Journal of Information Security and Applications 40 (2018) 183–192

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Secure transmission of multimedia contents over low-power mobile devices

Fadi Almasalha[a], Farid Naït-Abdesselam[b,c,*], Goce Trajcevski[c], Ashfaq Khokhar[c]

[a] Applied Science Private University, Al Arab Street 21, Amman, Jordan
[b] Paris Descartes University, LIPADE 45 Rue des Saints Pères 75006, Paris, France
[c] Iowa State University, Electrical and Computer Engineering Department, 2215 Coover, Ames, IA 50010, USA

ARTICLE INFO

Article history:

Keywords:
Security
Multimedia communications
Low-power
Mobile devices

ABSTRACT

Secure transmission of multimedia contents is computationally challenging due to the sheer volume of data involved, and achieving this by fully encrypting multimedia contents is not a viable solution for mobile devices, as they run on limited battery power and employ relatively slower processors compared to their desktop counterparts. Also, in some particular application scenarios, the value, secrecy, and/or privacy of the information is time dependent and short lived. This paper presents a highly scalable approach for securing multimedia streams on mobile devices by encrypting intelligently only selected portions of the bit stream. More importantly, the proposed solution works mainly on compressed bit stream and does not require any media decoding. It encrypts on the average less than 3% of a packet load and provides robust security equivalent to that of a fully encrypted bit stream. This solution has been implemented on desktop, laptop, netbook and Nokia N series platforms. The proposed scheme is approximately 15 times faster (on average) and processes 8 to 12 times more information for a given battery life when compared to a state of the art full encryption based solutions.

1. Introduction

With the emergence of online multimedia savvy services, such as social networks, video conferencing, and voice over IP, Internet traffic has been rapidly transformed from being pure text-oriented to multimedia-oriented. Unprotected Internet traffic can be easily sniffed and manipulated by hackers without much difficulty, underscoring the vital role of security and privacy tools in public networks. Not only private data is under attack by hackers, the new technologies also allow redistribution of copyrighted and commercially valuable data, also referred to as super distribution [1].

The industry standard for the protection of online data storage and transmission has emerged in the form of Virtual Private Networks (VPNs). The point to point secure tunnels, such as IPSEC, are used to create virtual networks, and can be configured to encrypt all the data transmission between the two end users. While these solutions may address sniffing related security issues, they may not be feasible for low powered mobile devices. The overhead of encryption algorithm can be reduced if not all the data is encrypted.

Numerous approaches have been proposed that are based on selective encryption of multimedia contents [2–4]. These approaches are media dependent and encrypt principle information units, for example I-Frames, motion vectors (in the case of MPEG coded bit streams), application headers, etc. Most of these solutions either require decoding of the bit stream or become explicit part of the encoder/decoder modules, thus requiring changing of the standard encoder/decoder modules. An ideal approach would be to develop a seamless selective encryption scheme, wherein the selective encryption treatment applied to the packets is transparent to end users. Also, such a scheme should have minimal effect on the quality and scalability of services offered to the authorized end-users. The selective encryption should be independent of the multimedia contents being streamed, thus eliminating the need for changes in the encoders and decoders. It should also be power and processor-efficient so that it could simultaneously support sharing of information among a heterogeneous set of devices including mobile platforms. In addition, in several application scenarios, the privacy or secrecy of the information is short lived and time dependent. For example, the video stream of a soccer match to paid customers is only valuable only during the game. After the game its value is significantly diminished. Therefore in such scenarios full encryption techniques are an overkill.

In this paper we consider RTP multimedia bit streams and present a highly scalable selective encryption solution that provides robust security, and is efficient in terms of processing/battery

* Corresponding author.
E-mail address: naf@parisdescartes.fr (F. Naït-Abdesselam).

https://doi.org/10.1016/j.jisa.2018.04.004
2214-2126/

power. The proposed selective encryption scheme identifies crucial fields in the RTP headers and, using a user key, encrypts only these fields. More importantly, it does not require media decoding and encrypts less than 3% of the bit stream assuming an average size of 800 bytes per RTP packets, while providing robust security equivalent to that of a fully encrypted bit stream.

We define security to be robust if both the encryption key and payloads are at least 128 bits long. The proposed scheme has been implemented on Nokia N-series mobile PDAs running Maemo, desktop computers, netbooks and laptops. We have used RTP packetized H.264 coded video bit stream in our experiments. The choice of these mobile platforms was mainly due to open-source modules and the flexibility offered by these platforms. The techniques developed in this paper can be adopted for any OS or device. Our results show on the average 15 time faster encryption speed, and 8 to 12 times more coded information for a given battery life, when compared to full encryption based solutions. A preliminary version of these results have appeared in [2].

