First published in the Proceedings of the 18th European Signal Processing Conference (EUSIPCO'10) in 2010, published by EURASIP.

(IN)SECURE MULTIMEDIA TRANSMISSION OVER RTP

Thomas Stütz and Andreas Uhl

Department of Computer Sciences
University of Salzburg
Jakob-Haringer-Str. 2
Salzburg
Austria
{tstuetz, uhl}@cosy.sbg.ac.at

ABSTRACT

Multimedia encryption is an intensely discussed topic where numerous approaches have been proposed. However, the concrete application scenario and the applicable notion of security of many proposed approaches is mostly not stated precisely or even at all. In this work security notions and application scenarios for multimedia encryption are discussed and a previously proposed multimedia encryption approach, i.e., RTP-header encryption, is analysed with respect to these scenarios and notions. We show that this selective encryption approach is insecure for sensible settings and for all sensible notions of security. The analysis focuses on the RTP packetization of H.264:AVC.

1. INTRODUCTION

"There are two kinds of cryptography in this world: cryptography that will stop your kid sister from reading your files, and cryptography that will stop major governments from reading your files."[9] This paper is about the former, i.e., we show that a previously presented approach [1] as well as extensions of that approach do not offer "real" security. The proposed approach is a selective encryption scheme [11, 4], that selectively encrypts RTP header data. It is the only approach (the authors are aware of) that selectively encrypts "network" protocol header data. On the opposite, SRTP (RFC 3711) only encrypts the RTP payload data and preserves the RTP header. Selective encryption has been extensively discussed for multimedia encryption [11, 4, 7]. Selective encryption requires to rethink the notion of security, as it can clearly never meet strong cryptographic security notions, which require that every information about the plaintext can not be computed from the ciphertext [5]. Naturally if only a fraction is encrypted, this security definition can not be met. Security notions for multimedia (selective) encryption are discussed in section 2. The multimedia transmission / distribution system considered in this work is outlined in section 3. The proposed encryption approach for this multimedia transmission system and its extensions are discussed in section 4. In section 5 we show that a previously presented RTP-encryption scheme is insecure for rather obvious and trivial reasons. We also show that even reasonable extensions of the approach still offer very limited security. In order to clarify our security analysis and evaluation we in-depth discuss security notions for multimedia encryption. In order to pin down the general to the concrete, the paper focuses on RTP transmission of H.264:AVC on top of a (DSL / PPPoE / WLAN / Ethernet) / IPv4 / UDP transmission system.

2. SECURITY NOTIONS FOR MULTIMEDIA ENCRYPTION

The application environment of multimedia cryptosystems often does not have the same stringent security requirements as assumed for conventional cryptosystems. For conventional cryptosystems the security notion of MP security (message privacy) is commonly assumed to be the best formalization of the actual security requirements [2]. If a system is MP-secure an adversary can not efficiently compute any property of the plaintext from the ciphertext. An exception is the message length, i.e., the number of bytes of plain and ciphertext. It is also agreed upon that for some applications the notion of MR security (message recovery) would be sufficient [2], where an adversary tries to recover the plaintext message in its entirety. Conventional security notions only consider a single message space in the formalization of a security notion. For multimedia encryption the setup is a bit more complex, as we do not only have to consider a single message space, rather we have to consider the space of raw multimedia data (media domain, i.e., raw video), on which similarity, fidelity and quality of the data is defined, and the space of compressed and coded multimedia data (compressed domain, i.e., bitstreams), which is commonly encrypted. A scheme, that performs compression (which results in a variable length bitstream) and afterwards encryption, is not MP secure on the raw multimedia data space. However, most commonly MP security on neither the raw domain nor on the compressed domain is a requirement for multimedia systems, but message recovery is far too weak as different raw and compressed datums can represent very similar, even visually indistinguishable reconstructions. Thus for multimedia security, we define security by the inability of an adversary to efficiently compute an approximation of the plaintext with a quality higher than targeted and refer to this notion as MQ security (message quality). The interpretation of quality may differ according to the targeted application scenario. Common application scenarios [3] are content confidentiality (no visual information of the content shall be discernible in the approximation) and sufficient encryption, which targets to sufficiently reduce the visual quality such that the business value of the media data is secured (e.g., encrypted videos shall not be pleasantly watchable). Similar sketches of multimedia security notions can be found in literature [8].

