Secure Network Coding for Multi-Resolution Wireless Video Streaming

Luísa Lima Steluta Gheorghiu João Barros Muriel Médard Alberto Lopez Toledo

***Abstract*— Emerging practical schemes indicate that algebraic mixing of different packets by means of random linear network coding can increase the throughput and robustness of streaming services over wireless networks. However, concerns with the security of wireless video, in particular when only some of the users are entitled to the highest quality, have uncovered the need for a network coding scheme capable of ensuring different levels of confidentiality under stringent complexity requirements. We show that the triple goal of hierarchical fidelity levels, robustness against wireless packet loss and efficient security can be achieved by exploiting the algebraic structure of network coding. The key idea is to limit the encryption operations to a critical set of network coding coefficients in combination with multi-resolution video coding. Our contributions include an information-theoretic security analysis of the proposed scheme, a basic system architecture for hierarchical wireless video with network coding and simulation results.**

***Index Terms*— Network coding, video streaming, wireless networks, multi-resolution coding, security**

I. Introduction

While there has been abundant research aiming at ensuring a reasonable quality of video experience for wireless users, the task of providing video streaming of variable quality to a heterogeneous set of receivers with different subscription levels is still an open issue. The key challenge is to serve wireless users with video streams that are both (i) of different quality, depending on subscription level, and (ii) with security guarantees to ensure that only authorized users will access the protected video streams.

In order to illustrate this problem let us consider the scenario in *Fig. 1*, in which nodes A, B and C are interested in a video stream served by node S, but they have paid for different video qualities, for example different layers of a multi-resolution video stream. Node S can connect to the receivers through 3 relay nodes in wireless range, but with poor channel quality. Due to the noisy nature of the wireless medium, reliable video transmission requires S to retransmit the lost packets using the feedback received from nodes A, B and C. Moreover, the relays need to synchronize and schedule transmissions to ensure that every receiver gets all the packets without duplicates. Under this scenario, video quality can decrease, because some video frames are not delivered in a timely fashion and are therefore skipped.

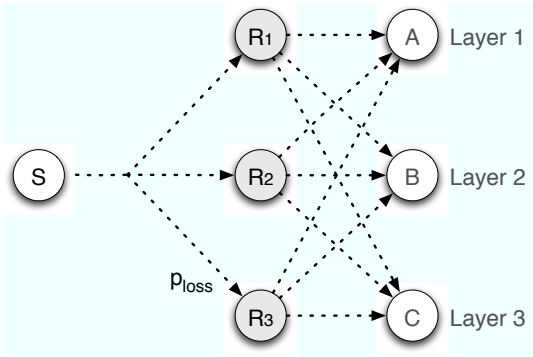

Fig. 1. A source S streams video to 3 sink nodes A, B and C through relay nodes R_1, R_2 and R_3 in a wireless setting. The probability of dropping a packet in each link (in dashed) is p_{loss}. The sinks subscribed for different video quality, thus one must devise mechanisms to ensure reliable delivery over the wireless medium, and protection against unauthorized access.

Moreover, given the broadcast property of the wireless medium, nodes that did not have subscription access to certain layers can potentially overhear the transmitted packets; e.g., in *Fig. 1*, node B could overhear layer 3 frames. Preventing unauthorized access to certain layers in the presence of relay nodes thus imposes a challenging security problem, in particular because encryption of the complete video stream is often deemed unfeasible in resource-limited mobile terminals. Real-time decoding of high-quality video already consumes a great deal of processing power, and can become overwhelming in conjunction with the resources required for the decryption of large files [1], [2]. Moreover, a lossy wireless medium imposes additional requirements to the security mechanisms, such as robustness to losses and limited synchronization to prevent scheduling problems.

A solution consists in reducing complexity by partially encrypting the video data [3], [4]. However, it is hard to evaluate the degree of security provided by these schemes [4]. The use of layered coding in wireless scenarios was seen as promising, but it is likely to yield prioritization and scheduling problems. For instance, [5] has shown that even the simple prioritization of the base layer is not a trivial task.

L. Lima (luisalima@dcc.fc.up.pt) is with the Instituto de Telecomunicações (IT) and the Department of Computer Science, Faculdade de Ciências da Universidade do Porto, Portugal. J. Barros (jbarros@fe.up.pt) is with the Instituto de Telecomunicações (IT) and the Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Portugal. M. Médard (medard@mit.edu) is with the Research Laboratory of Electronics at the Massachusetts Institute of Technology (MIT RLE). A. Lopez Toledo (alopezt@tid.es) and Steluta Gheorghiu (steluta@tid.es) are with Telefonica Research, Barcelona, Spain. Part of this work was done while the first author was a visiting student at the Research Laboratory of Electronics at the Massachusetts Institute of Technology. Part of this work was carried out with assistance of financial support from the European Community under grant FP7-INFSO-ICT-215252 (N-Crave Project). This work was partly supported by the Fundação para a Ciência e Tecnologia (Portuguese Foundation for Science and Technology) under grant SFRH/BD/24718/2005. A. Lopez Toledo is supported by the Institució Catalana de Recerca i Estudis Avançats (ICREA). Some of the results were also presented at the IEEE Information Theory Workshop in Volos, Greece, June 2009.

In order to tackle the above problems, we turn to a technique known as network coding. The key idea of network coding [6] is to allow nodes in a network to combine different information flows by means of algebraic operations. This principle leads to an unconventional way of increasing the throughput and robustness of highly volatile networks, such as wireless networks, sensor networks and peer-to-peer systems [7]. The benefits for wireless communications have been shown in [8], [9], [10] and [11]. Network coding can also minimize the decoding delay with feedback [12], making it suitable for multimedia streaming [13], [14], [15].

Protection of a wireless video stream, while increasing the overall robustness to losses and failures, reducing scheduling problems and adding resilience, is also possible using network coding. By viewing the network code as a cipher, it is possible to create a lightweight cryptographic scheme that reduces the overall computational complexity [16]. Thus, network coding inspires a reformulation of the typical separation between encryption and coding for error resilience. It is unnecessary to perform security operations twice, since we can take advantage of the inherent security of this paradigm [17], [18].

In this paper, we take advantage of the above benefits of network coding to develop and analyze a novel secure network coding architecture for wireless video. We consider a multicast setting in which several devices, which are in general heterogeneous and have limited processing capabilities, subscribe to multi-resolution streaming video in a lossy wireless network. We show how security operations performed at the network coding layer allow us to achieve our goals, which are (i) to reduce the number of encryption operations while meeting the prescribed security guarantees, (ii) to combine the resulting lightweight security scheme with efficient layered codes and streaming protocols for wireless video and (iii) to match network coding with scalable video streams, relying on network coding's asynchronous operation and inherent robustness to link failures and packet loss. Our main contributions are as follows:

- We propose a *secure scalable network coded method for video streaming* designed for delay-sensitive applications that exploits the robustness of network coding with manageable complexity and quantifiable security levels. We also show how hierarchical codes for scalable video based on successive refinement can be combined with network coding in scenarios where not all the nodes are authorized to receive the best quality;
- We carry out an *analytical evaluation* of the security properties of our scheme, and also address its performance and implementation in a wireless streaming service;
- We offer insights and *system considerations* regarding implementation in real scenarios;
- We provide a preliminary *proof-of-concept* for our network coded video architecture in several wireless scenarios via simulation.

The remainder of the paper is organized as follows. *Section II* describes the network setup and the attacker model, as well as the fundamental coding and encryption principles behind this work. *Section III* presents the proposed scheme and its security evaluation. Preliminary system aspects and implementation guidelines are presented in *Section IV*. The performance evaluation of the scheme is presented in *Section V*. Finally, *Section VI* concludes the paper.

II. Preliminaries

Let us consider the diagram in *Fig. 2*, where a source generates multilayer video that is encoded to be transmitted through a wireless network. We focus on how to create a secure scalable stream by matching the multilayer video with the network encoder.

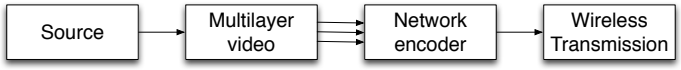

Fig. 2. Coding diagram considered. A source generates multilayer video. The video is fed to the network encoder and then undergoes the transmission in a wireless network.

A. Network Model and Abstractions

We consider an abstraction of a wireless network where the source and relay nodes only have access to the identifiers of the sinks (e.g. the IP addresses). Thus, there is no centralized knowledge of the network topology or of the encoding functions.

We adopt the model of video layers from [19], illustrated in *Fig. 3*. Video data is divided into groups of pictures (GoPs)[1] with a constant duration. The data is then encoded into L layers; each layer is divided into a fixed number of packets. Each layer is dependent on all previous layers, that is, layer 1 is necessary to decode layer 2, layer 2 is necessary to decode layer 3, etc.

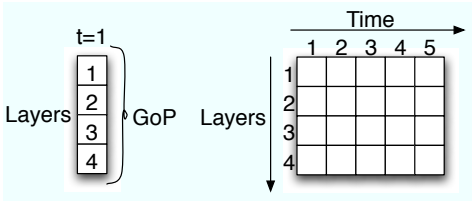

Fig. 3. Layer model. The video data is divided into groups of pictures (GoP) with the duration of 1 second. GoPs are then subdivided into layers.

