ASSESSING CONTEMPORARY ISSUES IN SECURE VIDEO COMMUNICATION SYSTEMS

RESUMO

A importância da transferência de informações de vídeo entre diferentes plataformas e redes, utilizando um número crescente de codificadores e padrões de vídeo, tem sido evidenciada pelos recentes avanços tecnológicos na área de comunicações. Além de inúmeros desafios em termos de desempenho, os aspectos de segurança associados à transmissão de vídeo passaram recentemente a receber maior atenção dos pesquisadores da área. Dentre os aspectos de segurança importantes para as transmissões de vídeo destacam-se: criptografia, marcas d'água e autenticação. Este artigo apresenta uma visão geral dos aspectos de segurança mais relevantes para o desenvolvimento de sistemas de comunicações por vídeo e utiliza uma ferramenta de simulação recém-desenvolvida para ilustrar como estes sistemas podem ser simulados e estudados em laboratório.

ABSTRACT

Recent advances in the field of video communications have emphasized the importance of delivering video contents to a wide variety of devices, through different networks, and using a growing number of codecs and standards. In addition to the performance challenges behind video communications, the security issues have received increasing attention from the research community. Among the most relevant security aspects of contemporary video communication systems are: encryption, watermarking and authentication. This paper presents an overview of the most relevant security aspects of contemporary video communication systems and uses a newly developed simulation tool to illustrate how they can be simulated and investigated in a lab setting.

KEY WORDS
Secure video communications, network security.

1 Introduction

The popularization of multimedia communications has brought about a number of new concerns to IT managers and system administrators, ranging from network performance aspects (multimedia files are usually large and may clog otherwise unsaturated network links), copyright and intellectual property aspects (much of the multimedia files swapped among users contains unauthorized copy of copyrighted material), and security concerns (such as the growing number of multimedia-specific exploits reported during the past few years). As a result, concerns regarding security, scalability and manageability of existing systems become more acute as current solutions may not satisfy the demands of multimedia communications [35].

A new field of research known as *multimedia security* has emerged over the past few years. Its scope includes content protection techniques – primarily via encryption – and copyright protection aspects – usually through watermarking. Additional topics of interest include data hiding (of which steganography is a popular example), digital media fingerprinting and authentication [18]. Most of these techniques can be applied to images, audio, and video clips. Since video is the richest media format of all, this paper presents an overview of the most relevant security aspects of contemporary video communication systems. In addition to discussing some of the main conceptual aspects, technical challenges and relevant related work in these fields, it uses a newly developed simulation tool (SimViKi) to illustrate how secure video communication systems can be simulated and investigated in a lab setting.

This paper is structured as follows: Section 2 presents relevant recent developments in the field of multimedia security, with emphasis on video encryption, watermarking, and authentication. Section 3 provides an overview of recently reported multimedia-specific security exploits and discusses mechanisms to prevent and/or detect them. Section 4 looks at the combined aspects of security, scalability, encryption, and transcoding, and how they can interact in a comprehensive video communication system. Section 5 provides an example of a relevant contemporary secure video communications scenario simulated with a tool developed by the authors, SimViKi. Finally, Section 6 presents some concluding remarks and directions for future work.

2 Multimedia security techniques

This section presents relevant techniques used towards achieving multimedia security. Many of these techniques are extensions or modifications of classical security schemes. Others are newly-developed multimedia-specific techniques that do not have a traditional counterpart.

2.1 Video encryption

Encryption plays a key role ensuring confidentiality in most implementations of security for video communications. However, general purpose encryption algorithms (e.g., AES) are typically not optimal for video encryption because these algorithms do not conform to requirements of video application. In order to overcome this problem a significant number of encryption algorithms specifically designed for digital videos have been developed [33]. These algorithms take into account video-specific aspects such as the level of security and perception, format compliance, bitstream expansion, and error tolerance [33].

To identify an optimal level of security we have to carefully compare the cost of the multimedia information to be protected against the cost of implementing the protection. Lightweight encryption, known as *degradation*, may be sufficient for distributing "low value"multimedia content. Often, degradation intentionally preserves some perceptional information with visual quality that is unacceptable for most viewing purposes, which is referred to as *perceptual encryption*. On the other hand, if the video contains sensitive information the cryptographic strength must be substantial so that no perceptual information should be preserved [33].

It is often useful that the encryption algorithm preserves the video's compression format, known as *format compliance*. In other words, after encrypting the encoded video, ordinary decoders should still be able to decode it. The produced output will appear either perceivable but distorted or non-perceivable and random, depending on the type of encryption (Figure 1).

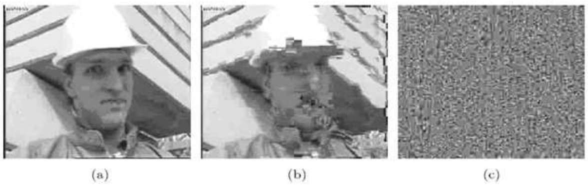


Figura 1: Decoded video produced by an ordinary decoder for: (a) video without encryption; (b) video encrypted with format-compliant perceptual encryption; and (c) video encrypted with high-security format-compliant encryption. (from [33]).

