https://doi.org/10.2352/ISSN.2470-1173.2017.7.MWSF-322

MP3 Partial Encryption for DRM

Martin Steinebach and Waldemar Berchtold, Fraunhofer SIT, Darmstadt, Germany

Abstract

Partial or selective encryption is a well-known concept in multimedia security. It aims to achieve a level of security of multimedia encryption comparable to common encryption by encrypting only a relevant subset of the complete stream or file. The prime benefit of partial encryption is better performance due to fewer encryption operations. In addition, partially encrypted media data can often be parsed as well as unencrypted media if no header data is encrypted.

The focus of partial encryption evaluation has almost always been the level of security that can be achieved. In this work, we discuss another aspect: when partial encryption of MP3 files is used in a DRM scenario, how many resources can be saved by it? As DRM usually is attacked by key sniffing or analogue recording, the security of the encryption itself is of lesser importance as long as it provides a sufficient hindrance to access the media data.

Motivation

As long as it has been possible to copy and distribute music and audio books, artists and their labels have been faced with the challenge of preventing the uncontrolled distribution of their work. Especially since the invention and wide availability of MP3 after which peer to peer networks and other ways of illegal distribution flourished, mechanisms for DRM (Digital Rights Management) emerged to protect the multimedia content. The basic idea of DRM is to restrict access to content by enforcing rules regarding the usage of the content.

One common way of enforcement is encryption. Content is stored in encrypted form on the user device. Consuming the content is only possible if the encryption key is available. Access to the key is bound to authentication mechanisms.

Common DRM solutions apply a full encryption to the content [17]. This means that all the content is encrypted by a standard encryption scheme.

Yet encrypting the whole audio file, makes it unplayable and suspicious to anti-virus software. Furthermore complete encryption (and later decryption) come with significant computational costs. Partial audio encryption (see also literature list) is an approach that relies on the fact, that some bits in the audio file are more important to the reproducibility of the content than others. A careful choice of what to encrypt can keep encryption light weight and the audio file properly formatted while making the audio stream useless without preceding decryption.

While partial encryption offers significant advantages in comparison to standard encryption, especially in the DRM scenario where security requirements are limited (or better: where weak security is commonly accepted due to the lack of secure key distribution). The important aspect of partial encryption here is that the computational costs of decryption at the end user device can become significantly lower. Therefore we aim at providing an efficient way to offer partial encryption for MP3 with a significantly lower computational cost at decryption compared to standard AES decryption as usually applied in DRM schemes like the Microsoft DRM packaging scheme.

Security needs in DRM

Security is a concept which depends strongly on the subject it is applied to. When we talk about security and encryption, confidentiality is most often the goal. Confidentiality in cryptography usually means that given an encrypted message, no meta-information about its content should be retrievable. This makes perfect sense in private emails, contracts, offers or even military secrets. Here strong cryptography is recommended.

But cryptography is also used to achieve other goals. In DRM, access control is usually the aim. It is important to stress that we talk about Digital Rights Management here, as in Document Rights Management confidentiality can be the primary aim. Commonly DRM is about restricting and controlling access to entertainment content. The key information about the content, as the general plot of a movie, the lyrics of a song or the subject of an ebook are likely to be known to the customer. They are not a secret to be kept confidential. The role of DRM and the cryptographic primitives used by it is to ensure that access to the content is controlled by rules. Circumvention of a DRM system can only then be seen as successful, if consuming the content is possible without significant loss of quality but at the same time ignoring the rules.

This points to a huge difference between security requirements of confidentiality and access control, at least in the sense it is enforced by DRM: an attack is only successful if it can circumvent the encryption without quality loss; getting a course idea about the content is not critical. Taking an JPEG images as an example, one would not worry about the possibility of retrieving only the lowest coefficients of the image or only a thumbnail-sized copy. Figure 1 illustrates this. Only if a full copy or one that is slightly distorted can be retrieved, DRM has failed.

Transparent encryption is a form of partial encryption explicitly using this idea to allow previews easily converted to full quality copies only with applying a cryptographic key.

The Potential of Partial Encryption

The previous section shows that the security requirements of encryption methods in DRM are limited compared to other applications. Encrypting only a subset of all data can be sufficient to ensure access control in entertainment.

This can reduce the computational cost of the encryption process: intuitively encrypting less data takes less time. But this is only true if it is trivial to localize the data to be encrypted and decrypted. Given the hardware-supported efficiency of standard encryption methods, a badly designed partial encryption method may be more computational expensive than a full encryption.