B. Threat Model

We consider the threat posed by a passive attacker with the following characteristics:

1) he can observe every transmission in the network;
2) he has full access to information about the encoding and decoding schemes;
3) he is computationally bounded and thus unable to break hard cryptographic primitives.

The goal of the attacker is to recover the multicast video stream at the highest possible quality.

[1]We use the terms video segment and GoP interchangeably.

C. Network Coding and Security

Random Linear Network Coding (RLNC) is a completely distributed scheme to implement network coding protocols, whereby nodes draw several coefficients at random and use them to form linear combinations of incoming packets [20]. The resulting packet is sent along with the global encoding vector, which records the cumulative effect of the linear transformations suffered by the original packet while on its path from the source to the destination. The global encoding vector enables the receivers to decode by means of Gaussian elimination.

The idea that inspired the scheme presented in this paper is SPOC (Secure Practical Network Coding) [16]. SPOC is a lightweight security scheme for confidentiality in RLNC, which provides a simple yet powerful way to exploit the inherent security of RLNC in order to reduce the number of cryptographic operations required for confidential communication. This is achieved by protecting (or "locking") only the source coefficients required to decode the linearly encoded data, while allowing relay nodes to run their network coding operations on substitute "unlocked" coefficients which provably do not compromise the hidden data.

III. Secure Network Coding for Video Streaming

In this section we introduce our security scheme and elaborate on its main properties.

A. Scheme Operation

The operations at the *source* are illustrated in *Fig. 4*, which also introduces the notation used in the examples in this section. The scheme starts with a one-time key distribution between the source and the receivers. As keys can be reused, only one key per layer is needed for multi-resolution encryption (a single key for the single resolution video case), that would be shared among all the receivers. Then, for each GoP, the source generates an $n \times n$ lower-triangular matrix $\mathbf{A}$, in which n is the number of layers in the GoP. Matrix $\mathbf{A}$ is used for encoding at the source only. Each non-zero entry of $\mathbf{A}$ is an element a_{ij} chosen uniformly at random from all non-zero elements of the field $\mathbb{F}_q \backslash \{0\}$.

The GoP is then divided into vectors $\underline{b}^{(1)} \dots \underline{b}^{(w)}$, in which the first symbol of each vector belongs to layer 1, the next symbol belongs to layer 2, etc. The number of vectors created[2] is $\lceil$size of GoP / n$\rceil$. Then, at least one symbol of each vector $\underline{b}^{(i)}$ is encrypted for each use of the encoding matrix. As layers are dependent — layer i is needed to decode layer $i+1$ — the best approach is to encrypt the more informative base layer of the GoP in order to achieve maximum security (in this case, b_1 for each vector $\underline{b}^{(i)}$). This is standard practice in multimedia security [4]. We denote the output of the operation of a stream cypher to a symbol P with a random key $\underline{K}$ as $E(P, \underline{K})$. Finally, the payload of the packets is composed by applying the encoding matrix $\mathbf{A}$ successively to the information symbols to be sent, i.e., the payload is formed by concatenating all the vectors $\mathbf{A}(E(b_1, \underline{K}), b_2, \dots, b_x)^T$.

[2]For clarity, we ignore inconsistencies regarding the proportion between the number of symbols in the layers.

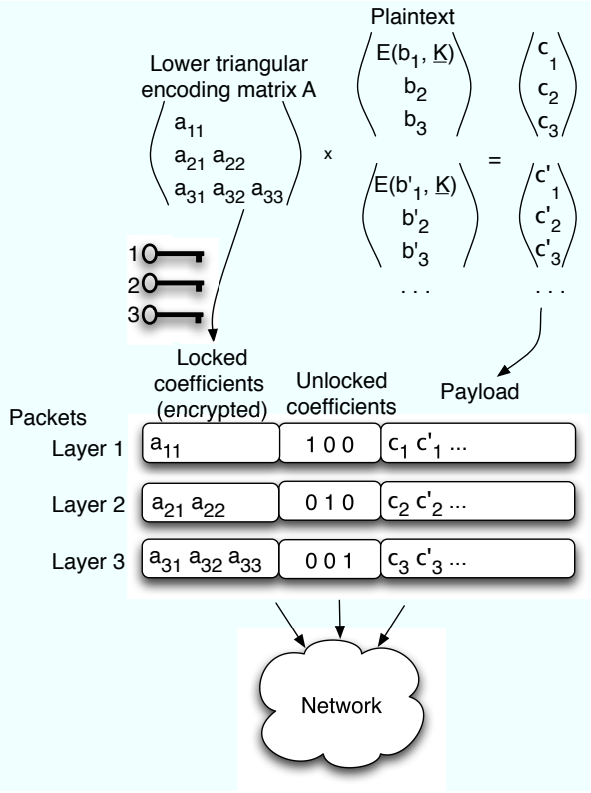

Fig. 4. Illustration of the operations performed at the source. First, a 3×3 lower triangular matrix in which each non-zero element is chosen uniformly at random out of all non-zero elements of a finite field is generated. The plaintext is divided into vectors of 3 elements and the first position of each vector is encrypted using a stream cypher. The matrix is multiplied by each of the vectors to generate the payload. The coefficients of matrix $\mathbf{A}$ are locked using one different key for each line of the matrix and placed in the header of the packets. One line of the identity matrix is generated for each line of the locked coefficients. The packets are then sent out to the network.

Next, the source encrypts each line of matrix $\mathbf{A}$ with the corresponding layer key. Matrix $\mathbf{A}$ is the *locked coefficients* matrix. The source then generates a $n \times n$ identity matrix $\mathbf{I}$, which corresponds to the *unlocked coefficients*. The packets are composed by the header, which includes the *locked* and *unlocked* coefficients, and the payload. Note that, because of the nested structure of coding, determined by the triangular matrix, a packet from layer 1 corresponds to the first line of matrix $\mathbf{A}$, a packet from layer 2 corresponds to the second line of matrix $\mathbf{A}$, etc, so that each packet of layer x includes packets from layers $1, \dots, x-1, x$. Note also that when performing a linear combination of one packet of layer x with a packet of layer $y > x$, the resulting packet belongs to layer y.

The *relays* encode packets according to the rules of standard RLNC protocols [20]. The algebraic coding is performed indistinguishably on unlocked coefficients, locked coefficients and payload. Relays identify the layer of a packet by looking at the first non-zero position in the unlocked coefficients, and packets are mixed with packets of the same or lower layers only.

The *receivers* apply Gaussian elimination following standard RLNC over the unlocked coefficients. The locked coefficients are recovered by decrypting each line of the matrix with the corresponding key. The plaintext is then obtained by

forward substitution.

Note that the protected symbols should be encrypted with the key for the lowest level in the network (that is, $\underline{K}_1$), so that all legitimate participants in the protocol can decrypt the locked symbols. If layer 1 is to be accessible by all nodes in the network, the first line of the matrix should be sent unencrypted and the encryption of symbols should start at symbol 2. We do not provide further details of this case for want of space.

Table I summarizes the scheme operation. In what follows, we elaborate on the matching of multiresolution video and security, prioritization and scheduling issues. Finally, we provide the security analysis.

TABLE I

SUMMARY OF PROPOSED SCHEME

Initialization (source nodes):
• A key management mechanism is used to exchange n shared keys with the sink nodes (one for each layer); • The source node generates a $n \times n$ lower triangular matrix $\mathbf{A}$ in which each of the non-zero entries is an element from the multiplicative group of the finite field, $a \in \mathbb{F}_q \backslash \{0\}$; • The coefficients corresponding to a distinct line of the $n \times n$ identity matrix are added to the header of each coded packet. These correspond to the *unlocked* coefficients. • Each line l of the matrix $\mathbf{A}$ is encrypted with shared key $\underline{K}_l$ and placed in the header of each packet. These coefficients correspond to the *locked* coefficients; • The source node applies the matrix $\mathbf{A}$ to the packets to be sent, and places them in its memory.
Initialization (relay nodes):
• Each node initializes n buffers, one for each layer in the network.
Operation at relay nodes:
• When a packet of layer l is received by a node, the node stores the packet in the corresponding buffer; • To transmit a packet of layer l on an outgoing link, the node produces a packet by forming a random linear combination of the packets in buffers $1, \ldots, l$, modifying both the unlocked and locked coefficients without distinction, according to the rules of standard RLNC based protocols.
Decoding (sink nodes):
When *sufficient packets are received:* • The sink nodes perform Gaussian elimination on the matrix of unlocked coefficients, applying the same operations to the remainder of the packet, thus obtaining the original locked coefficients and coded packets; • The receiver then decrypts the locked coefficients using the corresponding keys $\underline{K}_i$ for level i; • The receiver performs forward substitution on the packets using the locked coefficients to recover the original packets; • The receiver decrypts the encrypted symbols to form the original plaintext.