In many applications, it is required that the encryption transformation preserves the size of a bitstream. This is known as the *constant bitrate* requirement. However, more often than not, it is simply preferred that the output produced by an encryption-equipped encoder and the output produced by an ordinary encoder have similar sizes (a near-constant bitrate). A near-constant bitrate is likely to occur when a block cipher is used for encryption, since in that case the encrypted output is always a constant multiple of the blocksize [33].

In [32] a *correlation-preserving* video encryption algorithm is proposed in which encryption and decryption are performed outside the current video system (Figure 2). This results in no modification to the codec and in fully application-compliant video output. Moreover, the proposed correlation-preserving framework leads to promising results in the compression of video sequences with relatively little motion (e.g., videoconferencing) and can also be applied to a novel permutation-based steganography scheme [32].

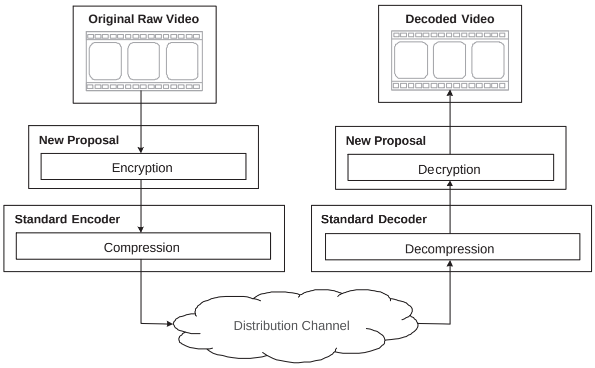


Figura 2: Architecture of the video encryption method proposed by Socek et al. (from [32]).

Finally, *error-tolerance*, or error-resilience, is of high importance for many multimedia systems. Advanced video coding systems (e.g. H.264) have their own error correcting mechanisms. Hence, a video encryption algorithm that preserves these mechanisms is favorable for video systems with noisy channels. For the most part, modern cryptography is designed for a generic bitstream, and as such, it disregards the aforementioned properties of a digital video and the requirements of a typical digital video application [33].

2.2 Video authentication

Video authentication is a process used to ascertain the trustworthiness of digital video. A video authentication system uses a combination of techniques such as digital signature, digital watermarking, error correction coding, and cryptographic hash functions to ensure the integrity of digital video, and verify that it has not been tampered with [9].

A typical video authentication system is shown in Figures 3 and 4. In the authentication process (Figure 3), for a given video, the authentication algorithm processes the features extracted from the video and outputs the authentication data which is encrypted using the encryption key to form the signature. The video integrity is verified by computing the new authentication data using the same authentication algorithm and features. The new authentication data is compared with the original authentication data as shown in Figure 4. If both match, the video is treated as authentic otherwise it is construed to be tampered [9].

Several video authentication solutions have been proposed during the past few years. While most focus on

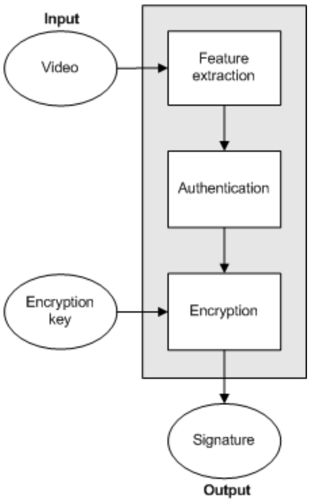


Figura 3: A typical video authentication system: authentication process (redrawn from [9]).

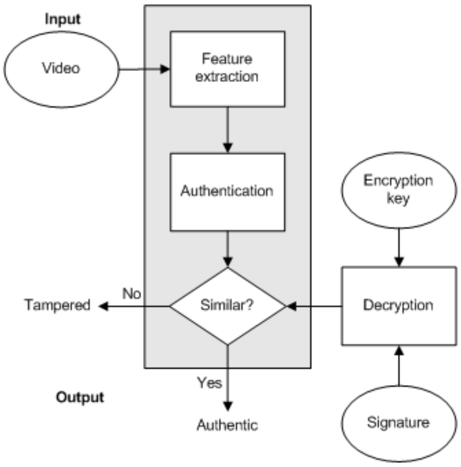


Figura 4: A typical video authentication system: verification process (redrawn from [9]).

frame-based video authentication (e.g., [9]), other approaches implement MPEG-4 compliant object-based authentication (e.g., [19]).

Robust, reliable and computationally feasible content-based video authentication techniques are currently active and challenging fields of research. Robustness to (e.g., JPEG and MPEG) compression (as well as other forms of image processing) requires the introduction of a tolerance bound at the integrity evaluation decision. In consequence, malicious content manipulations within the tolerance bound will not be detected [28].