Multimedia formats of the MPEG family are very well suited

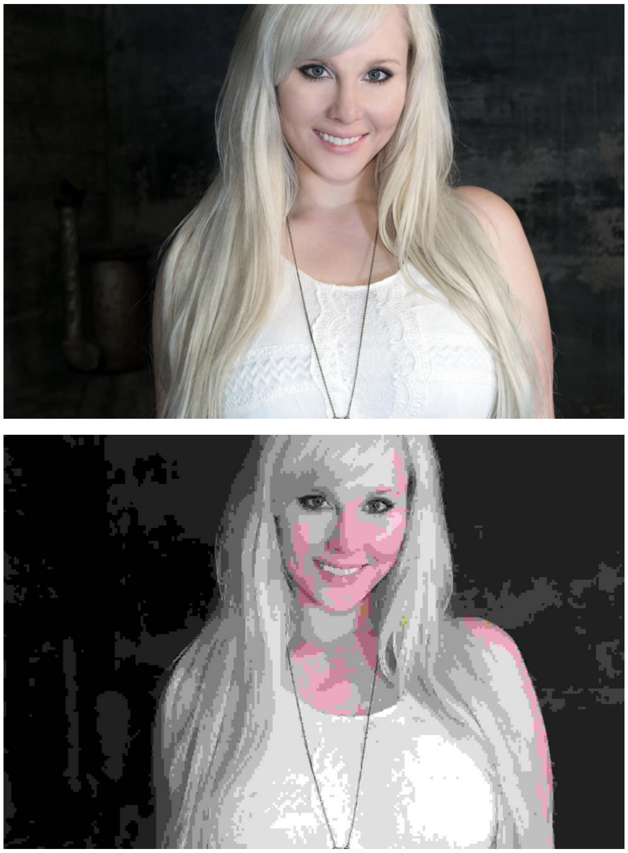

Figure 1. *Top:Original, Bottom: JPEG quality factor 1. This illustrates the difference between confidentiality and access control: The lower image would give away the identity of the person shown on the photo, therefore confidentiality would be lost, but the low quality would not allow reuse or a good customer experience, therefore access control in the DRM sense would not be violated. Image CC0 / pixabay*

to deal with this challenge. They offer a data structure where the different functions like header, sample data, quantization information can easily be accessed. Even more, during compression and decoding the data needs to be accessed by the CoDec exactly in the specified way, knowing the role of the data it currently handles.

Therefore integrating encryption and decryption directly into the CoDec allows partial encryption without any overhead for seeking suitable data to be encrypted. When the right data type is handled during coding or decoding, encryption or decryption of the current data is executed. The multimedia file only needs to accessed once for encoding and decoding, and not twice as in full encryption or non-integrated partial encryption. It should be mentioned that the encryption scheme for full and partial encryption can be identical. There are no specific requirements or limitations beyond supporting an arbitrary amount of data to be encrypted.

So if applied to a suitable file format partial encryption can reduce the cost of access control without much effort. In addition, partial encryption allows encrypted files which can still be parsed, allowing time position seeking without the need for decryption.

The Risk of Partial Encryption

Still, partial encryption is not widely used. One reason often mentioned is that for DRM, the best possible security scheme should be applied. This seems to be more a legal than a technical argument, possible coming from the discovered weakness (DeCSS, see e.g. [8])of the CSS DVD protection scheme ([7]), where the encrypted was circumvented by brute force.

The development of computational power also may be a reason why partial encryption is rarely applied today. Back in 1995 when SECMPEG was introduced [9], full encryption of video files would have caused an unacceptable overhead. Partial encryption was a compromise between security needs and computational feasibility. This compromise is not necessary today, as encryption of virtually unlimited amounts of data is not a challenge anymore.

From a strictly security-centric perspective, it therefore does not make sense to still apply partial encryption today. At best, it can achieve a comparable level of security, but it will not be more secure as it uses identical cryptographic primitives but only on a subset of the data.

Related Work

Partial encryption, and especially MP3 encryption, has been the subject of numerous works. The focus of these works is usually the algorithm's security. A common goal is to design an algorithm allowing partial encryption at a security level comparable to full encryption by standard algorithms like AES. In his book about multimedia encryption, Shiguo [3] stresses the importance of performance gain and also provides a metric for it by comparing compression and encryption time. Still, the prime focus of his discussion is security.

For MP3, several partial encryption strategies have been proposed, for example:

- Servetti et al. [1] encrypt bit allocation parameters.
- Gang et al, [4] apply scrambling of code books, regions and granules.
- Kwon et at. [5] use the MDCT coefficients for encryption and watermarking.

- Yen et al. [6] address encryption of side-information, sign-bits and Huffman-encoded data.

All these works have in common that they mention the efficiency of the partial encryption by discussing the portion of the MP3 to be encrypted and comparing it to full encryption. Still, no actual measurements for real-world performance are given.

Mp3 is not a very accessible format. Starting with the ISO specifications is not recommended. A good starting point is the introduction by Brandenburg [12]. There are also numerous blogs about the mp3 format offering introductions on decoding mp3.