B. Bringing Security to Multiresolution Video: Triangular Encoding Matrix

As we have seen, upon generating a new GoP, the source divides it into vectors $\underline{b}^{(1)} \ldots \underline{b}^{(w)}$, mixing all layers, and applies the matrix $\mathbf{A}$ to each of them to obtain the payload, that is, $\underline{c}^{(i)} = \mathbf{A}\underline{b}^{(i)}$. To achieve security, the key idea is to encrypt each line of the matrix $\mathbf{A}$ using a different layer key, as illustrated by the example in *Fig. 5*. Note that only the recipients with the corresponding keys can decode the encrypted line, and consequently the layer. Standard network coding operations can be employed over the *unlocked coefficients* also when the layers are encrypted with different keys. Furthermore, even if packets from different layers are combined, reverting the operations through the use of unlocked coefficients subsequently reverts *all* combinations of different layers, so that the original information can be recovered[3].

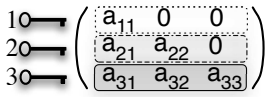

Fig. 5. Illustration of the encryption of the locked coefficients. The first layer corresponds to the first line of the matrix and is encrypted with the key for layer 1. The remaining locked coefficients are encrypted line by line according to a similar mechanism.

Note that traditional RLNC mixes all packets by using a full square matrix. This, however, is not suitable for layered coding, since it is not possible to extract individual layers unless one matrix is used for each layer. Our triangular matrix coding effectively mixes the layers, allowing for differentiated recovery of successive layers by nodes with different access levels, while relying on the dissemination of lower-level packets to achieve the resilience necessary for higher-level packets to be delivered in a timely fashion. Moreover, the triangular matrix form provides priority to the base layer, as all upper layer packets contain the base layer. Thus, the common prioritization and scheduling of the base layer is solved in a natural way. In *Section V-B* we compare our scheme with traditional RLNC addressing scheduling and prioritization issues.

The choice of a triangular matrix further meets two important requirements. First, it allows us to remove the arbitrary delay introduced by the typical RLNC full-matrix at the source, since the source can code packets as soon as they are generated and does not have to wait for the end of the generation to send them. Furthermore, the use of a triangular matrix also allows for a unique mapping between the unlocked and locked coefficients that does not compromise security: a non-zero unlocked coefficient in column i corresponds to the combination of packets $p_1, \ldots, p_i$ inside the corresponding packet. This is a way of determining the layer of a packet at relay nodes and allow the use of the feedback strategies for minimizing the decoding delay mentioned in *Section I*.

C. Security analysis

We now introduce the model used to perform the security analysis, which is similar to the one in [21]. Let $\mathbf{A} = (a_{ij})$ be the $n \times n$ lower triangular encoding matrix used for performing coding at the source. Each of the non-zero coefficients $a_{ij}, i \geq j$ is uniformly distributed over all non-zero elements of a finite field $\mathbb{F}_q$, $q = 2^u$, and mutually independent.

Let the original data, or plaintext, be a sequence of w vectors $\underline{b}^{(1)} \ldots \underline{b}^{(w)}$, in which $\underline{b}^{(x)} = (b_1^{(x)}, b_2^{(x)}, \ldots, b_n^{(x)})^T$, $1 \leq x \leq w$. All vectors $\underline{b}^{(x)}$ are independent of $\mathbf{A}$. We assume that the successive refinement algorithm used to generate the scalable video is optimal. Thus, $P(B_i = b_i) = (q-1)^{-1}, \forall b_i \in$

[3]For simplicity of the discussion, and without loss of generality, we consider matrix $\mathbf{A}$ to have one row per layer.

$\mathbb{F}_q \backslash \{0\}$. For simplicity in the proofs, we also consider that the plaintext is pre-coded to remove zeros. This can be achieved by mapping elements of $\mathbb{F}_q$ into $\mathbb{F}_{q-1}$, thus incurring a negligible rate penalty of $(q-1)/q$.

We generalize the proofs to include more than one encrypted symbol per use of the encoding matrix, and represent the number of encrypted symbols per reuse of the encoding symbols as m. We abstract from the particular cypher used for locking the coefficients. For the plaintext, we consider the use of a stream cypher such that the probability of the output of the encoding operation $E(P, \underline{K})$ is independent of the plaintext P and the distribution of the output is uniform among all non-zero elements of $\mathbb{F}_q \backslash \{0\}$, that is, $P(E(P, \underline{K})) = (q-1)^{-1}$. The parameters of the cypher should be adjusted to approximate these criteria [22]. In the proofs, to obtain these properties, we consider the use of a one time pad in which one symbol of the key is used for each symbol of the plaintext that is encrypted. The key is represented by w random vectors $\underline{K}^{(1)} \dots \underline{K}^{(w)}$, each with m positions (that is, with wm symbols of key in total). Furthermore, $P(K_i = k_i) = (q-1)^{-1}, \forall k_i \in \mathbb{F}_q \backslash \{0\}$.

We denote the vector to which the matrix is applied, that is, the vector $(E(b_1, K_1^{(1)}), \dots, E(b_m^{(x)}, K_m^{(x)}), b_{m+1}^{(x)}, \dots, b_n^{(x)})^T$, as $\underline{e}^{(x)}$. Each payload vector is represented by $\underline{c}^{(x)} = (c_1^{(x)}, \dots, c_n^{(x)})^T$, where x corresponds to reuse x of $\mathbf{A}$ and

$$c_i^{(x)} = \sum_{j=1}^{\min(m,i)} a_{ij} E(b_j^{(x)}, K_j^{(x)}) + \sum_{l=m+1}^{i} a_{il} b_l^{(x)}.$$

In all the proofs, random variables are described in capital letters and instances of random variables are represented in lowercase letters. Vectors are represented by underlined letters and matrices are represented in boldface.

Without loss of generality, we abstract from the network structure and consider the payload of all packets together in the security proofs. We characterize the mutual information [23] (denoted by $I(\cdot;\cdot)$) between the encoded data and the two elements that can lead to information disclosure: the encoding matrix and the original data itself. *Theorem 1* evaluates the mutual information between the payload and the encoding matrix, and *Theorem 2* evaluates the mutual information between the payload and the original data.

Theorem 1: The mutual information between $\mathbf{A}$ and $\mathbf{A}\underline{E}^{(1)}$, $\mathbf{A}\underline{E}^{(2)}$, $\dots$, $\mathbf{A}\underline{E}^{(w)}$ is zero:

$$I(\mathbf{A}; \mathbf{A}\underline{E}^{(1)}, \mathbf{A}\underline{E}^{(2)}, \dots, \mathbf{A}\underline{E}^{(w)}) = 0.$$

Proof: See Appendix.

Theorem 1 is a generalization of the result in [24] and shows that the cost of a statistical attack on the encoding matrix is the cost of a brute-force attack on all entries of the matrix, independently of the number of reuses.

Theorem 2: The mutual information between $\underline{B}^{(1)}, \dots, \underline{B}^{(w)}$ and $\mathbf{A}\underline{E}^{(1)}$, $\dots$, $\mathbf{A}\underline{E}^{(w)}$ is given by the expression:

$$I(\underline{B}^{(1)}, \dots, \underline{B}^{(w)}; \mathbf{A}\underline{E}^{(1)}, \dots, \mathbf{A}\underline{E}^{(w)}) = \log(q-1) \max\left(f(w,n,m), 0\right),$$

where $f(w,n,m) = w(n-m) - \frac{n(n+1)}{2}$.

Proof: See Appendix.

The equation in *Theorem 2* shows that the cost of attacking the plaintext is the cost of discovering the encoding matrix. Thus, we get a threshold at which there is a reduction of the search space needed to attack the plaintext due to multiple reuses of the matrix $\mathbf{A}$. Notice that there is no disclosure of the plaintext with a single use of the encoding matrix. Below the number of uses in the threshold, the mutual information is 0 and thus, it is not possible to perform a statistical attack on the payload. When the number of uses of the encoding matrix surpasses the threshold, the mutual information grows with w. In the extreme case in which the number of encrypted symbols is equal to the number of symbols in the matrix, the mutual information is always zero (however, in this case, we would not require the encoding matrix to be hidden).

The triangular matrix grants unequal protection to the layers of the plaintext. We can easily see that the search space for discovering layer $i+1$ is larger than the search space to discover layer i. Take, for instance, the case in which $m = 0$ – then, for layers i and $i+1$, an attacker needs to guess, respectively, i and $i+1$ entries of the matrix.

We believe that the expression in *Theorem 2* allows us to fine tune the trade-off between complexity and security by varying n (the size of the matrix), m (the number of encrypted symbols) and the size of the field.

IV. System Aspects

We now discuss practical system aspects of our scheme. Let us consider a scenario such as the one in *Fig. 1*, with a system architecture as depicted in *Fig. 6*. We will discuss the different components of the system and their practical implications next.

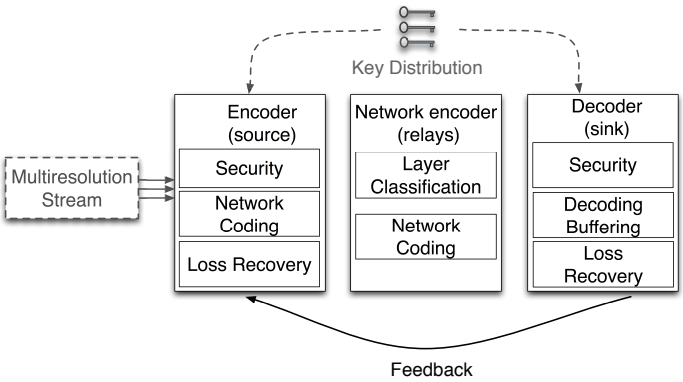

Fig. 6. Modules of a potential system implementation. Entities that are external to our system (that is, key distribution and generation of a multiresolution stream) are in dashed.