2.3 Video and image watermarking

Watermarking is the process of embedding data into a multimedia element such as image, audio and video. For multimedia the watermark should be invisible to the human observer. A watermarking algorithm consists of the watermark structure, an embedding algorithm and an extraction or detection algorithm. Watermarks can be embedded into multimedia directly or after the multimedia element has been transformed by a mathematical transform such as DCT, DWT or DFT. Performance issues include robustness against attempts to remove the watermark, capacity (how watermark data can be hidden) and how transparent the watermark is under normal conditions [15].

Application Class	**Watermark Purpose**	**Application Scenarios**
Protection of Intellectual Property Rights	Conveys information about content ownership and intellectual property rights	Copyright Protection, Copy Protection, and Fingerprinting
Content Verification	Ensures that the original multimedia content has not been altered, and/or helps determine the type and location of alteration	Authentication and Integrity Checking
Side-Channel Information	Represents side-channel used to carry additional information	Broadcast Monitoring and System Enhancement

Tabela 1: Classification of watermarking application scenarios (from [27])

There is an important difference between encryption and watermarking in enforcing protection against unauthorized use. With an encryption-based technology it is possible to protect audiovisual content from eavesdroppers because only the authorized user will have access to the decryption keys. Watermarks do not preclude access to the watermarked content, but are used to protect the content owner against unlawful actions by malicious users such as tampering with the original contents (in which case a *fragile* watermark is often used) or misrepresenting the contents as their own (which calls for a *robust* watermark). Table 1 provides a classification of watermarking scenarios based on the type of information conveyed by the watermark [27]. A contemporary review of the state of the art in multimedia watermarking techniques is provided in [26].

3 Multimedia-specific requirements for network security

In this section we look briefly at the impact that multimedia communications have had on network security monitoring devices, particularly how intrusion detection systems (IDS) and firewalls can cope with the growing – although relatively modest – number of multimedia-specific security exploits.

3.1 Multimedia-specific security exploits

The number of security exploits targeted at multimedia files and applications has been growing steadily over the past few years. Recent examples of multimedia-related vulnerabilities include the multiple heap, stack, integer, and buffer overflow vulnerabilities found in Apple QuickTime [4, 3, 1, 2], the GIF-related heap overflow problem in Firefox [25], and the JPEG and PNG exploits reportedly found in Microsoft products [23, 24], and described below.

The JPEG exploit allows an attacker to gain control of an exploited system. The JPEG specification allows the embedding of comments in the JPEG file. The comment sections start with a hex value of `0xFFFE` to signal the start of the comment, followed by a two-byte value, which specifies the length of the comment, plus two bytes (for the field itself). The two-byte field theoretically allows 65,533 bytes of comment data (invisible when the JPEG is viewed). If the comment field is empty, the length value must contain the minimum length, or a value of 2. (2 bytes in length). However, if a specially crafted JPEG file sets this length to a 0 or 1 (illegal values), it causes a buffer overflow condition [7].

The PNG exploit uses especially crafted PNG chunks to create a buffer overflow condition that allowed an attacker to gain control of a target system. Similarly to the JPEG exploit, the end user would likely not realize their system was compromised, because the image would still display correctly even though there was a malformed section within the file header. This vulnerability is particularly dangerous when combined with MSN Messenger, because the user does not have to request or accept an image to have their system compromised. The attacker can use a PNG file as a buddy icon, which is automatically transferred during a chat session, thereby allowing them to gain control of the target system. The MSN Messenger PNG vulnerability is detailed in the advisory posted at CoreLabs [13]. This exploit is composed by placing specific values in the *IHDR* and *tRNS* chunks of the image. The *colors used* and *palette used* flags must be set in the *color type* field and the *alpha channel used* flag must not be set. The *color type* field must have a value of `0x03` and the contents of the tRNS chunk must exceed 256 and reach a function pointer address.

3.2 The role of firewalls and IDS

Firewalls usually cope with multimedia-related security risks by employing packet filters, which typically work by looking at predefined fields within the packet (header) and employing pattern matching techniques. More robust solutions may employ a combination of *stateful filters* – essentially, improved versions of packet filters which are able to extract information from the application layer and change their behavior according to ongoing traffic – and *proxy servers* – which allow the session flow to be retained, inspected and forwarded at the application layer [29, 30].

Intrusion Detection Systems (IDS) have become valuable tools for ensuring system and network security. IDS scan ongoing traffic in search of patterns and signatures that might indicate malicious or unauthorized activity [20, 10]. One of the issues currently facing network-based IDS is the high computational cost of doing real-time analysis when a large amount of traffic is passing through a connection. In such cases IDS usually have no choice but to skip packets [10]. The increase in multimedia traffic over communication networks, whether in the form of downloading or streaming large audio and video files, or due to the convergence of voice, data and video over IP, seems to compound the problem even further.

Currently, IDS are capable of blocking multimedia content based on port number (in the case of streaming audio/video), string matching of content type (e.g., content: "User-Agent $|3A|$ Quicktime") and file extension. None of these techniques verify the validity of the content; they simply assume that if data appears to be (from external identifiers, such as MIME) multimedia, then it is. They are usually not capable of detecting vulnerable files because they lack knowledge of the relevant transfer protocols or file formats.