2. Related work

Existing selective encryption approaches have been effectively applied to different multimedia codecs such as MPEG-1, MPEG-2, MP3, MPEG-4, and H.264 [3,4]. Meyer and Gadegast [5] was among the pioneer works that was based on partial encryption of MPEG-1 bit streams. The principle data to be secured included: all the headers, I-frames, and I-blocks. They proposed a number of combinations of the above scheme to attain different levels of security. Spanos and Maples [6] proposed to encrypt the I-frames and the ISO start and end codes of the MPEG stream. Tang [7] proposed an approach to use random permutation list instead of the zigzag order for mapping an 8x8 DCT block to a 1x64 block. Without the actual permutation list, it would be difficult to perform the inverse DCT transform on this data. This approach yielded non-optimal compression. Qiao and Nahrstedt [8] proposed to encrypt every other bit in the stream. Thus anyone without the knowledge of the actual decryption key would not be able to use this data. This scheme does reduce the overhead compared to encrypting the whole stream by 50%, but is still high compared to other selective encryption approaches. Shi and Bhargava [9] proposed to encrypt only the sign bits of transform coefficients. They first proposed it for DCT coefficients, and then extended it to motion vectors. An interesting approach was proposed by Allatar et al. [10], to encrypt every nth I-macroblock. This provided robust security, but only feasible if n was small, in the range of 2–3. Zeng and Lei [11] presented an algorithm to do selective bit scrambling, block shuffling, and block rotation of the transformed data. This scheme did reduce the processing overheads, but had the drawback of extra space requirements to hold this information. The proposal from Cheng and Li [12] also suffered from the same drawback in terms of extra space requirements for storage of decoding information. Wu and Kuo [13] proposed the use of multiple Huffman coding tables. This approach resulted in inefficient compression. Jun and Festijo [14] proposed selective encryption scheme for RTP packet but it uses a novel per packet key method using Diffie–Hellman key exchange procedure that is repeated for the selected RTP packets.

Since multimedia contents are jitter-sensitive, all our efforts have been made to reduce the overheads involved in the encryption process. The common observations made by most of these schemes are as follows:

(1) Encryption and decryption of an entire video stream takes a considerable amount of time. Therefore, only a critical portion of video can be encrypted to limit the cost of the operation.

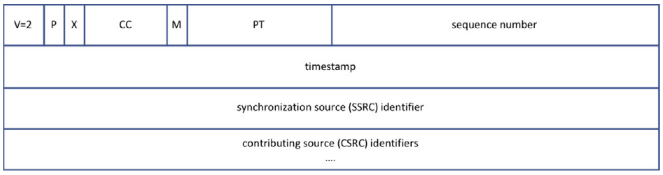

Fig. 1. RTP field header (RFC 3550).

(2) The protection level for the content must be identified. Optimal levels of security have to be enabled to the copyrighted entertainment materials to ensure both profit and demand.
(3) The protected media may have a limited lifetime. The encryption overhead should be optimal so that the protected media is not rendered useless due to not falling in the demand time zone. For example, security would make no sense if due to decryption overheads a protected live soccer broadcast can be viewed only after the match is over or vice versa. At the same time, to be secure it must not be broken until excerpts from the match are broadcast publicly in the succeeding sports programs. Therefore a protection scheme should strike a balance between restrictions and accessibility.

3. Proposed selective encryption approach

To avoid the modification of encoder/decoder modules as required by most of the existing selective encryption approaches, we investigate the role of RTP header information and identify critical fields without which proper play out of the media will not be possible. For the sake of completeness we first provide a brief review of the RTP header fields. Readers familiar with RTP headers may skip to Section 3.2.

3.1. RTP header format (RTP, RFC 3550)

RTP provides end-to-end network transport functionality for applications that need to transmit real-time data, especially multimedia content. RTP does not support resource reservation or guarantee Quality of Service. The data is monitored through the presence of a Control Protocol (RTCP), which provides basic identification and other controlling features [15]. The format of an RTP packet header is depicted in Fig. 1.

The fields in the RTP header are as follows:

- Version - V - 2 bits: Identifies the version of RTP.
- Padding P 1 bit: If this bit is set, it indicates the presence of one or more padding octets at the end of the stream. The last octet contains the number of octets used to do the padding.
- Extension X 1 bit: If set, indicates the presence of extension header after the fixed header block.
- CCRC Count CC 4 bits: Contains the number of CSRC identifiers that follow the header.
- Marker M 1 bit: Interpretation of the marker bit is defined by profile.
- Payload Type PT 7 bits: This field identifies the format of the RTP payload and determines its interpretation by the application.
- Sequence Number -16 bits: This field is incremented after each RTP packet transmission and is used by the receiver to detect packet loss and to reorder packets.
- Timestamp 32 bits: It indicates the sampling instant of the first octet in the RTP data packet.
- Synchronization Source - SSRC 32 bits: This field identifies the synchronization stream. It is randomly chosen and is unique across all the packets for a given stream.