Other Media Types

There are also a lot of publications on partial encryption for video. The survey on video encryption by Liu and Koenig [10] provides a systematic overview of the different approaches. The basic strategies and options for audio and video are similar as in both media types there are different types of data that can be encrypted at various stages of the process between raw data and a lossy compression media stream. In their work [11] Wu and Kuo discuss combined lossy compression and encryption for video and image and already show the significant gain of performance compared to full encryption.

Other Multimedia Encryption Strategies

Besides the concept of partial encryption discussed here there are other encryption methods specifically addressing multimedia content. This includes robust encryption [19] where the goal is to establish encryption resistant to media processing like lossy compression or transfer errors.

By its name, visual cryptography [18] could also be confused with partial encryption for images, but here the idea is rather the representation of the one-time-pad concept in the visual domain.

Research Goal

As discussed above, partial encryption offers significant advantages in comparison to full encryption, especially in the DRM scenario where security requirements are limited (or better: where weak security is commonly accepted due to the lack of secure key distribution). Most often DRM scenarios are asymmetric: Encryption is done at a server where virtually unlimited computational power is available while decryption takes place a a client with limited power, especially in the case of mobile devices. An important aspect of partial encryption here is that the computational costs of decryption at the end user device can become significantly lower compared to full decryption.

Therefore we aim at providing an efficient way to offer partial encryption for MP3 with a significantly lower computational cost at decryption compared to standard full AES decryption as usually applied in DRM schemes like the Microsoft DRM packaging scheme.

Own Design

The implementation was done by modifying the source code of LAME, one of the most common mp3 codecs. At program call the user can specify the –encrypt_sfc or –decrypt_sfc flag followed by a 256 bit (32 byte) key to either encode a source file to MP3 and encrypt it or decode an MP3 file to WAV and decrypt it.

A PCM audio file is converted into an encrypted mp3 file in the following manner:

- Compression: Standard MP3 conversion of PCM data by LAME is executed until frame is to be written.
- Bit stream generation: Our modification grabs the selected parts of the file and hands it over to encryption. In this state these parts (SFC and SF) are variables directly accessible; there is no need for parsing.
- Encryption: Encryption of the selected parts of the MP3. In the example implementation this is done with a key-depended one time pad.
- Writing: MP3 bit stream with encrypted part is written to hard disk.

Decryption of an encrypted mp3 to a PCM audio stream requires the following steps

- Parsing: File is loaded and interpreted by LAME until encrypted parts are reached. These encrypted parts are again directly available from Lame.
- Decryption: Encrypted parts are decrypted, allowing handle the frame correctly.
- PCM computation: LAME decodes MP3 to PCM data
- Streaming: PCM Data stream is provided

For encryption and decryption we utilize the AES part of the open source PolarSSL library. To encrypt random amounts of bits (AES only provides encryption of bytes) we generate a one time pad by encrypting a fixed sequence of bits with the secret AES key and then XOR the required number of bits of the mp3 data with it.

Encryption target

In our implementation we focus on scale factor encryption. They are known to be a suitable target for partial encryption as they have a strong impact in perceived quality when modified but only take approximately 3% of an MP3 file. We use two option here: Encryption of the scale factors (SF) as well as of the scale factor compression variables (SFC).

SF in mp3 is used for countering quantization noise. Encrypting SF therefore causes an increase of quantization noise, reducing the perceived quality, but not making it hard to listen to the audio. It can be seen as a sort of transparent encryption or preview quality. There are up to 4 bit of information per channel, granule and band.

SFC controls the number of bits used for the individual scale factors. Encryption of SFC leads to the CoDec using a random number of bits during decoding, resulting in false decoding of frames due to wrong assumptions of bit allocations. The audio artefacts caused by this are bursts of noise, while in the background the audio file can still be recognized. Actual consumption for entertainment is not realistic in this state. SFC is a table-lookup information. For each channel and granule, there are 4 bits of information. Therefore, for a common stereo file, 16 bit of data are used per frame.

To compute the percentage of the mp3 to be changed by SFC encryption, we need take a typical frame rate of 160kbps and a sampling rate of 44.1kHz. A frame consists of 1,152 frames per channel. This makes 38.28 frames or a maximum of 612.48

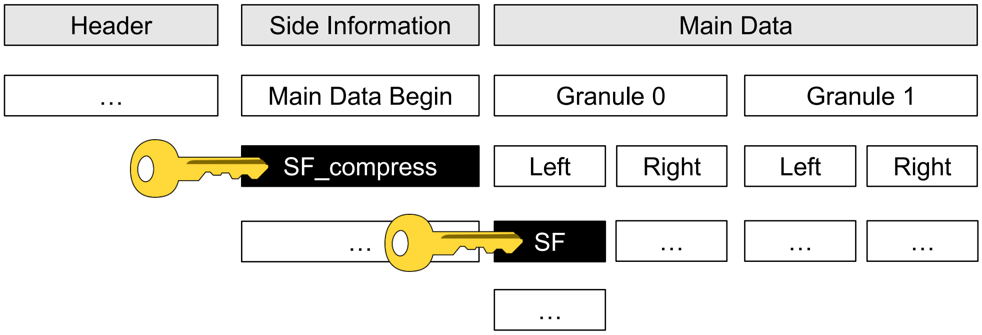

Figure 2. *Scale factors and scale factor compression information are candidates for partial encryption. They only make a small percentage of the overall mp3 data.*