In the security analysis section 5 we primarily consider whether content confidentiality and sufficient encryption can be achieved.

2.1 Security Notions for Selective Encryption

We think that selective encryption does not require specific security notions, rather the security notions for multimedia apply (MQ security). It has been frequently argued that the selectively encrypted data must be unpredictable from the ciphertext [7], however this property is not sufficient. The encrypted data must also be of absolute necessity for the computation of the approximation. As we will see in section 5 this is definitely not the case for the RTP-header data, although the data is not efficiently computable from the ciphertext.

3. MULTIMEDIA TRANSMISSION OVER RTP

In this work we consider a multimedia transmission system, which is built on IPv4 (RFC 791), UDP (RFC 768), and RTP (RFC 3550). The underlying link and network layer is assumed to be WLAN, Ethernet, PPPoE and DSL, but only the maximum transmission unit (MTU) of these systems is of interest in the scope of this work. In this work our focus is on H.264:AVC video streams [6]; the packetization of H.264:AVC in RTP is standardized as well (RFC 3984).

3.1 Media Codec, H.264:AVC

The video codec H.264:AVC compresses raw video data (consisting of pictures) to a bitstream, which in its most basic form solely consists of NALUs (network abstraction layer units). Depending on the type of data contained in a NALU, the NALU header is formatted, only the forbidden_zero_bit (F in figure 4) always has to be equal to zero. The semantics for nal_ref_idc (NRI) are precisely defined in [6, sect. 7.5.1], in short it shall be equal to zero for less important data and shall not be zero for important data, such as IDR frames (instantaneous decoding refresh, comparable to I-frames in previous MPEG-standards) and sequence parameter sets (SPS) and picture parameter sets (PPS). SPS and PPS contain the necessary information (profile, color mode, bit-depth of colors, resolution, ...) to decode NALUs from the video coding layer (VCL). Important NUTs (NAL unit types) are 7 (SPS), 8 (PPS), 5 (coded slice of an IDR-picture), and 1 (coded slice of a non-IDR picture). A non-IDR picture is either a P-picture (predicted from a single reference picture) or a B-picture (bi-predicted from two reference pictures).

3.2 RTP Packetization

The NALUs can be encapsulated in RTP employing RFC 3984. Most important for the context of this work is whether network or application layer fragmentation is performed. Network fragmentation means that the NALUs are directly encapsulated leaving the possibly necessary adaptation of the network packets to the size of the MTU (maximum transmission unit) to the network layer (commonly IP). Application layer fragmentation means that the NALUs are either aggregated or fragmented on the application layer to form a RTP packet for transmission. The goal is to produce RTP packets close, but below the smallest MTU in the transmission system. The RTP header fields V, P, X, and CC are of minor importance for the scope of this work. The PT field indicates the payload type, which may also be employed to separate different streams with different payload types, i.e. differently coded. Most important is the sequence number, which is chosen randomly for the first RTP packet and afterwards incremented for each packet sent (see section 5.1 of RFC 3550). The timestamp gives the presentation time of the coded data contained in the RTP packet and can be used to synchronise streams (e.g., video and audio). All RTP packets with the same SSRC (see figure 3) share the same timing and sequence number space, and can be used to separate different streams from the same participant within the same RTP session, e.g. separate video cameras. The CSRC field is optional (only present if indicated by CC) and is only used if RTP mixers are present, which is very uncommon. Thus the CSRC field is irrelevant for the application case we are looking at, namely a client-server-streaming-scenario (the most common real-world application case). Additional streams, most commonly for audio, are sent in a separate RTP session, which is indicated by different UDP ports (see section 2.2 of RFC 3550). The UDP header is given in figure 2.