A. Key distribution

Our scheme requires shared keys between sources and destinations. While the specifics of a key distribution mechanism are not relevant for this paper, examples include offline pre-distribution of keys or authentication protocols such as Kerberos or a Public Key Infrastructure (PKI). Note that the need for keys to be shared among several legitimate nodes in a network arises frequently in multicast scenarios and is

commonly denominated as broadcast encryption or multicast key distribution [25]. Layer l nodes should keep l keys (one for each layer), and thus, the number of keys exchanged is equal to $\sum_{l=1}^{L} lt_l$, in which t_l represents the number of recipients of layer l in the network and L the total number of layers in the stream.

B. Multiresolution Encoder and Security

The main requirements of security protocols for multimedia streams [4] are (i) to work with low complexity and high encryption efficiency, (ii) to keep the file format and synchronization information and (iii) to maintain the original data size and compression ratio. As we can see from *Section III* , we have designed our scheme to meet criterion (i). Criterion (ii) is codec-dependent, but in general our scheme is able to meet it. Taking for example the MJPEG video codec[4] [26], we can use the JPEG2000 option of placing all headers from all blocks of the image on the main header of the file and satisfy criterion (ii). Finally, network coding does not change the size or compression ratio of the stream, so our scheme satisfies criterion (iii).

As shown in *Section III-C*, the maximum level of security is obtained when the compression is optimal and yields a result that is nearly uniform. Thus, our scheme imposes a set of parameters for the codec in order to maximize the entropy of the file. In the MJPEG codec, two such coding decisions would be to choose larger tile sizes and maximum compression rate on the arithmetic coding step. Another approach would be to perform an extra data protection step together with compression (see [26]). The size of the base layer can be seen as another parameter to increase the compression ratio. As an example, in JPEG2000, each encoded symbol increases the resolution of the stream, therefore it is possible to vary the size of each layer taking the constraints of the security mechanism into consideration.

C. Source Encoder

The *source encoder* includes *security*, *loss recovery* and *network coding* modules. The *security module* and its interoperation with *network coding* are described in *Section III*. However, we do use more than one row of the matrix for each layer. In that case, the mapping between the unlocked and locked coefficients suffers a shift: if 2 packets per layer are used, a packet with unlocked coefficients vector $(1, 1, 0, \ldots 0)$ belongs to layer 1 and a packet with vector $(1, 1, 1, 0, \ldots 0)$ belongs to layer 2. The division of the payload into vectors should also accommodate this shift. Codecs in which each new symbol (decoded in order) contributes to increased resolution of the output video (such as the MJPEG2000) might benefit from an approach with a finer granularity. This granularity can be fine-tuned by the number of lines of the encoding matrix that belong to each layer. Another important system requirement is to use an encryption mechanism for which the ciphertext is of the same size of the plaintext (e.g. AES in stream cipher mode) in order to keep the size of the symbols constant.

An important aspect of the encoder is the rate at which intermediate nodes generate and send linear combinations to the receiver. If a relay generates and forwards a linear combination every time he receives an innovative packet from the server, then many redundant packets may arrive at destinations. To solve this issue, the server generates a credit for each coded packet, which is further assigned to one of the intermediate relays [27]. Next, only the relay who receives also the credit associated with the packet is allowed to send a linear combination.

After transmitting a complete generation, and before streaming the next one, the server starts the loss recovery process. To recover lost packets, the server sends redundant linear combinations for each layer, mixing all packets of the layer. This process continues until all the receivers for that layer can decode or the server has another segment to stream.

D. Network (Relay) Encoder

The *network encoder* is a component of the wireless relays of the network and includes *layer classification* and *network coding*. As mentioned in *Section III*, packets of layer l should only be combined with packets of lower layers, i.e. , $l, l-1, \ldots 1$. This is done in order to maintain the diversity of layers in the network, because when combining a packet of layer l with layer $l+1$, the layer of the resulting packet is $l+1$. After classifying the packet, a relay generates and forwards a linear combination if he received the credit assigned to that packet.

E. Decoder

The *decoder* is a component of the receiver that includes *security*, *decoding and buffering* and *feedback*. When enough packets are received, the receiver performs Gaussian elimination to decode packets using the unlocked coefficients. The security process corresponds to the recovery of the locked coefficients and encrypted symbols of the payload and is explained in *Section III*.

Since in our scheme relay nodes perform coding on the packets of the same (and lower) layers, the shape of the triangular matrix sent by the source is not kept through the network. Thus, a received packet, even if innovative in terms of rank, might not be decodable immediately. Hence, our system requires a decoding buffer at the receivers. This decoding buffer takes into account the maximum allowable delay of the video stream, similar to the play buffer at the receivers, and will preemptively flush the current undecoded packets if the delay requirement is not met. Once a full layer is decoded, it is stored in the playback buffer.

A node starts the playback once it decodes a number of segments in the lowest quality. If a frame is not received until the time of playback, then it is discarded and the subsequent frame is played instead. Likewise, if the frame is available in a lower quality, it is played in a lower quality than the one the node has access to. At timestep k the node plays segment k in the quality in which it is available. If the segment was not decoded not even in the lowest quality, then the node

[4]In MJPEG, several JPEG2000 images are concatenated to generate a video stream. Each image is compressed separately.

stops the playback process and starts buffering. If after some buffering timeout, the node decodes segment k, then it plays it in the quality in which it is available; otherwise, the node skips segment k and plays the next one.

We consider a system with minimal feedback, in order to free the wireless channels from unnecessary transmissions. The receivers send positive feedback to the server whenever they decode a segment in the desired quality. For example, a layer 3 receiver sends a unique feedback packet when it has decoded layers 1, 2 and 3.

V. Evaluation

In this section we evaluate our system in terms of security complexity and we provide an evaluation of its performance in a lossy wireless scenario.

A. Security Performance

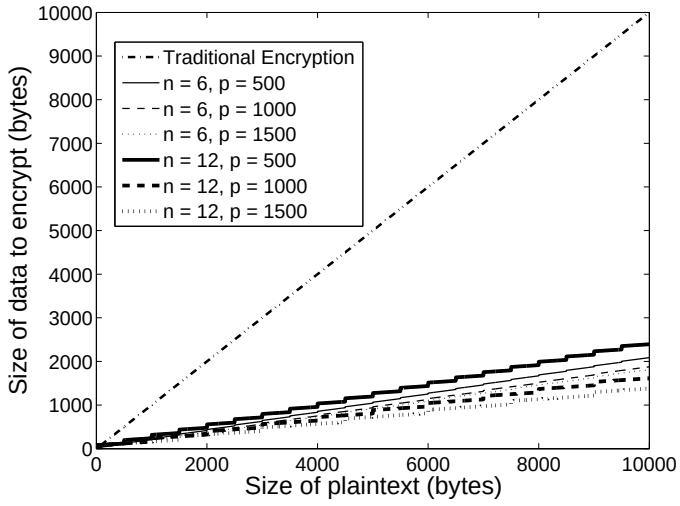

Fig. 7. Size of data to be encrypted for our scheme versus traditional encryption (encryption of the whole data).

1) Encryption volume: Fig. 7 compares the volume of data to be encrypted according to the size of the plaintext for our scheme and traditional encryption, for typical packet sizes of 500 bytes (for video packets in cellular networks) [28], 1000 bytes (for example, for video over wifi networks) and 1500 bytes (the typical IP packet size). We consider one encrypted symbol per generation. For the traditional encryption mechanism, which performs end-to-end encryption of the entire payload, the volume of data that must be encrypted increases linearly with the size of the protected payload. It is not difficult to see that our scheme substantially reduces the size of information to be encrypted. The gains get higher as the maximum size of the packet increases, since the number of matrices to be generated is smaller, and more data can be sent in each packet containing the same matrix of coefficients.

Naturally, the required number of cryptographic operations is directly related to the volume of data to be encrypted. If we consider a stream cipher, the number of encryption operations increases linearly with that volume, and therefore, the computational complexity is greatly reduced by our scheme as shown in *Fig. 7*. Note that these values are indicative only, and correspond to the theoretical gains when the size of the packet is the only parameter determining the number of reuses of the encoding matrix. The security penalty, which is quantified in *Section III-C*, is not considered for the purposes of this analysis. Note as well that the end values depend on the design of the codec, as well as on the size chosen for each layer.

TABLE II
Volume overhead of locked coefficients (per packet).