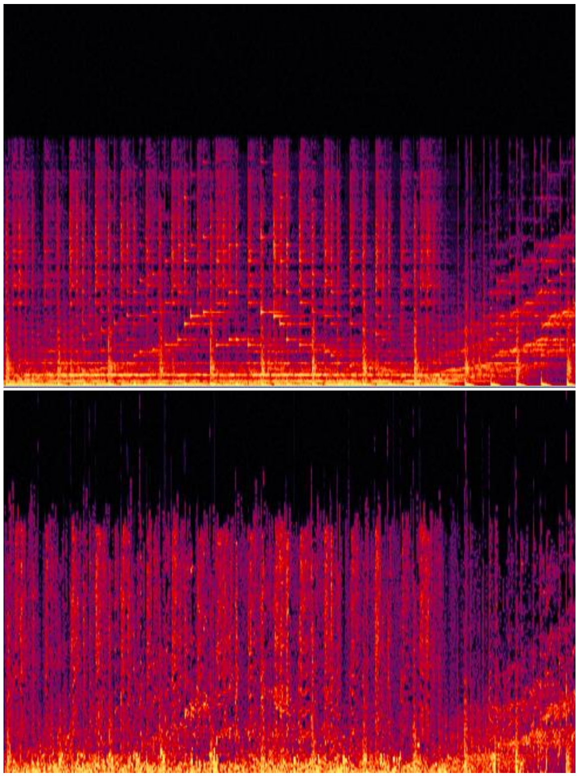

Figure 3. *Top: Original spectrum for 10 seconds of audio. Bottom encrypted spectrum. Bursts of noise are visible.*

(38.28 x 16) bit of SFC information per second. Therefore SFC makes roughly 0,4% of an mp3 frame.

The resulting audio quality is scrambled in both cases. SFC encryption has a stronger impact than SF encryption. Sometimes the resulting audio structure could be sufficient to guess a song title. As an access control mechanism preventing playing back MP3 files for entertainment, the loss of perceived quality is more than sufficient.

File size

In some cases partial encryption leads to size increase of the encrypted files compared to unencrypted ones. One major reason behind this is the encryption of quantized data. Quantization in lossy audio compression is applied to achieve a trade-off between data faithfully representing the original audio samples and creating data that can be compressed well with standard entropy compression algorithms. In mp3, entropy compression is executed by Huffman-encoding the quantized sample data. Scale factors and side information are not compressed in this way, therefore encryption of this data has no effect on the file size. This of course requires bit-wise encryption.

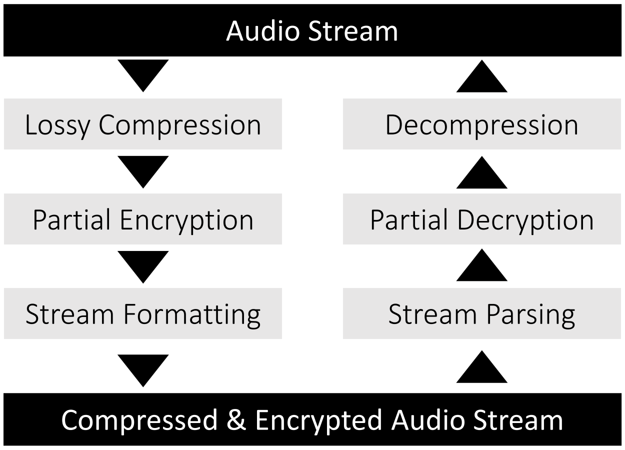

Figure 4. *Assumed process chain for a maximum efficiency gain by partial encryption. Seeking the areas to be encrypted requires no additional time*

A note on encryption

The encryption and decryption schemes applied in the design described above are obviously not optimal: using a one time pad allows decryption of other files encrypted with the same pad if one can access both encrypted and decrypted copies of the first mp3 file.

Using a different AES key for calculating the pad counters this security issue, but requires individual key handling for each file. A better strategy would be to either use an encryption scheme allowing bit-wise stream encryption or to crop the bits beyond the last full byte of the content to be encrypted and use a byte encryption scheme.

A note on efficiency

The efficiency gain of the proposed approach is not a general one; it depends on the overall handling of the encrypted audio file.

