF95] S. Eckart and C. Fogg. ISO/IEC MPEG-2 Software Video Codec. *SPIE*, 2419:100–109, 1995.

[HC96] D. Harkins and D. Carrel. The Resolution of ISAKMP with Oakley. Internet Draft, IETF, November 1996. Network Working Group.

[HG96] F. Hartung and B. Girod. Digital Watermarking of Raw and Compressed Video. Technical report, University of Erlangen-Nuremberg, 1996.

[HMR94] Hugh Harney, Carl Muckenhirn, and Thomas Rivers. Group Key Management Protocol (GKMP) Architecture. Internet Draft, IETF, September 1994. Network Working Group.

[IR95] M. Izquierdo and D. Reeves. Statistical Characterization of MPEG VBR Video Traffic. In *IS&T*, Feb 1995.

[JLS96] P. Jelenkovic, A. Lazar, and N. Semret. Multiple Time Scales and Subexponentiality in MPEG Video Streams. In *IFIP-IEEE Conference on Broadband Communications*, April 1996.

[LCY94] S. Lam, S. Chow, and D. Yau. An Algorithm for Lossless Smoothing of MPEG Video. Technical Report TR-94-04, University of Texas at Austin, Feb 1994.

[MS95] T. Mapels and G. Spanos. Performance Study of a Selective Encryption Scheme for the Security of Networked, Real-Time Video. In *4th International Conference on Computer Communications and Networks*, Las Vegas, Neveda, September 1995.

[OLS95] Jr. Olen L. Stokes. *Transmission of MPEG Compressed Video Through B-ISDN ATM Networks*. PhD thesis, North Carolina State University, Raleigh, NC, 1995.

[PS96] B. Pfitzmann and M. Schunter. Asymmetric Fingerprinting. In *Advances in Cryptology - EUROCRYPT 96*, pages 84–95, 1996.

[PSR93] K. Patel, B. Smith, and L. Rowe. Performance of a Software MPEG Video Decoder. In *ACM Multimedia*, pages 75–82, Anaheim, CA, August 1993.