Marques and Baillargeon [11, 22] have demonstrated that adding specialized multimedia knowledge to an IDS packet analysis capabilities may help detect multimedia-specific exploits (and – as a bonus – achieve substantial computational savings) in both streaming and non-streaming scenarios. They have presented a method to improve the performance of IDS based on multimedia traffic classification: by embedding multimedia-specific knowledge into the IDS, trusted multimedia contents can be identified and allowed to bypass the detection engine, thereby allowing the IDS to focus on other traffic. Moreover, IDS become capable of detecting multimedia-specific exploits which would otherwise go by unnoticed.

4 Secure transcoding and secure scalable streaming (SSS)

The security aspects of multimedia applications described in Section 2 are primarily oriented towards protecting against unauthorized use of (copyright-protected and/or sensitive) multimedia contents. In this Section we focus on security aspects that are network-oriented, such as secure transcoding, secure scalable streaming (SSS), and related efforts.

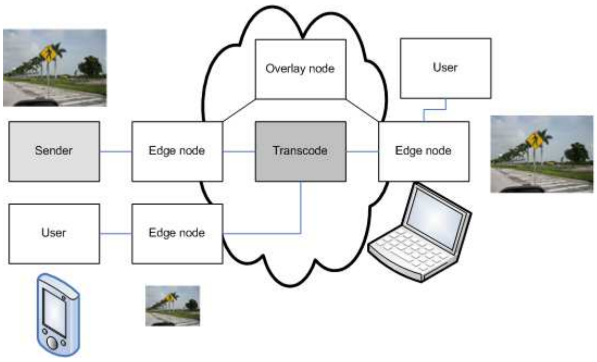


Figura 5: A typical video transcoding scenario (adapted from [39]).

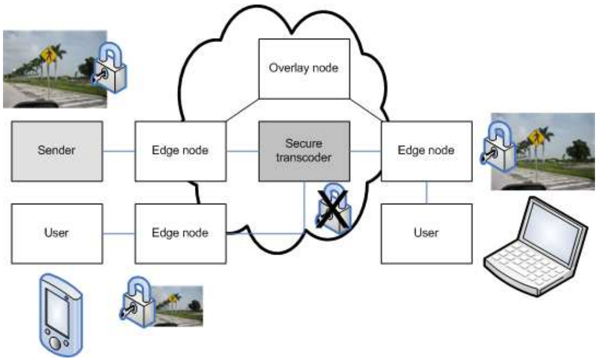


Figura 6: A secure video transcoding scenario (adapted from [39]).

The combined use of scalable video coding techniques, video streaming and video transcoding has allowed transmission of video contents through computer networks to a wide variety of clients, with different screen size, mobility aspects, and quality requirements. Extending these techniques to a secure scenario presents a number of challenges that are the topic of active research in secure transcoding, SSS, and related efforts [37, 8]. Typical solutions involve exploring selected portions of the encoded video stream in such a way that allows the transcoders' basic functions to be performed without a need for decryption and subsequent encryption, i.e., without allowing for the possibility of the transcoders themselves becoming a security vulnerability.

A typical scenario for which these techniques are useful is depicted in Figure 4. In this case, a high-resolution video is uploaded to the network. A transcoder is in charge of generating a lower-quality version of the same material for users with smaller screens, for instance. If the video is sent in the open, conventional scalable video coding techniques should be enough to allow proper transcoding and there are no security issues to take into account. If, however, the scenario is slightly modified to a case where the video is encrypted prior to uploading and the transcoder becomes a secure transcoder (Figure 5), the question becomes: how do you transcode encrypted streams without introducing a security risk by giving the transcoder a copy of the decryption key? The answer to this question has been the goal of recent work by Wee, Apostolopoulos and colleagues [38, 37, 36, 8, 40, 14, 39].

Wee, Apostolopoulos and colleagues have developed techniques for secure scalable streaming (SSS) (a combination of scalable coding and progressive encryption) for the case of wireless networks [38], modified it to fit the JPSEC framework [37, 36, 40, 14], and later extended it for non-scalable video [8]. Their work illustrates well the trade-offs between encryption and (scalable) coding, the additional complexity introduced by transcoders (and the need of delivering different quality variants of the same video program) and the increasing role of standards (e.g., JPSEC) in providing an elegant framework for its implementation.

5 Case study

This section presents a secure video communications scenario (Figure 7) that involves the aspects discussed in Section 2, namely: encryption, transcoding, authentication via watermarking, and general security measures, such as firewalls, intrusion and extrusion detection.

This scenario incorporates a number of realistic components. The desired multimedia content resides outside of the company's network and is downloaded to a local server only once. The transcoder-enabled cache (TeC) functions based on the number of clients and the importance of the content. Upon receiving a single copy of the watermarked, encrypted, optimal-quality video it then verifies the authenticity (watermark) of the video program and, upon request, forwards the content to various clients throughout the internal network.