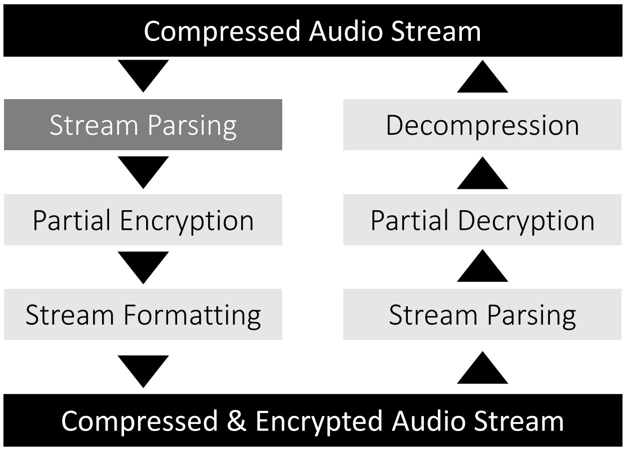

Figure 5. *When processing already compressed audio, additional parsing and often also entropy decompression is required*

When as shown in figure 4 the starting point is a PCM audio file, lossy compression needs to be executed to create a compressed and encrypted audio. In this case the partial encryption can become part of the stream formatting after quantization takes place. Hence no additional position seeking or file processing is required and the overhead is minimal. This approach is also called ”joint compression and encryption” in the literature.

When as in figure 5 an already compressed file is the starting point, file parsing and in some cases also handling entropy compression is necessary. This causes overhead not necessary with full compression.

So the performance gain of the encryption part is more significant when dealing with a PCM file. Still, the overall performance will be better with an already compressed file as file parsing has a lower cost than transcoding, creation of a perceptual model, bit allocation and quantization.

Other audio formats

The concept of scale factors is commonly used in lossy audio compression. Most often the basic idea is to group coefficients of one spectral band, determine the band’s importance by a psychoacoustic model and then quantize the coefficients. Scale factors are usually the information needed to inverse the quantization.

In mp2 audio (still used e.g. in DAB) or AAC scale factors are used in the way described above. Scale factor encryption leads to scrambled audio with random power of the individual frequency bands. Using scale factor encryption in these formats is therefore a good strategy as it reduces the perceived quality largely but does not endanger the correct handling of the audio file due to incorrect size assumptions.

Evaluation

To test the performance of our implementation we used the Linux time command to measure user and system time. We tested normal encoding and decoding with an unmodified version of LAME and with our version. With our version we then tested the performance of encoding with encryption and decoding with decryption for the SFC and the SF encryption based approaches. The tests were performed on a Ubuntu 14.04 LTS OS in a virtual machine with access to 4 GB of RAM and two virtual cores of an Intel Core i5-4200 with 1.6 GHz each. The timing tests were performed 10 times each and averaged, the first part is user time, the second part system time. Finally we tested encrypting/decrypting with 7zip, a popular archiver offering AES256 encryption, to compare the costs of encoding with subsequent encryption and decryption with subsequent decoding. 7zip was executed in copy mode to ensure no additional performance was lost due to lossless compression.

	Encoding	Decoding
Unmodified LAME	0.336 s	0.139 s
Modified LAME (no encryption)	0.387 s	0.138 s
LAME SFC crypting	0.332 s	0.141 s
LAME SF crypting	0.350 s	0.153 s
Lame+AES	0.408 s	0.191 s

Time measurements for the different encryption and decryption schemes. (Time in Linux system time)

For DRM usuage decoding performance is most important. Here the SFC decryption only increases computation time by 1.4% while in comparison additional AES file decoding increases the process by 37.4%.

SFC Analysis

As SFC partial encryption is the suggested approach for efficient mp3 partial encryption, we take a deeper look at the distribution of SFC to ensure that it is not too easy to simply guess the correct SFC and replace the encrypted values.

We took an example mp3 file (195kbps VBR, stereo, 44.1kHz) with 210 seconds duration and extracted all SFC values. The histogram of all values is shown in figure 7. For simplicity, we converted the 8 bits into decimal values from 0 to 255. There are 8 groups visible in the histogram and some bins are empty. So there is no equal distribution of SFC values into all bins, but the present randomness seems to be sufficient to prohibit efficient automated guessing. Still, a detailed numerical analysis would be necessary for a deeper assessment of the security of the encryption.

In figure 8 we show a sequence of 100 SFC values taken from the same file. Here we see that also in consecutive frames randomness is given and no direct dependence seems to occur.

Outlook

In common DRM there is a decision between watermarking and encryption, often called hard or soft DRM. While obviously combining both mechanism would improve the overfall security by providing access control and forensic tracking at the same time, the two mechanisms are usually not applied together.

The reason behind this is the challenge to efficiently mark and encrypt a media file. When full encryption comes first, watermarking in the encrypted domain would be necessary. When watermarking comes first, each marked file needs to be encrypted afterwards, increasing the overhead.