Maximum IP packet size	#Coded packets h	Overhead in $\mathbb{F}_q$ $q=2^8$	$q=2^{16}$
500	4	0.80%	1.60%
	8	1.60%	3.20%
	12	3.20%	6.40%
1000	4	0.40%	0.80%
	8	0.80%	1.60%
	12	2.40%	4.80%
1500	4	0.27%	0.53%
	8	0.53%	1.07%
	12	0.80%	1.60%

2) Communication and Computational overhead: The ability to reduce the volume of data to be encrypted comes at the cost of including locked coefficients in the data packet. In *Table II* we show the overhead introduced by our scheme for each packet and for coefficients with size of 8 and 16 bits, for some values of reference for wireless networks with nodes with several processing capabilities. Note that the inclusion of locked and unlocked coefficients allows us to avoid the use of *homomorphic hash functions*, which are very expensive in terms of computation [29]. Due to the inclusion of an extra set of coefficients (the locked coefficients), our scheme requires additional operations, which are shown in *Table III*. For the purpose of our analysis, we consider that, in comparison to the multiplication, the sum operation yields negligible complexity.

TABLE III
Computational cost of including the locked coefficients

Node	Operation	Detailed Cost	Total Cost
Source Node	Generation of vectors of identity matrix	negligible	–
	Encryption of locked coefficients	See *Section V-A.1*	
Relay Node	Performing extra random linear operations on locked coefficients (combining t packets)	nh multiplication operations and $(n-1)h$ sum operations	$O(nt)$
Sink node	Decrypt locked coefficients to obtain the matrix M_L of plain-text locked coefficients	See *Section V-A.1*	$O(n^2)$
	Forward-substitution using recovered locked coefficients	$O(n^2)$	
	Decrypt one encrypted symbol per use of the encoding matrix	See *Section V-A.1*	

B. Wireless Video Performance

We evaluate the performance of the protocol described in *Section IV* in the multi-hop multi-path scenario from *Fig. 1*, in which the server S sends video to 3 heterogenous receivers A, B and C, through relays R_1, R_2 and R_3, over lossy wireless links. In this section we will focus solely on the performance

of the scheme in terms of throughput and robustness to losses, and its ability to deliver quality video to a heterogeneous set of receivers.

We compare our layered network coding model (*scheme NC1*) with standard RLNC (*scheme NC2*) and an implementation without network coding (*scheme WoNC*). In *scheme NC2* the server sends a different stream for every layer. Each segment is encoded in different qualities, using a full coefficient matrix for each layer. Relay nodes perform RLNC operations on the received packets that belong to the same generation and to the same or lower layers. In this case, since a sink of layer L needs to receive a full-rank matrix for layers $1, 2, \ldots L$, sinks acknowledge each layer that they decode. Error recovery is similar to *scheme NC1*. In *scheme WoNC*, the server sends the native packets without coding them. In this case, the intermediate nodes just forward uncoded packets normally. The sinks send as feedback the *ids* of the packets they received. If some packets are lost, the server retransmits them.

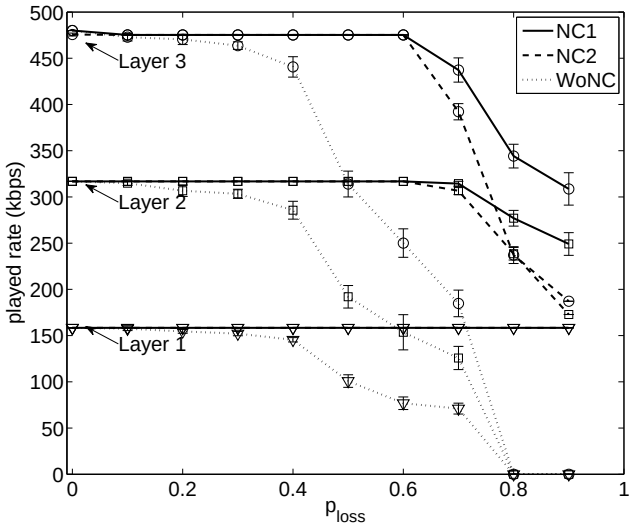

Fig. 8. Played rate in function of loss probability p_{loss}, for our scheme (*NC1*), three streams with network coding (*NC2*) and without network coding (*WoNC*).

Simulation Setup

We use the ns-2 simulator 2.33 [30], with the default random number generator for this version. The network coding libraries are independently programmed. The video stream is a constant bit rate traffic over UDP, where the server is streaming at 480 kbps during 100 seconds. Each layer has a fixed size of 20 packets and we consider 3 layers for the system, which yields a generation of 60 packets, corresponding to 1 second of video. The packet size is 1000 bytes. As a propagation model, we use *two-ray ground* and we consider the loss probability p_{loss} as a simulation parameter. Since it was shown that RTS/CTS has a negative impact on the performance, we disable it for all experiments. In order to simulate heavy loss conditions, we also disable MAC layer retransmissions. The rate at the MAC layer is 11 Mbps.

The receivers start to playback the video stream once they have decoded at least 5 segments of the lowest quality. The buffering timeout for a segment that has not been decoded until its playback deadline arrives is set to 1 second. Furthermore, we consider a perfect feedback channel (that is, no feedback packets are lost). In order to take full advantage of the broadcast nature of the wireless medium, the relays listen to transmitted packets in promiscuous mode.

We consider the following metrics: (i) *played rate* at the receivers, (ii) *initial buffering delay*, the time interval from receiving the first packet to the beginning of the playback, (iii) *decoding delay*, the time elapsed from receiving the first packet of a a segment until that segment is decoded, (iv) *skipped segments*, percentage of segments skipped at playback, (v) *lower quality segments*, percentage of segments played in lower quality than the one requested, (vi) *playback quality*, average quality in which each segment is played and (vii) *load on the server*, defined as the ratio between the total rate sent by the server and the streaming rate. In all plots, each point is the average of 10 runs and the vertical lines show the standard deviation.

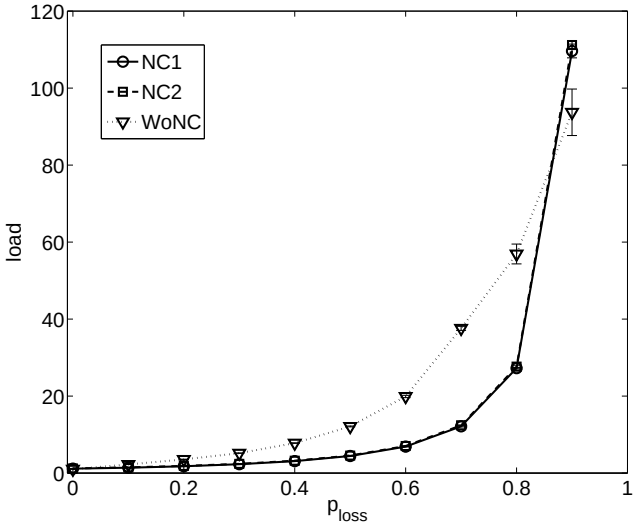

Fig. 9. The load on the server in function of the loss probability p_{loss}.

Results

Fig. 8 shows the rate played by each receiver vs. loss probability. *Scheme NC1* and *scheme NC2* are less affected by losses, due to the inherent reliability of network coding in volatile environments, with our scheme performing consistently better. *Scheme WoNC*, as expected, performs poorly as the medium becomes unreliable. We can see in *Fig. 9* that the load on the server grows exponentially as the loss increases. In general, the network coding approaches need to send less coded packets to recover losses. At $p_{loss} = 0.9$, the load is slightly higher for network coding since the server preemptively sends redundant packets until it receives the feedback from the receiver that the segment is decoded, while for *scheme WoNC* the server retransmits packets only when it receives feedback from the receivers. Since most of the packets are dropped, *scheme WoNC* never retransmits.

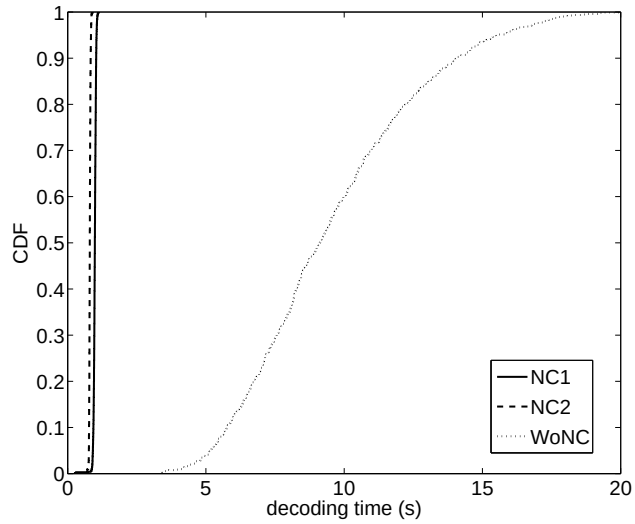

Fig. 10. CDF of decoding delay for loss probability $p_{loss} = 0.4$, for layer 3.

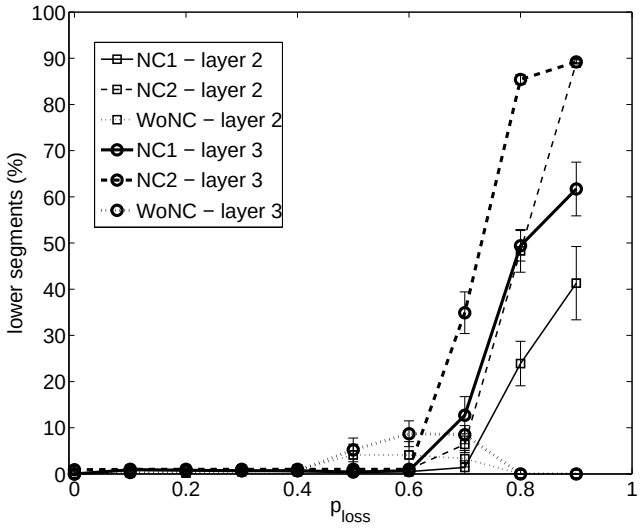

Fig. 12. The percentage of segments played in lower quality in function of the probability of loss p_{loss}.