The client base represents a wide variety of platforms, privileges, and needs. Clients may (and usually do) require transcoding, may decrypt content (if authorized to do so), and may be mobile. Transcoders are located in the internal network and are able to transcode video without decryption and subsequent encryption. The scenario also accounts for the possibility that transcoders may be compromised by (internal) intruders injecting content into the video stream.

In this scenario outbound multimedia traffic is inspected for permissions embedded within its watermarks. Ideally, proprietary video is watermarked to denote copyright and importance. Video tutorials, meetings, and product announcements can contain information that would otherwise disclose too much of the internal workings, operations, and developments within a company. We propose a network component to implement extrusion detection [12], the "waterwall"which complements the combined intrusion detection actions performed by the firewall and an IDS. The waterwall should be a device on the perimeter of a network, dropping any multimedia that contains a specific watermark, and generating a corresponding alert.

In summary, the proposed model allows the simulation of an enormous number of variables (from the impact of

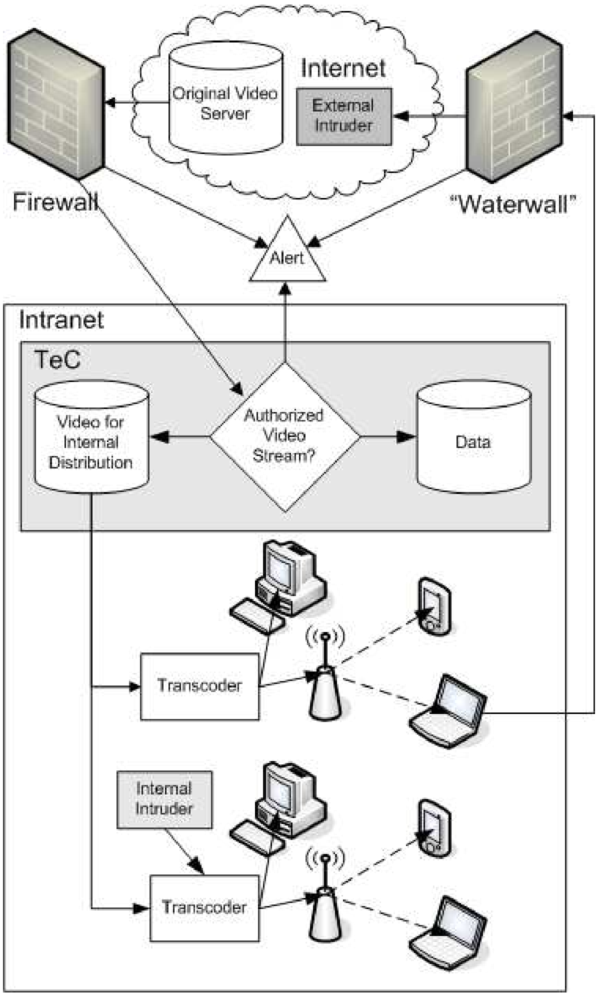


Figura 7: A secure video communications scenario combining encryption, watermarking, transcoding, intrusion and extrusion detection.

choosing one encryption algorithm over another to the specification of network-related impairments that may affect the received video stream) that are relevant in today's secure video communication systems and the evaluation of their role in the overall performance of the system.

5.1 SimViki: a versatile tool for simulating video communication scenarios

We have developed a new framework for simulation of video communication systems that builds upon the OMNeT++ [34] platform and the video traces research developed at Arizona State University (ASU) [5], integrates it with MATLAB [6], allowing the use of actual video files (in addition to pre-existing video traces [31]), different encoding and security-related (e.g., encryption) algorithms, therefore improving and extending the functionality of OMNeT++ and making it a one-of-a-kind tool for video communications teaching and research purposes.

5.2 Design and implementation aspects

OMNeT++ provides a component-based architecture for models. Components (modules) are programmed in C++, and then assembled into larger components and models using a high-level language. OMNeT++ has extensive GUI support. Due to its modular architecture, the simulation kernel (and models) can be embedded easily into other applications [34]. In our tool, OMNeT++ provides the network simulation core functionality.

Video traces are an alternative to bit streams in which only the number of bits used for the encoding of the individual video frames is provided. As a result, there are no copyright issues, but the ability to infer information about the visual quality of the encoded/received video sequence is lost [31]. Video traces have become very popular among video communication and networking researchers, with several video traces from public video trace libraries [31, 5, 16] currently available. Today, the issues of quality and their correlation to the video frame sizes are of great importance, driven by the need for differential quality of service (QoS), requiring new traces to be made available as well as the existing interfaces to network simulation tools (such as OMNeT++) to be updated and adapted [17] .

The developed tool overcomes some of the limitations of conventional video traces. Instead of enhancing the contents of the video trace (as in [21]), we expand the scope of the simulation environment to allow use of actual video sequences, easy inclusion and configuration of codecs and security-related processing blocks (e.g., encryption/decryption algorithms), and visualization of results – using MATLAB's image processing and viewing capabilities – that demonstrate what the video sequence would actually look like at the receiving end or in any meaningful point within the network. Auxiliary modules were also developed to implement video file creation and conversion functionality, MPEG-1 and M-JPEG parsing of previously encoded video sequences, Perl and MATLAB scripts for controlling the main modules, and a GUI-based wizard for easy configuration of the tool's main options.