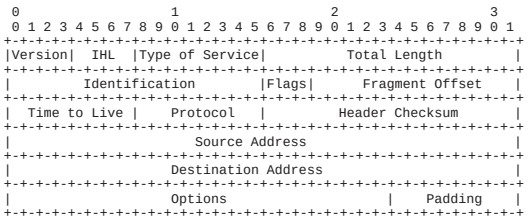


Figure 1: IPv4 header

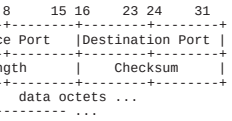


Figure 2: UDP header

3.3 Network and Transport Layer

In our application case, the UDP packet is then encapsulated in an IPv4 packet (see figure 1). Most interestingly is the Identification field, which has to be chosen in a way to ensure that fragments of different datagrams are not mixed.
Further relevant documents for a broader coverage of the topic of multimedia streaming over RTP are RFC 4566 (SDP, session description protocol), and RFC 2250 (the RTP payload format for MPEG1/MPEG2).

4. RTP-ENCRYPTION FOR (IN)SECURE MULTIMEDIA TRANSMISSION

The basic idea is to only selectively encrypt some RTP header fields. The assumption is that these fields are crucial for the

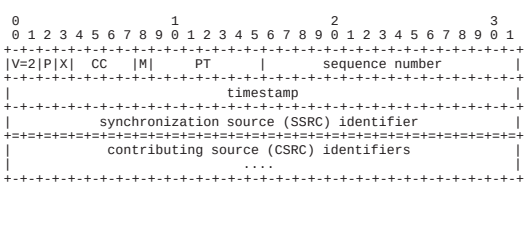


Figure 3: RTP header

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|F|NRI|    NUT  |
+-+-+-+-+-+-+-+-+
```

Figure 4: NAL unit header

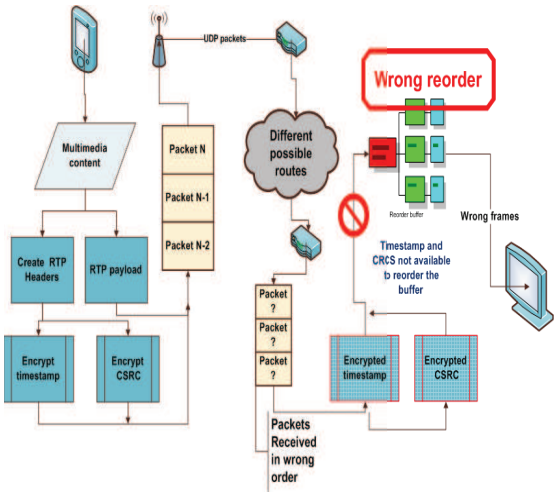


Figure 5: The (in)secure transmission scheme taken from [1]

assembly of the RTP packets and thus the decoding of the contained video stream. In [1] it is proposed to encrypt the timestamp and the CSRC field of the RTP header (see figure 5), it is argued that reordering the stream is hard without the timestamp. In the following we extend the approach of [1] in a way that could be considered to improve security (which is not the case as we will show later), namely we encrypt all of the RTP header data. The question is whether in-order transmission of RTP packets with encrypted headers is appropriate for secure multimedia transmission. The RTP-full header encryption scheme could be extended by an explicit RTP-packet permuter, which guarantees out-of-order transmission of the RTP packets.
The following distinct schemes are considered

1. encryption of the timestamp and CSRC field of the RTP header [1],
2. encryption of the entire RTP-header, and
3. permutation of the RTP packets (i.e., RTP packet index permutation) together with the encryption of the entire RTP header.

Another considerable aspect is whether network or application layer fragmentation is performed, i.e., IP fragmentation or RTP/H.264:AVC fragmentation.

5. ANALYSIS OF RTP-ENCRYPTION

The selective encryption and possibly permutation of RTP packets has several points of attack which can be categorized by the layer in which they operate:

- Network layer (IPv4)
  E.g. the identification field reveals the original order of the packets (Linux kernel 2.6.24). The source and destination address help to identify an RTP session.
- Transport layer (UDP)
  The ports and the IP addresses are unique for an RTP session.
- Application layer (RTP)
  The sequence number and the timestamp are not independent, the payload type can be exploited to separate streams.
- Packetization layer (RTP Payload format for H.264)
  Aggregation produces a packet which contains several NALUs in the correct order (see RFC 3984). Fragmentation packets have an fragmentation packet header, which can help to order the fragments of an NALU.
- Video codec layer (H.264:AVC)
  The H.264 bitstream also introduces order on the NALU sequence, there are several fields in the slice header that can be exploited to sort the NALU stream. A VCL NALU can be decoded given only PPS and SPS, which recovers at least IDR-pictures perfectly and also the residual information of a non-IDR picture. If a VCL NALU (e.g., with NUT 1 or 5) is fragmented, its correct decodability can be exploited to sort its fragments.
- Raw data layer (Properties of the reconstructed visual data, e.g., visual difference of consecutive frames, blockiness)
  We assume that if the decoding order is correct the MSE of two consecutive frames is minimal. Also if the decoding order of NALUs is wrong, blocking artifacts in the reconstruction are very likely to occur.