There are approaches allowing both encryption and watermarking without influencing each other, but these do not reflect the state of the art of both individual technologies [15]. This also includes our own previous work on the topic [16].

With partial encryption and the concept of pre-marking chunks of the media file and then assembling it out of the set

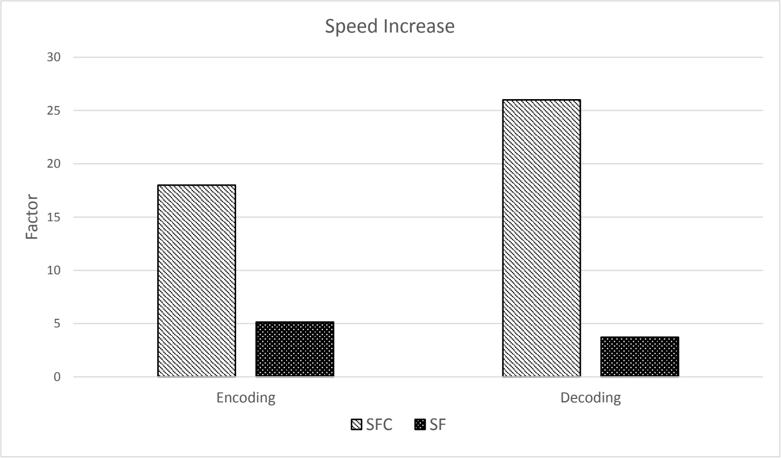

Figure 6. *This figure shows the reduced computational cost for SFC and SF encryption compared to full AES encryption. The y-axis tells how much faster than full AES encryption the different schemes are.*

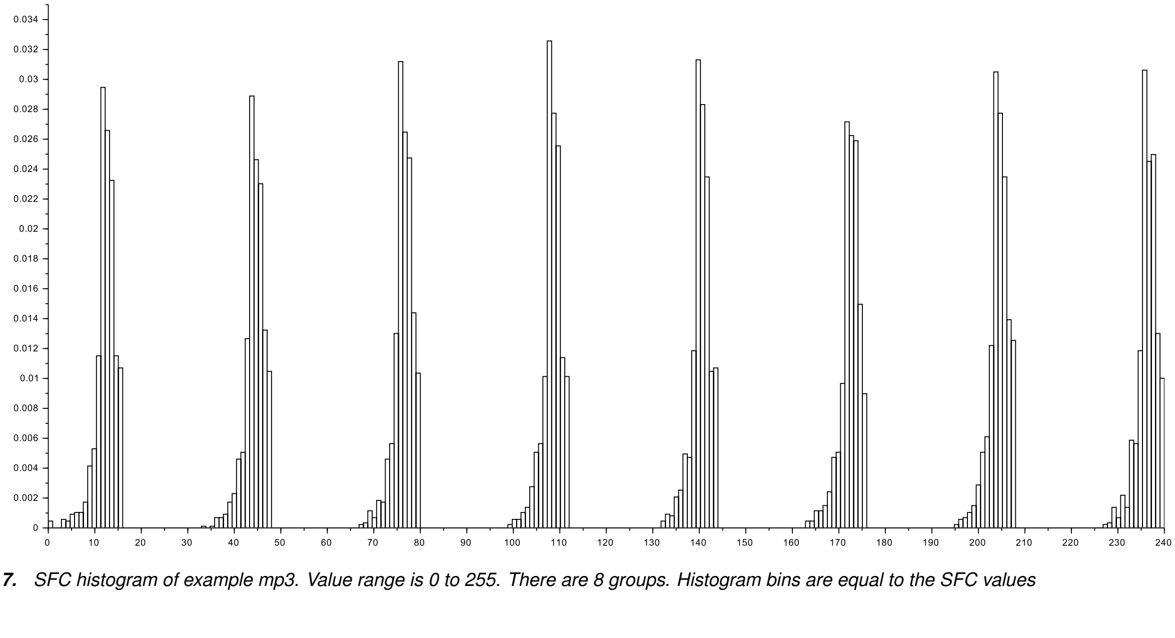

Figure 7. *SFC histogram of example mp3. Value range is 0 to 255. There are 8 groups. Histogram bins are equal to the SFC values*