5.3 Simulation results

In this section we present a selection of snapshots and plots obtained while running a simulation of the proposed model.

Figures 8, 9 and 10 are three snapshots of the same OMNeT++ block diagram, in which two probes – before and after one of the transcoders – are emphasized, suggesting test points where visual as well as statistical data can be collected for instant display and/or future analysis. Together they show representative results over time as the tool simulates the desired video stream – as well as other possible traffic – being generated and making its way through the company's network.

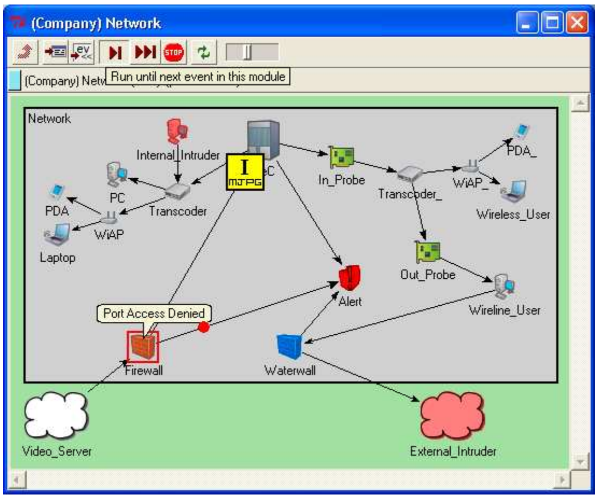


Figura 8: Snapshot indicating firewall activity rejecting unauthorized access to the company's network.

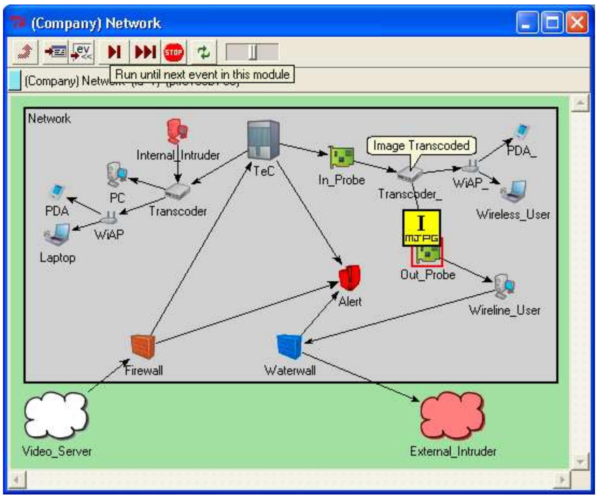


Figura 9: Snapshot showing transcoded frame on its way to the intended client.

Figure 8 shows the firewall in action, rejecting an external attempt to access internal data (while a previously received I-frame from an M-JPEG video sequence makes its way to the TeC).

Figure 9 shows the moment in which a successfully transcoded I-frame reaches the probe placed after the transcoder, at which point it can be seen in an auxiliary MATLAB image display window. Depending on the nature of the video stream, the inspected frame may be encrypted, watermarked or both.

Figure 10 shows an extrusion detection scenario. In this case an attempt to export protected (watermarked) video contents to an unauthorized external user is detected, prevented, and the corresponding alert generated.

Figure 11 shows a sample frame and the equivalent watermarked (with a barely noticeable star) and perceptually encrypted versions of it.

Our tool allows an intuitive combination of MATLAB and OMNeT++ screens into a single GUI, as shown in

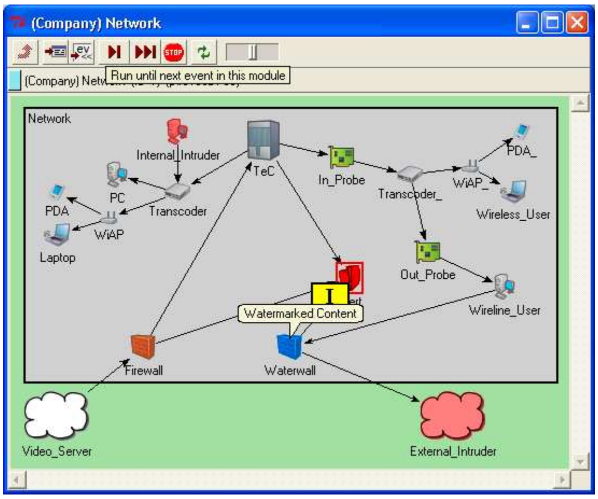
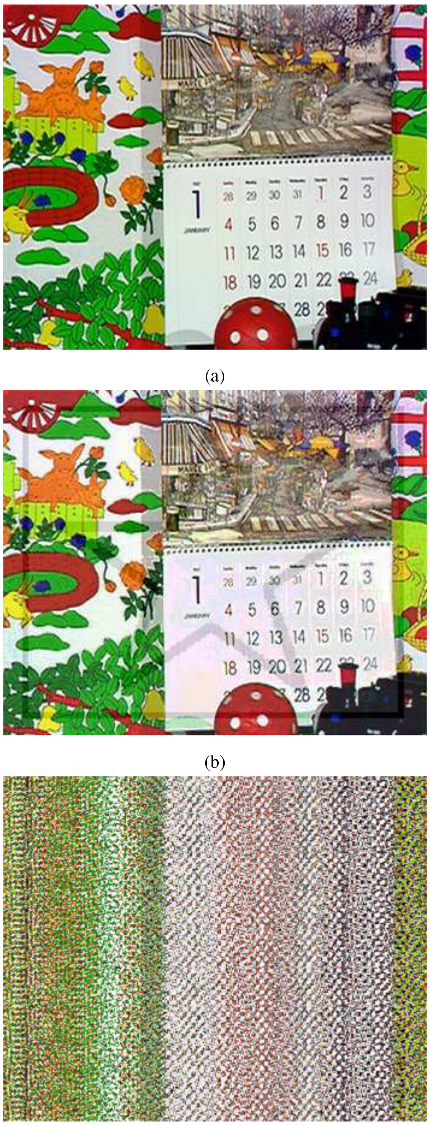