Thus there are plenty of points of attack, so the question rather seems to be how deep do an attacker has to go to break the RTP-header encryption and permutation, as whether it is possible.

Experimental Details

The experimental results have been derived with the following software: the H.264 reference implementation (JM 16.0), x264 (0.57.x), and custom software for NALU processing. The arguments on video streaming with RTP have been reality-checked with the popular video-streaming software, VLC (v.0.8.6e). All evaluations have been performed on a standard 32-bit Linux (Ubuntu 8.04) with kernel 2.6.24.

5.1 Encryption of Timestamp and CSRC Field of the RTP Header

The assumption is that these fields are crucial for the assembly of the RTP packets and thus the decoding of the contained video stream. A wrong assumption given that the sequence number is left in plaintext, which enables to effectively sort the RTP packets. Also the argument that RTP streams can no longer be separated as the CSRC field is encrypted is simply wrong. The CSRC field is optional (only used with RTP mixers, i.e., commonly not used). Thus the approach is completely insecure for obvious reasons. The only inconvenience is that the frames per second of the video stream have to be guessed (given that the fps are commonly around 30 quite an easy task) and that audio streams need to be aligned (a task commonly performed by users as video and audio tracks are frequently misaligned). Thus this scheme [1] is insecure for all relevant notions of security, even MR security (message recovery) on the raw media domain, i.e., the video can be entirely reconstructed.

5.2 Encryption of the Entire RTP Header

Even if the entire RTP header is encrypted, this scheme assumes that the network securely shuffles the packets, an assumption which is presumably not justified and there is no evidence for it presented in [1]. Even if we assume that the network securely shuffles the packets, different RTP sessions can be differentiated by different UDP ports and the order of the RTP packets can be trivially reconstructed if the IPv4 implementation chooses to reflect the order of the RTP packets in its selection of the IPv4 identification field. This is the case on the authors' test streaming system (VLC 0.8.6e, Ubuntu 8.0.4, Linux kernel: 2.6.24) and has been experimentally verified with wireshark (v1.0.0). If by unlikely chance the network sufficiently shuffles the RTP packets (it can not be guaranteed that the network actually delivers packets out-of-order) and the original order is not leaked by the IPv4 header (e.g., in the case of IPv6) then the analysis of this scheme is similar to the analysis of the permutation and header-encryption of RTP packets.

However, in our real-world application case this scheme is insecure with respect to the relevant security notions, MQ and even MR on the raw media domain. Thus it can definitely not be applied for content confidentiality and sufficient encryption.

5.3 Permutation and Header-Encryption of RTP Packets

This scheme is the only one, which could actually be hard to break. One could even claim that the complexity of sorting the RTP packets is of $\mathcal{O}(n!)$, where n is the number of permuted packets. This claim does not hold if we assume that a decoding system is capable to identify wrong partial decoding orders and correct partial decoding orders (a partly valid assumption for H.264 bitstreams as we will show later). It is highly likely that a decoding system can differentiate between random sequences and correctly formatted compressed data, due to the stringent syntactical and semantical requirements imposed by the format. In that case the exponential complexity of finding the correct permutation ($\mathcal{O}(n!)$) is reduced to far more feasible complexity of $\mathcal{O}(n^2)$. The following simple algorithm with this worst case complexity finds the correct permutation:

```
sort( $permuted_packets ){
  $tmp_packets = {} ;
  while
  ($packet = take_from_front( $permuted_packets ))
  {
    append_at_end( $tmp_packets, $packet );
    if ( !decodable( $tmp_packets ) )
    {
      take_from_end( $tmp_packets );
    }else{
      append_at_end( $permuted_packets, $packet );
    }
  }
  return $tmp_packets;
}
```

For our specific application case with H.264:AVC, we first consider the case of network fragmentation.

5.3.1 Network Fragmentation

In that case entire NALUs can be reconstructed. The SPS and PPS sets have to be found (trivial as they are indicated by the NUT in NALU header, see figure 4) and then IDR-pictures can be decoded. Thus content confidentiality can not be achieved. Also the non-IDR-frames can be decoded with zero reference pictures, which results in some edge information of the original picture (see figure 6(a)). NALU sequences containing only one IDR-picture and several P-pictures can be effectively sorted. Thus if the number of permuted packets is shorter than the number of P-pictures the NALU sequence can be effectively sorted on the basis of H.264 slice header fields. We have experimentally verified this (see section 5). For I (P B)* sequences, i.e, sequences consisting of an IDR-picture followed by a P-picture and a B-picture, some quality reductions may have to be accepted. If the permutation ranges over several IDR-pictures, the properties of the reconstructed visual data have to be taken into account, most relevant is the MSE / PSNR of two consecutive pictures. We assume that the correct decoding order minimizes the overall sum of MSE difference between consecutive frames. Figure 7 and 8 show the reconstruction with either a correct or an incorrect P-pictures (from adjacent I-pictures). Thus to break the scheme with respect to full message recovery on the raw visual data (MR-security) is harder, though not impossible.

In summary, the scheme is insecure for all security notions if, as common, sequences with many P-pictures are employed, e.g., the default configuration of the heavily employed open source codec x264 uses 250 P-pictures. In a real-world streaming scenario it is impractical to use more than 3-4 seconds of the video bitstream in the permutation process, i.e., 120 to 160 RTP packets. Thus for real-world application the scheme can not offer any security at all.

5.3.2 Application Layer Fragmentation

This case is the hardest case to break. For application layer fragmentation the size of the MTU has to be considered, table 1 summarizes most of the practically relevant MTUs. Larger NALUs (than the MTU) are split into several packets. The NALU size does mainly depend on the properties of the raw visual data (compressibility, resolution) and the compression options (quality, i.e. quantization parameter). Each NALU with a length above the MTU is split into fragments at most as big as the smallest MTU. The MTU also has to include the overhead of the underlying protocols, in our application case a maximum size of 1400 byte for a NALU fragment is a safe choice. Thus only NALUs above 1400 bytes are split into fragments.

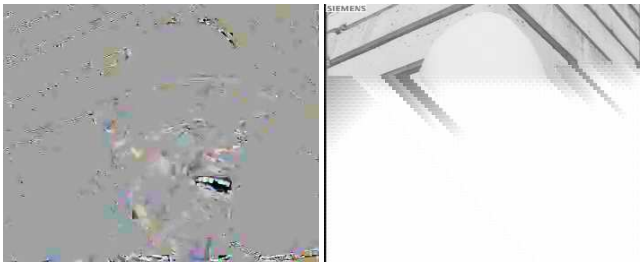
SPS and PPS are very unlikely to be split as they are small (see section 5). If RFC 3984 is used with FU-B fragmentation packets one can identify fragments of the same NALU. Thus there is only one question: can we sort the fragments of a VCL NALU by decodability constraints? If this is the case then we can proceed as in the network fragmentation case.