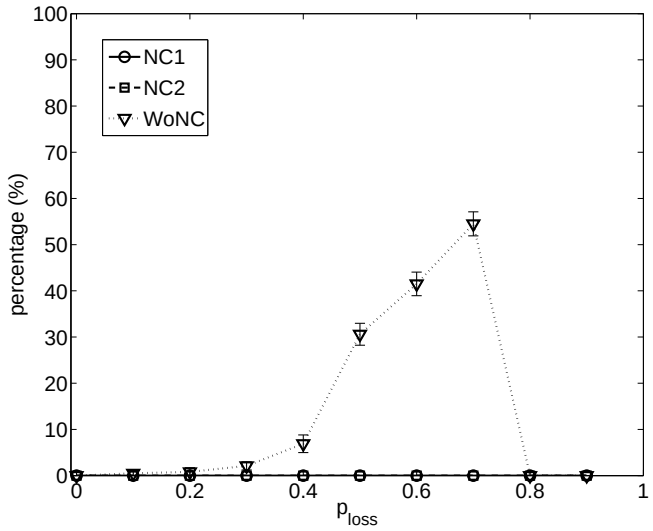

Fig. 11. The percentage of skipped segments with the probability of loss, p_{loss}, for layer 3.

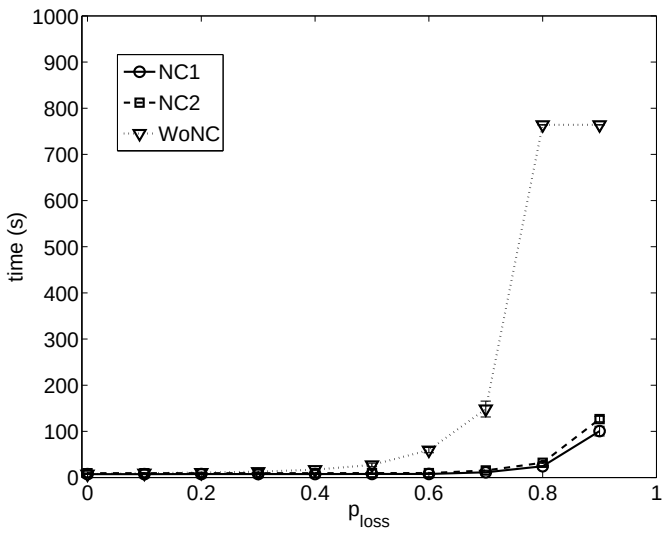

Fig. 13. Initial buffering delay in function of loss probability p_{loss}, for layer 3.

Fig. 10 shows that the network coding approaches are able to decode segments within a second as the server sends redundant linear combinations in a feed-forward manner. *Scheme WoNC* needs a longer decoding time, because the server waits for the feedback before retransmitting. The plot shown corresponds to a layer 3 receiver and the behavior for other layers is similar.

Figs. 11 and *12* show the percentage of segments that are skipped and played in lower quality, respectively. Note that with network coding, no segments are skipped for any layers, and, as expected, more segments are played in lower quality as the losses increase. On the other hand, without network coding, there are fewer segments played in lower quality, but at the same time the percentage of skips grows significantly with p_{loss}, because the packets retransmitted by the server do not arrive at the receivers in due time. This effect is exacerbated at higher losses, where no segment is ever played (and hence never skipped either).

We can see in *Fig. 13* that for our scheme, the receivers buffer for a shorter time before starting the playback. The initial buffering delay grows slowly with the probability of loss, because a single network coded packet can recover multiple losses. For *scheme WoNC*, when losses are high, the receivers are not able to decode anything, thus they never start to play the file.

The plots shown in *Figs. 11* and *13* correspond to layer 3. The behavior for other layers is similar and slightly better, since layer 3 receivers need to receive more packets than lower layer nodes.

Fig. 14 shows the average quality in which every segment is played, when $p_{loss} = 0.4$. A skipped segment accounts as played in a quality equal to 0. Note that the network coding approaches show a high resilience to errors and the video file is constantly played in the desired quality by each receiver

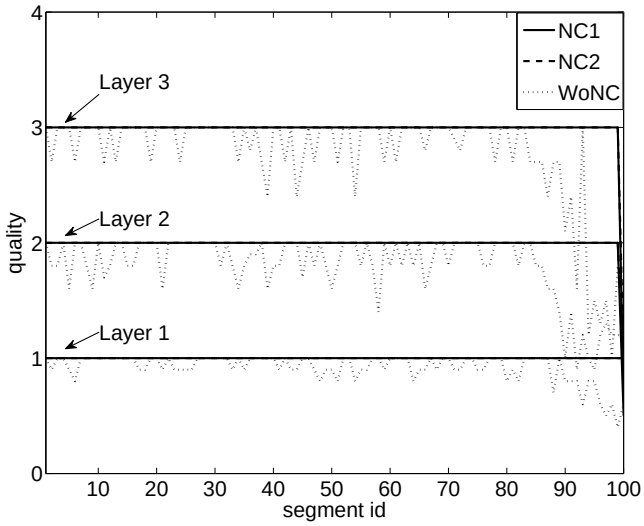

Fig. 14. Played quality for $p_{loss} = 0.4$.

compared to *scheme WoNC*, again with our scheme showing better performance.

Finally, note that our scheme outperforms *scheme NC2* due to the triangular encoding matrix used for coding and to the nested structure of the video layers. These characteristics result in a higher robustness to losses (*Fig. 8*), better video quality with fewer skips and fewer segments played in lower quality (*Fig. 12*) and shorter buffering delay (*Fig. 13*).

VI. Conclusions and Further Work

We presented a practical scheme for scalable video streaming that exploits the algebraic characteristics of Random Linear Network Coding. On the one hand our proposal ensures differentiated levels of security for distinct users. On the other hand, the properties of the network coding paradigm assure the resilience to packet losses over wireless channels. The security evaluation proves that it is possible to reduce significantly the number of encryption operations (or, equivalently, the complexity requirements) while quantifying the security levels. Our work was focused on eavesdropping attacks. Network pollution attacks can be dealt with using the techniques in [31] albeit at some cost in terms of delay and complexity. As part of our ongoing work we are looking at ways to mitigate the effects of such Byzantine attacks under the real-time constraints of streaming services.

Acknowledgements

The authors would like to acknowledge the help of Rui Costa and Tiago T. V. Vinhoza (Universidade do Porto) in the mathematical proofs presented in the paper, and insightful discussions with Manuel Barbosa (Universidade do Minho), Matthieu Bloch and Demijan Klinc (Georgia Institute of Technology), as well as João P. Vilela and João Mendes (Universidade do Porto).

References

[1] A.S. Tosun and W. Feng, "Lightweight Security Mechanisms for Wireless Video Transmission," *Proc of the Intl. Conf. on Information Technology: Coding and Computing*, pp. 157–161, 2001.

[2] A.S. Tosun and W.C. Feng, "Efficient multi-layer coding and encryption of MPEG video streams," *Proc. of the 2000 IEEE International Conference on Multimedia and Expo (ICME 2000)*, vol. 1, 2000.

[3] A. S. Tosun and W. C. Feng, "Lightweight security mechanisms for wireless video transmission," *International Conference on Information Technology: Coding and Computing*, vol. 0, pp. 0157, 2001.

[4] Shiguo Lian, *Multimedia Content Encryption: Techniques and Applications*, Auerbach Publications, Boston, MA, USA, 2008.

[5] J. Kritzner, U. Horn, M. Kampmann, and J. Sachs, "Priority Based Packet Scheduling with Tunable Reliability for Wireless Streaming," *Lecture Notes in Computer Science*, pp. 707–717, 2004.

[6] R. Ahlswede, N. Cai, S.Y.R. Li, and Raymond W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[7] J. Widmer and J.Y. Le Boudec, "Network coding for efficient communication in extreme networks," *Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp. 284–291, 2005.

[8] C. Fragouli, D. Katabi, A. Markopoulou, M. Medard and H. Rahul, "Wireless Network Coding: Opportunities & Challenges," in *MILCOM 2007*, Orlando, FL, October 2007.

[9] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *Proc. of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 243–254, 2006.

[10] J. Jin, B. Li, and T. Kong, "Is Random Network Coding Helpful in WiMAX?," in *IEEE 27th Conference on Computer Communications (INFOCOM 2008)*, 2008, pp. 2162–2170.

[11] C. Fragouli, D. Katabi, A. Markopoulou, M. Medard, and H. Rahul, "Wireless network coding: Opportunities & challenges," in *IEEE Military Communications Conference, 2007. MILCOM 2007*, 2007, pp. 1–8.