Figura 10: Snapshot showing an extrusion detection scenario.



(a)

(b)

(c)

Figura 11: Representative frame: (a)original, (b) watermarked, (c) (perceptually) encrypted.

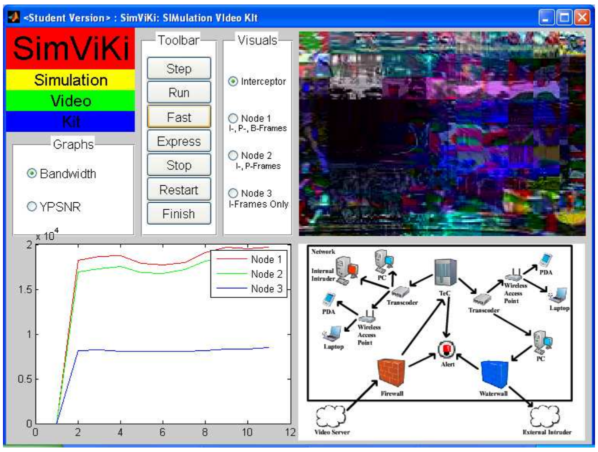


Figura 12: SimViKi Simulator showing an encrypted frame (upper-right) along with a simultaneous network state (lower-right) and bandwidth plot (lower-left).

Figure 12. Functionality extends beyong the OMNeT++ block diagram (and its implicit dynamic behavior). The user can see actual frames in real-time as they are processed. Relevant plots are displayed in the bottom-right portion of the screen. The interface allows the user to select different options using radio buttons, also in real-time.

6 Concluding remarks

This paper discussed contemporary issues in secure multimedia communications, particularly secure video communications. A model that includes contemporary security-related tasks (encryption, authentication, intrusion detection) has been presented, implemented, and simulated using a newly developed tool. Simulation results demonstrate that the proposed framework provides a versatile and capable way to experiment with video communication scenarios. Ongoing and future work includes expanding the tool's functionality to a broader number of external algorithms and stand developing further secure multimedia communications experiments for both teaching and research.

Referências

[1] Apple QuickTime FlashPix integer overflow. http://www.kb.cert.org/vuls/id/570689.

[2] Apple QuickTime JPEG integer overflow. http://www.kb.cert.org/vuls/id/289705.

[3] Apple QuickTime MPEG-4 movie buffer overflow. http://www.kb.cert.org/vuls/id/587937.

[4] Apple QuickTime Multiple Remote Buffer and Integer Overflow Vulnerabilities. http://www.frsirt.com/english/advisories/2006/1778.

[5] Arizona State University – Video Traces Research Group. http://trace.eas.asu.edu/.

[6] MATLAB. http://www.mathworks.com/products/matlab/.

[7] Security Watch Letter: Inside the JPEG Virus. http://www.pcmag.com/article2/0,1759,1661942,00.asp.

[8] J. G. Apostolopoulos. Secure media streaming & secure adaptation for non-scalable video. In *ICIP*, pages 1763–1766, 2004.

[9] P. K. Atrey, W.-Q. Yan, and M. S. Kankanhalli. A scalable signature scheme for video authentication. *Journal of Multimedia Tools and Applications*, 2006.

[10] R. Bace. An Introduction to Intrusion Detection and Assessment. Technical report, ICSA Labs, 2000.

[11] P. Baillargeon. A Method for Adding Multimedia Knowledge For Improving Intrusion Detection Systems. Master's thesis, Florida Atlantic University, August 2005.

[12] R. Bejtlich. *Extrusion Detection: Security Monitoring for Internal Intrusions*. Addison-Wesley, November 2006.

[13] Core Security Technologies Advisory. MSN Messenger PNG Image Parsing Vulnerability. http://www.coresecurity.com/, 2005.