If RFC 3984 is used with FU-A fragmentation packets one can identify whether a packet was fragmented and whether it is the start or the end of a fragmented packet. Also the NUT of the original unfragmented NALU is contained in every single fragment. The question is not only whether we can sort the fragments of a VCL NALU, but whether the decodability check is robust enough to reliably identify fragments of other NALUs.

Given that we can reconstruct partial picture information, content confidentiality can not be achieved even if application layer fragmentation is employed. It is also doubtful that sufficient encryption can be implemented securely.

Link	MTU
PPPoE (DSL)	1492
Ethernet	1500
Gigabit	9000
WiFi 802.11	2312

Table 1: MTU sizes



(a) P-picture (NALU 13) decoded without reference picture (b) I-picture partially decoded

Figure 6: Foreman: partial decoding attempts

Figure 7: Foreman: correct P-frame, PSNR to previous frame 36.06dB

Figure 8: Foreman: wrong P-frame, PSNR to previous frame 19.45dB

6. CONCLUSION

RTP-header encryption and RTP packet permutation can not reliably offer security, which has been analysed and evaluated for RTP transmission of H.264/AVC. It is assumed that analogue attacks can be derived for any payload format, i.e., video or audio (compression) format. In general approaches that permute bitstream fragments are very likely to offer very limited security or none at all. Attacks against such schemes can exploit information leakages on many levels, most importantly on the bitstream level and in the raw multimedia domain, which will often enable to recover the entire plaintext. In conclusion, such schemes do not seem very promising.

REFERENCES

[1] F. Almasalha, N. Agarwal, and A. Khokhar. Secure multimedia transmission over RTP. In *Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM'08)*, Berkeley, CA, USA, Dec. 2008. IEEE Computer Society.

[2] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. Format-preserving encryption. In *Proceedings of Selected Areas in Cryptography, SAC '09*, volume 5867, pages 295–312, Calgary, Canada, Aug. 2009. Springer-Verlag.

[3] D. Engel, T. Stütz, and A. Uhl. A survey on JPEG2000 encryption. *Multimedia Systems*, 15(4):243–270, 2009.

[4] B. Furht, E. Muharemagic, and D. Socek. *Multimedia Encryption and Watermarking*, volume 28 of *Multimedia Systems and Applications*. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 2005.

[5] O. Goldreich. *The Foundations of Cryptography*. Cambridge University Press, 2001.

[6] ITU-T H.264. Advanced video coding for generic audivisual services, Nov. 2007.

[7] T. D. Lookabaugh and D. C. Sicker. Selective encryption for consumer applications. *IEEE Communications Magazine*, 42(5):124–129, 2004.

[8] Y. Mao and M. Wu. A joint signal processing and cryptographic approach to multimedia encryption. *IEEE Transactions on Image Processing*, 15(7):2061–2075, July 2006.

[9] B. Schneier. *Applied cryptography (2nd edition): protocols, algorithms and source code in C*. Wiley Publishers, 1996.

[10] T. Stütz and A. Uhl. Format-compliant encryption of H.264/AVC and SVC. In *Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM'08)*, Berkeley, CA, USA, Dec. 2008. IEEE Computer Society.

[11] A. Uhl and A. Pommer. *Image and Video Encryption. From Digital Rights Management to Secured Personal Communication*, volume 15 of *Advances in Information Security*. Springer-Verlag, 2005.