[12] J. Widmer R. A. Costa, D. Munaretto and J. Barros, "Informed network coding for minimum decoding delay," in *Fifth IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, Atlanta, Georgia, USA, September 2008.

[13] H. Seferoglu and A. Markopoulou, "Opportunistic network coding for video streaming over wireless," *Packet Video 2007*, pp. 191–200, 2007.

[14] N. Sundaram, P. Ramanathan, and S. Banerjee, "Multirate Media Streaming Using Network Coding," *Proc. 43rd Allerton Conference on Communication, Control, and Computing, Monticello, IL, Sep*, 2005.

[15] P. Frossard, J.C. de Martin, and M. Reha Civanlar, "Media streaming with network diversity," *Proceedings of the IEEE*, vol. 96, no. 1, pp. 39–53, Jan. 2008.

[16] J. P. Vilela, L. Lima, and J. Barros, "Lightweight Security for Network Coding," *Proc. of the IEEE International Conference on Communications (ICC 2008), Beijing, China*, pp. 1750–1754, May 2008.

[17] L. Lima, M. Médard, and J. Barros, "Random linear network coding: A free cypher?," in *IEEE International Symposium on Information Theory, Nice, France*, June 2007.

[18] K. Han, T. Ho, R. Koetter, M. Medard, and F. Zhao, "On network coding for security," in *IEEE Military Communications Conference (MILCOM 2007)*, Oct. 2007, pp. 1–6.

[19] Z. Liu, Y. Shen, S. S. Panwar, K. W. Ross, and Y. Wang, "Using layered video to provide incentives in p2p live streaming," in *P2P-TV '07: Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV*, New York, NY, USA, 2007, pp. 311–316, ACM.

[20] T. Ho, M. Médard, R. Koetter, D.R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[21] L. Lima, J. P. Vilela, J. Barros, and M. Médard, "An Information-Theoretic Cryptanalysis of Network Coding – is protecting the code enough?," *Proc. of the International Symposium on Information Theory and its Applications, Auckland, New Zealand*, Dec. 2008.

[22] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A concrete security treatment of symmetric encryption," in *Proc. of the 38th Annual Symposium on Foundations of Computer Science, 1997*, 1997, pp. 394–403.

[23] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory*, Wiley-Interscience, August 1991.

[24] P.F. Oliveira and J. Barros, "A Network Coding Approach to Secret Key Distribution," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 414–423, 2008.

[25] MJ Moyer, JR Rao, and P. Rohatgi, "A survey of security issues in multicast communications," *IEEE Network*, vol. 13, no. 6, pp. 12–23, 1999.

[26] D. T. Vo and T. Q. Nguyen, "Quality enhancement for motion jpeg using temporal redundancies," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 609–619, May 2008.
[27] B. Radunovic, C. Gkantsidis, D. Gunawardena, and P. Key, "Horizon: balancing tcp over multiple paths in wireless mesh network," in *Proc. of the 14th ACM international conference on Mobile computing and networking (MobiCom '08)*, New York, NY, USA, 2008, pp. 247–258.
[28] TIA/EIA IS-707-A-2.10, *Data Service Options for Spread Spectrum Systems: Radio Link Protocol Type 3*, Jan. 2000.
[29] C. Gkantsidis and P.R. Rodriguez, "Cooperative security for network coding file distribution," *Proc. of the IEEE Infocom 2006, Barcelona, Spain*, 2006.
[30] S. Mccanne, S. Floyd, and K. Fall, "ns2 (network simulator 2)," http://www-nrg.ee.lbl.gov/ns/.
[31] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, "Resilient Network Coding In the Presence of Byzantine Adversaries," *Proc. of the IEEE INFOCOM 2007, Anchorage, Alaska, USA*, 2007.

APPENDIX

Proof for Theorem 1

We restrict our presentation to the main ideas for the proof due to lack of space. For compactness, we write line i of $\mathbf{A}$ as $\mathbf{A}_i$. The set of lines $i \dots l$ of the matrix $\mathbf{A}$ is represented as $\mathbf{A}_{i:l}$, and the vector formed by the positions $i \dots l$ of the vector $\underline{b}$ is represented as $\underline{b}_{i:l}$. First, we have that

$$I(\mathbf{A}\underline{E}^{(1)}, \dots, \mathbf{A}\underline{E}^{(w)}; \mathbf{A}) = H(\mathbf{A}) - H(\mathbf{A}|\mathbf{A}\underline{E}^{(1)}, \dots, \mathbf{A}\underline{E}^{(w)})$$

Now, we can reorder the random variables $\underline{C}_i^{(x)}$ in the expression $H(\mathbf{A}|\underline{C}^{(1)}, \dots, \underline{C}^{(w)})$ by line and then by reuse (corresponding to the first use of line i of $\mathbf{A}$, followed by the second use of the same line, etc.). Then, by applying the chain rule of entropy, we obtain:

$$\begin{aligned} &H(\mathbf{A}_1, \dots, \mathbf{A}_n | C_1^{(1)}, \dots C_1^{(w)}, \dots, C_n^{(1)}, \dots C_n^{(w)}) = \quad (1)\\ &H(A_{11}|C_1^{(1)}, \dots, C_n^{(w)}) + H(A_{21}|C_1^{(1)}, \dots C_n^{(w)}, A_{11}) + \\ &H(A_{22}|C_1^{(1)}, \dots, C_n^{(w)}, A_{11}, A_{21}) + \dots + \\ &H(A_{nn}|C_1^{(1)}, \dots, C_n^{(w)}, A_{11}, \dots A_{nn-1}) \end{aligned}$$

We now consider each of the terms of this equation separately. The general term $H(A_{ij}|C_1^{(1)}, \dots, C_n^{(w)}, A_{11}, \dots, A_{ij-1})$ is conditioned on all entries $\mathbf{A}_{1:i-1}, A_{i1} \dots A_{ij-1}$. Note that from $\mathbf{A}_{1:i-1}$ and $C_1^{(1)} \dots C_1^{(w)}, \dots C_{i-1}^{(1)}, \dots, C_{i-1}^{(w)}$ it is possible to obtain $b_1^{(1)}, \dots, b_{i-1}^{(w)}$. We have that

$$\begin{aligned} &H(A_{ij}|C_1^{(1)}, \dots, C_n^{(w)}, A_{11}, \dots, A_{ij-1}) \leq \\ &H(A_{ij}|C_1^{(1)}, \dots, C_n^{(w)}, A_{11}, \dots, A_{ij-1}, A_{ij+1} \dots, A_{nn}). \end{aligned}$$

The strategy is to condition on all entries of $\mathbf{A}$ except for A_{ij}. Now, on the right-hand side of the conditional, we have a system of equations. In order to determine A_{ij} from the system of equations determined by these conditions, it suffices to discover one of the variables $B_i, \dots, B_n$, thus

$$\begin{aligned} &H(A_{ij}|C_1^{(1)}, \dots, C_n^{(w)}, A_{11}, \dots, A_{ij-1}, A_{ij+1} \dots, A_{nn}) = \\ &H(B_i^{(1)}) = \dots = H(B_i^{(w)}) = \dots = H(B_n^{(1)}) = \dots = H(B_n^{(w)}). \end{aligned}$$

Since, by assumption, $H(B_k^{(x)}) = H(A_{ij})\ \forall x, k, i, j$, then:

$$\begin{aligned} &H(A_{ij}|C_1^{(1)}, \dots, C_n^{(w)}, A_{11}, \dots, A_{ij-1}, A_{ij+1} \dots, A_{nn}) \\ &= H(A_{ij}). \end{aligned}$$

Furthermore, from [24], we have that when A_{ij} appears in multiple equations (for example in $A_{ij}b_1^{(1)} + B_i^{(1)}a_{ij+1} = c'_1, \dots, A_{ij}b_1^{(w)} + B_i^{(w)}a_{ij+1} = c'_w$, where $c'_1, \dots c'_w$ are obtained by subtraction of the constants in the right-hand side of the equations) then $H(A_{ij}|A_{ij}b_1^{(1)} + B_i^{(1)}a_{ij+1} = c'_1, \dots, A_{ij}b_1^{(w)} + B_i^{(w)}a_{ij+1} = c'_w) = H(A_{ij})$.

The final result can be obtained by induction on the number of lines of the matrix and reuses. Then, $I(\mathbf{A}\underline{E}^{(1)}, \dots, \mathbf{A}\underline{E}^{(w)}; \mathbf{A}) \leq H(\mathbf{A}) - (H(A_{11}) + \dots + H(A_{nn}))$, and since $I(\cdot;\cdot) \geq 0$, the result follows.