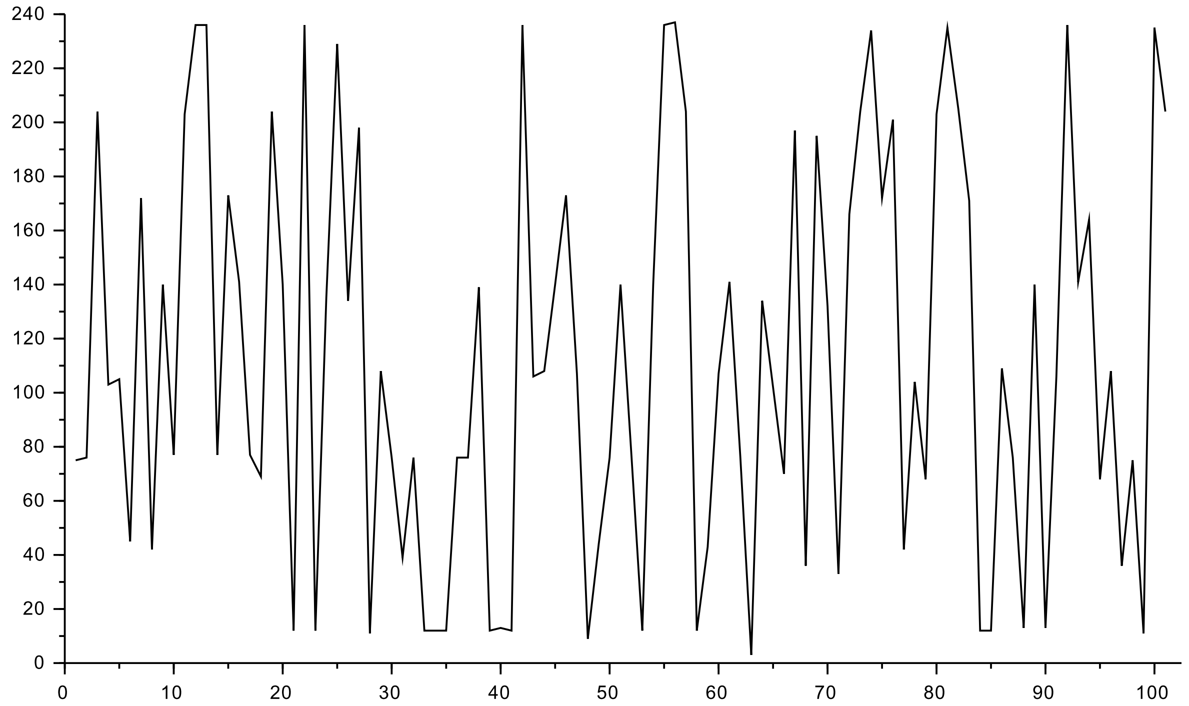

Figure 8. *Plot of 100 consecutive SFC values. A high variance is clearly visible. X=value#, Y=SFC value*

of prepared chunks rapidly, called container watermarking in our

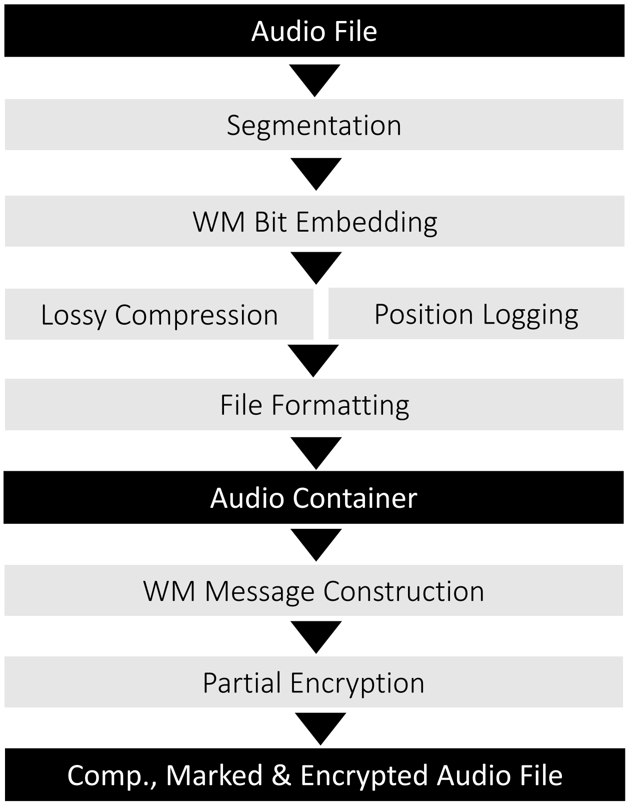

Figure 9. *Partial encryption allows the efficient combination of watermarking and encryption in DRM scenarios.*

Future Work

Security analysis of the presented approach is superfluous at best right now. One could argue that in DRM applications this is not really necessary, but on the other hand a security failure like in CSS should be prevented.

Full integration of partial encryption into a showcase would also be helpful. After we have shown the potential of saving computational power, it would be interesting to see how much impact a light-weight DRM has on typical power consumption and therefore time-before-charge of mobile devices.

Acknowledgments

This work was supported by the German Federal Ministry of Education and Research (BMBF) as well as by the Hessen State Ministry for Higher Education, Research and the Arts (HMWK) within CRISP.