[14] F. Dufaux, S. Wee, J. Apostolopoulos, and T. Ebrahimi. JPSEC for secure imaging in jpeg 2000. In *SPIE Proc. Applications of Digital Image Processing XXXVII*, August 2004.

[15] A. M. Eskicioglu and E. J. Delp. An overview of multimedia content protection in consumer electronics devices. *Signal Processing: Image Communication 16*, 2001.

[16] F. H. Fitzek and M. Resslein. MPEG-4 and H.263 video traces for network performance evaluation. http://www.tkn.tu-berlin.de/publications/papers/TKN0006.pdf.

[17] F. H. P. Fitzek, P. Seeling, and M. Reisslein. Using network simulators with video traces, May 08 2003.

[18] B. Furht, D. Socek, and A. Eskicioglu. *Multimedia Security Handbook*, chapter Fundamentals of Multimedia Encryption Techniques. CRC Press, 2005.

[19] D. He, Q. Sun, and Q. Tian. A secure and robust object-based video authentication system. *EURASIP Journal on Applied Signal Processing*, 2004.

[20] R. Kemmerer and G. Vigna. Intrusion Detection: A Brief History and Overview. *IEEE Computer*, 35(4):27–30, April 2002.

[21] O. A. Lotfallah, M. Reisslein, and S. Panchanathan. A framework for advanced video traces: Evaluating visual quality for video transmission over lossy networks. *EURASIP Journal on Applied Signal Processing*, 2006.

[22] O. Marques and P. Baillargeon. A Multimedia Traffic Classification Scheme for Intrusion Detection Systems. In *Proc. of the IEEE ICITA'05*, Sydney, Australia, July 2005.

[23] Microsoft Security Bulletin MS04-028. Buffer Overrun in JPEG Processing (GDI+) Could Allow Code Execution (833987). `http://www.microsoft.com/technet/security/bulletin/MS04-028.mspx`, 2004.

[24] Microsoft Security Bulletin MS05-009. Vulnerability in PNG Processing Could Allow Remote Code Execution (890261). `http://www.microsoft.com/technet/security/bulletin/MS05-009.mspx`, 2005.

[25] Mozilla Foundation Security Advisory 2005-30. GIF heap overflow parsing Netscape Extension 2. `http://www.mozilla.org/security/announce/mfsa2005-30.html`, 20045.

[26] E. Muharemagic and B. Furht. *Multimedia Security Handbook*, chapter A Survey of Multimedia Watermarking Techniques. CRC Press, 2005.

[27] A. Nikolaidis, S. Tsekeridou, A. Tefas, and V. Solachidis. A survey on watermarking application scenarios and related attacks. In *Proceedings of the 2001 International Conference on Image Processing*, pages 991–994, April 2001.

[28] M. Queluz. Authentication of digital images and video: Generic models and a new contribution. In *Signal Processing: Image Communication 16*, pages 461–475, 2001.

[29] U. Roedig, R. Ackermann, C. Rensing, and R. Steinmetz. A distributed firewall for multimedia applications. `http://www.cs.ucc.ie/misl/publications/files/ddfwroedig.pdf`.

[30] U. Roedig and J. B. Schmitt. Multimedia and firewalls: a performance perspective. *Multimedia Syst.*, 11(1):19–33, 2005.

[31] P. Seeling, M. Reisslein, and B. Kulapala. Network Performance Evaluation Using Frame Size and Quality Traces of Single-Layer and Two-Layer Video: A Tutorial. Technical report, Arizona State University, 2004.

[32] D. Socek, H. Kalva, S. S. Magliveras, O. Marques, D. Culibrk, and B. Furht. New approaches to encryption and steganography for digital videos.

[33] D. Socek, H. Kalva, S. S. Magliveras, O. Marques, D. Culibrk, and B. Furht. A permutation-based correlation-preserving encryption method for digital videos. *ICIAR 2006 - International Conference on Image Analysis and Recognition*, September 2006.

[34] A. Varga. The OMNeT++ discrete event simulation system. *Proceedings of the European Simulation Multiconference (ESM'2001)*, June 2001.

[35] S. Voloshynovskiy, O. Koval, F. Deguillaume, and T. Pun. Multimedia security: Open problems and solutions. *In Proceedings of NATO - Advanced Study Institute: Security Through Science Program, Nork, Yerevan, Armenia*, October 2005.

[36] S. Wee and J. Apostolopoulos. Secure scalable streaming and secure transcoding with JPEG, 2000.

[37] S. Wee and J. Apostolopoulos. Secure scalable streaming enabling transcoding without decryption. *IEEE ICIP*, October 2001.

[38] S. Wee and J. Apostolopoulos. Secure scalable video streaming for wireless networks, 2001.

[39] S. J. Wee. Mobile Streaming Media Overlays Secure Media Delivery Technologies. `http://www.onr.navy.mil/about/conferences/rd_partner/2005/docs/past/2004/2004_wee_mobile_streaming_media.pdf`.

[40] S. J. Wee and J. G. Apostolopoulos. Secure transcoding with JPSEC confidentiality and authentication. In *ICIP*, pages 577–580, 2004.