Proof for Theorem 2

We only provide the main ideas for the proof due to lack of space. We start by noting that

$$I(\underline{C}^{(1)}, \dots, \underline{C}^{(w)}; \underline{B}^{(1)}, \dots, \underline{B}^{(w)}) = H(\underline{B}^{(1)}, \dots, \underline{B}^{(w)}) - \sum_{\underline{c}^{(1)} \dots \underline{c}^{(w)}} H(\underline{B}^{(1)}, \dots, \underline{B}^{(w)}|\underline{c}^{(1)}, \dots, \underline{c}^{(w)}) P(\underline{c}^{(1)}, \dots, \underline{c}^{(w)})$$

Now, we take

$$\begin{aligned} &P(\underline{B}^{(1)}, \dots, \underline{B}^{(w)}|\underline{C}^{(1)}, \dots, \underline{C}^{(w)}) \\ &= \sum_{\mathbf{A} \in \mathcal{S}_A} P(\underline{B}^{(1)}, \dots, \underline{B}^{(w)}|\underline{C}^{(1)}, \dots, \underline{C}^{(w)}, \mathbf{A}) P(\mathbf{A}|\underline{C}^{(1)}, \dots, \underline{C}^{(w)}) \\ &= \sum_{\mathbf{A} \in \mathcal{S}_A} \sum_{\underline{K} \in \mathcal{S}_K} P(\underline{B}^{(1)}, \dots, \underline{B}^{(w)}|\underline{C}^{(1)}, \dots, \underline{C}^{(w)}, \mathbf{A}, \underline{k}) P(\underline{k}) P(\mathbf{A}) \end{aligned}$$

From *Theorem 1* we have that $P(\mathbf{A}|\underline{C}^{(1)}, \dots, \underline{C}^{(w)}) = P(\mathbf{A})$. Given $\underline{c}^{(1)} \dots \underline{c}^{(w)}$, $\mathbf{A}$ and $\underline{k}$ it is possible to recover $\underline{B}^{(1)}, \dots, \underline{B}^{(w)}$ uniquely and so $P(\underline{B}^{(1)}, \dots, \underline{B}^{(w)}|\underline{C}^{(1)} = \underline{c}^{(1)}, \dots, \underline{C}^{(w)} = \underline{c}^{(w)}, \mathbf{A}, \underline{K}) = 1$. For simplicity we assume that the used key $\underline{K}$ is of the size of the text to be encrypted and that each of its symbols is independent and uniformly distributed. It follows that $P(\underline{k}) = (q-1)^{-wm}$, in which m is the number of symbols of $\underline{B}$ encrypted for each use of the encoding matrix. The probability of each matrix is equal to $(q-1)^{-n(n+1)/2}$, since each of its $n(n+1)/2$ symbols occurs with equal probability and belongs to $\mathbb{F}_q \backslash \{0\}$. The size of set $\mathcal{S}_K$ is 1, since there is only one key that can generate $\underline{E}^{(1)} \dots \underline{E}^{(w)}$ from $\underline{B}^{(1)} \dots \underline{B}^{(w)}$. The size of set $\mathcal{S}_A$ is the number of degrees of freedom left when both $\underline{c}^{(1)}, \dots, \underline{c}^{(w)}$ and $\underline{b}^{(1)}, \dots, \underline{b}^{(w)}$ are given. It is equal to $|\mathcal{S}_A| = (q-1)^{\max\left(\frac{n(n+1)}{2} - wn + wm, 0\right)}$. It follows that

$$\begin{aligned} &P(\underline{B}^{(1)}, \dots, \underline{B}^{(w)}|\underline{C}^{(1)}, \dots, \underline{C}^{(w)}) \\ &= (q-1)^{\max\left(\frac{n(n+1)}{2} - wn + wm, 0\right)} (q-1)^{-n(n+1)/2} (q-1)^{-wm} \end{aligned}$$

Thus, $P(\underline{B}^{(1)}, \dots, \underline{B}^{(w)}|\underline{C}^{(1)}, \dots, \underline{C}^{(w)})$ does not depend on $\underline{C}^{(1)} \dots \underline{C}^{(w)}$, and:

$$\begin{aligned} &I(\underline{C}^{(1)}, \dots, \underline{C}^{(w)}; \underline{B}^{(1)}, \dots, \underline{B}^{(w)}) \\ &= \log(q-1) \left(f(w,n,m) + \max\left(-f(w,n,m), 0\right)\right), \end{aligned}$$

where $f(w,n,m) = w(n-m) - \frac{n(n+1)}{2}$. The result follows. ■

Luísa Lima received her degree in Computer Science and Network Engineering at the Universidade do Porto, Portugal, in 2005. She is currently pursuing the Ph.D. degree in the same University and is a researcher in the Networking and Information Processing Group (NIP) of the Instituto de Telecomunicações (IT). She collaborates regularly with the Research Laboratory of Electronics at MIT.

Luísa's research interests include network coding, security, random graphs, video streaming and computer simulation. She was awarded the Doctoral Scholarship from the Portuguese Foundation for Science and Technology and the Best Student Award for her undergraduate studies.

Steluta Gheorghiu received the Engineering degree in Computer Science and Automatic Control at Polytechnic University of Bucharest, Romania, in 2005. She is currently a PhD student at Polytechnic University of Catalonia, Spain, working full-time with the Internet Systems and Networking Group at Telefonica Research Lab in Barcelona, Spain. Her research interests include network coding and wireless systems.

João Barros is an Associate Professor at the Department of Electrical and Computer Engineering of the University of Porto and the coordinator of the Porto Laboratory of the Instituto de Telecomunicações. In February 2009, Dr. Barros was appointed National Director of the CMU-Portugal Program, a five-year international partnership between Carnegie Mellon University and 12 Portuguese Universities and Research Institutions, with a total budget of 56M Euros. He received his undergraduate education in Electrical and Computer Engineering from the Universidade do Porto (UP), Portugal and Universitaet Karlsruhe, Germany, until 1999, and the Ph.D. degree in Electrical Engineering and Information Technology from the Technische Universitaet Muenchen (TUM), Germany, in 2004. From 2005 to 2008, João Barros was an assistant professor at the Department of Computer Science of the University of Porto. The focus of his research lies in the general areas of information theory, communication networks and data security. Dr. Barros received a Best Teaching Award from the Bavarian State Ministry of Sciences, Research and the Arts, as well as scholarships from several institutions, including the Fulbright Commission and the Luso-American Foundation. He held visiting positions at Cornell University and the Massachusetts Institute of Technology, where he spent a sabbatical in 2008. Beyond his duties as Secretary of the Board of Governors of the IEEE Information Theory Society, his service included co-chairing the 2008 IEEE Information Theory Workshop in Porto, Portugal, and participating in several Technical Program Committees, including ITW 2009, WiOpt (2008 and 2009), ISIT 2007, IS 2007, and IEEE Globecom (2007 and 2008).

Muriel Médard is a Professor in the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology. She was previously an Assistant Professor in the Electrical and Computer Engineering Department and a member of the Coordinated Science Laboratory at the University of Illinois at Urbana-Champaign. From 1995 to 1998, she was a Staff Member at MIT Lincoln Laboratory in the Optical Communications and the Advanced Networking Groups. Professor Médard received B.S. degrees in EECS and in Mathematics in 1989, a B.S. degree in Humanities in 1990, a M.S. degree in EE in 1991, and a Sc. D. degree in EE in 1995, all from the Massachusetts Institute of Technology (MIT), Cambridge. She serves as an Associate Editor for the Optical Communications and Networking Series of the IEEE Journal on Selected Areas in Communications, as an Associate Editor in Communications for the IEEE Transactions on Information Theory and as a Guest Editor for the Joint special issue of the IEEE Transactions on Information Theory and the IEEE/ACM Transactions on Networking on Networking and Information Theory. She has served as a Guest Editor for the IEEE Journal of Lightwave Technology and as an Associate Editor for the OSA Journal of Optical Networking.

Professor Médard's research interests are in the areas of network coding and reliable communications, particularly for optical and wireless networks. She was awarded the IEEE Leon K. Kirchmayer Prize Paper Award 2002 for her paper, "The Effect Upon Channel Capacity in Wireless Communications of Perfect and Imperfect Knowledge of the Channel," IEEE Transactions on Information Theory, Volume 46 Issue 3, May 2000, Pages: 935–946. She was co-awarded the Best Paper Award for G. Weichenberg, V. Chan, M. Médard, "Reliable Architectures for Networks Under Stress", Fourth International Workshop on the Design of Reliable Communication Networks (DRCN 2003), October 2003, Banff, Alberta, Canada. She received a NSF Career Award in 2001 and was a co-winner of the 2004 Harold E. Edgerton Faculty Achievement Award, established in 1982 to honor junior faculty members "for distinction in research, teaching and service to the MIT community." She was named a 2007 Gilbreth Lecturer by the National Academy of Engineering. Professor Médard is a Fellow of IEEE.

Alberto Lopez Toledo is a researcher in the Internet Systems and Networking Group at the Telefonica Research Lab in Barcelona, Spain. He also serves as an Adjunct Professor in the Department of Communication and Information Technologies at Universitat Pompeu Fabra (UPF). Previously he was a researcher at the Telematics Engineering Department at the Universidad Politecnica de Madrid (UPM). Alberto received the M.S. degree in Computer Science (with highest honors) from the University of Murcia (UMU), Spain, in 1999 and the M. Sc. and the Ph.D. degrees in Electrical Engineering from Columbia University in 2002 and 2007 respectively.

Alberto's research interests are in the area of wireless systems and cross-layer design. Alberto received the Spanish National Academic Excellence Award, the Edwin Howard Armstrong Memorial Award, and the La Caixa Foundation and Rafael del Pino Foundation fellowships. He is currently a Institució Catalana de Recerca i Estudis Avançats (ICREA) fellow.