Implementation of the partial encryption scheme was executed by Florian Wendel during his Multimedia Security P4 at TU Darmstadt.

References

[1] A. Servetti, C. Testa, J. Carlos, and D. Martin. 2003. Frequency-Selective Partial Encryption of Compressed Audio,International Conference on Audio, Speech and Signal Processing, Hong Kong

[2] L. Gang, A. N. Akansu, M. Ramkumar, and X. Xie. 2001. Online music protection and MP3 compression. In Proceedings of International Symposium on Intelligent Multimedia, Video and Speech Processing

[3] Lian, Shiguo. Multimedia content encryption : techniques and applications, 1st ed. ISBN 9781420065275, Auerbach,2009

[4] L. Gang, A.N. Akansu, M. Ramkumar und X. Xie, Online music protection and MP3 compression, in: Proc. of Int. Symposium on Intelligent Multimedia, Video and Speech Processing, May 2001, pp. 1316

[5] Goo-Rak Kwon, Chuntao Wang, Shiguo Lian, Suk-seung Hwang. Advanced partial encryption using watermarking and scrambling in MP3;Multimedia Tools and Applications August 2012, Volume 59, Issue 3, pp 885-895

[6] Chih-Hsu Yen, Hung-Yu Wei, and Bing-Fei Wu. New Encryption Approaches to MP3 Compression; WSEAS Conferences, Venice, Italy, 2004

[7] J. A. Bloom, I. J. Cox, T. Kalker, J. P. M. G. Linnartz, M. L. Miller and C. B. S. Traw, ”Copy protection for DVD video,” in Proceedings of the IEEE, vol. 87, no. 7, pp. 1267-1276, Jul 1999. doi: 10.1109/5.771077

[8] Guadamuz, Andrs, Trouble with Prime Numbers: DeCSS, DVD and the Protection of Proprietary Encryption Tools. Journal of Information, Law & Technology, Vol. 3, 2002.

[9] Meyer, J., and F. Gadegast. Security Mechanisms for Multimedia Data with the Example MPEG-1 Video, Project Description of SECMPEG, Technical University of Berlin, Germany, 1995

[10] Fuwen Liu, Hartmut Koenig, A survey of video encryption algorithms, Computers & Security, Volume 29, Issue 1, February 2010, Pages 3-15, ISSN 0167-4048

[11] Chung-Ping Wu and C. C. J. Kuo, ”Design of integrated multimedia compression and encryption systems,” in IEEE Transactions on Multimedia, vol. 7, no. 5, pp. 828-839, Oct. 2005.

[12] Brandenburg, Karlheinz. ”MP3 and AAC explained.” Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding. Audio Engineering Society, 1999.

[13] Steinebach, Martin, Sascha Zmudzinski, and Fan Chen. ”The digital watermarking container: secure and efficient embedding.” Proceedings of the 2004 workshop on Multimedia and security. ACM, 2004.

[14] Wolf, Patrick, Enrico Hauer, and Martin Steinebach. ”The video watermarking container: efficient real-time transaction watermarking.” Electronic Imaging 2008. International Society for Optics and Photonics, 2008.

[15] V. C. Prasad and S. Maheswari, ”Robust watermarking of AES encrypted images for DRM systems,” Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), 2013 International Conference on, Tirunelveli, 2013, pp. 189-193. doi: 10.1109/ICE-CCN.2013.6528490

[16] Steinebach, Zmudzinski, Blke; Audio watermarking and partial encryption, Proceedings of SPIE - Volume 5681, Security, Steganography, and Watermarking of Multimedia Contents VII, Edward J. Delp III, Ping W. Wong, Editors, ISBN: 0819456543, SPIE, Bellingham, USA,2005

[17] Qiong Liu, Reihaneh Safavi-Naini, and Nicholas Paul Sheppard. 2003. Digital rights management for content distribution. In Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003 - Volume 21 (ACSW Frontiers ’03), Chris Johnson, Paul Montague, and Chris Steketee (Eds.), Vol. 21. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 49-58.

[18] Naor, Moni, and Adi Shamir. ”Visual cryptography.” Workshop on the Theory and Application of of Cryptographic Techniques. Springer Berlin Heidelberg, 1994.

[19] Piao, Yong-Ri, Dong-Hak Shin, and Eun-Soo Kim. ”Robust image encryption by combined use of integral imaging and pixel scrambling techniques.” Optics and Lasers in Engineering 47.11 (2009): 1273-1281.

Author Biography

Dr. Martin Steinebach is the manager of the Media Security and IT Forensics division at Fraunhofer SIT. From 2003 to 2007 he was the manager of the Media Security in IT division at Fraunhofer IPSI. He studied computer science at the Technical University of Darmstadt and finished his diploma thesis on copyright protection for digital audio in 1999. In 2003 he received his PhD at the Technical University of Darmstadt for this work on digital audio watermarking.