Towards Secure Multiresolution Network Coding

Luísa Lima João Barros Muriel Médard Alberto 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 streaming multimedia, in particular when only a subset of the users in the network is 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 consider schemes which exploit the algebraic structure of network coding to achieve the dual goal of hierarchical fidelity levels and efficient security. The key idea is to limit the encryption operations to the encoding vector, in combination with multi-resolution multimedia coding.**

***Index Terms*— network coding, streaming, multi-resolution, security, wireless, multimedia**

I. Introduction

Although much has been accomplished in terms of ensuring quality of experience for wireless multimedia streaming, it is not yet completely clear how heterogeneous users with different subscriptions can be provided simultaneously with a secure stream and differentiated levels of quality in a robust way. To ensure graceful degradation in the presence of packet losses and differentiated service provision to distinct users, typical video codecs, such as the MPEG family, adopt a multi-resolution source coding approach to generate a scalable video stream with multiple layers. The quality of experience for a user basically depends on the number of layers it is capable of recovering [1]. Unequal error protection for scalable video streaming is discussed for instance in [2], where fountain codes are applied to different layers thus ensuring graceful degradation over a wide range of packet loss probability.

Coding techniques also play an increasingly important role in the networking modules of the system architecture, in particular with the advent of network coding. The key idea of network coding [3] 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 ([4], [5]). Random Linear Network Coding (RLNC) can be implemented in a distributed fashion, whereby nodes draw several coefficients at random and use them to form linear combinations of incoming packets [6]. The resulting packet is sent along with the global encoding vector [6], which records the cumulative effect of the linear transformations the original packet suffers while on its path from the source to the destination. The global encoding vector is what enables the receivers to decode by means of Gaussian elimination. The benefits of network coding for wireless communications have been uncovered in several recent contributions [7], [8], [9].

Since network coding increases the overall computational complexity and introduces some decoding delay, it is fair to question its suitability for wireless media, in particular for video. Recent solutions for minimizing the decoding delay by means of feedback [10] and for applying network coding to the requirements of multimedia streaming [11], [12] offer encouraging evidence that the benefits justify the costs. In [13], opportunistic network codes are constructed in a way that maximizes the video quality instead of the network throughput. Reference [14] addresses multi-rate media streaming with network coding by formulating the problem in terms of linear programming with the goal of achieving a rate equal to the min-cut max-flow bound of each receiver. A survey of several techniques to exploit path diversity for media streaming, both with and without network coding, is provided in [12].

Network coding also offers advantages from a security point of view. One example is SPOC (S̲ecure P̲ractical Netwo̲rk C̲oding) [15], a lightweight cryptographic scheme that dramatically reduces the overall computational complexity by encrypting only the encoding vector (called the *locked coefficients*) and viewing the network code as a cipher in itself. Such a reduction of the number of encryption operations is deemed to be crucial for media transmission. In fact, as higher quality bit streams become available, the real-time decompression process can consume almost all the processing power and become overwhelming in conjunction with the resources required for the decryption of large files [16], [1].

The obvious potential in combining the security properties of network coding with its aforementioned benefits in robustness for streaming applications, motivates us to develop and analyze a 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 streaming video in a lossy wireless network. Our goals are (i) to reduce the number of encryption operations while meeting the prescribed security guarantees and (ii) to combine the resulting lightweight security schemes with efficient layered codes and streaming protocols for wireless video. Our main contributions are as follows:

- *Security Schemes:* We propose a set of security mechanisms designed for delay-sensitive applications that harvest the robustness of network coding with manageable complexity and without compromising security;
- *Scalable Network Coded Video:* We show how hierarchical codes for scalable video based on successive

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. A. L. Toledo (alopezt@tid.es) is 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.

978-1-4244-4536-3/09/$25.00 ©2009 IEEE

refinement can be combined with network coding in scenarios where not all the nodes are authorized to receive the best quality;

- *Performance Considerations*: We provide an analysis of the performance and the overhead of our schemes.

II. Preliminaries

Our goal is to provide confidentiality to video data while (1) keeping the number of encryption operations to a minimum without compromising security, (2) applying immediate decoding techniques for network coding thus complying with streaming delay requirements, (3) relying on RLNC's inherent robustness to packet loss and failures in the wireless network and (4) matching network coding with scalable video streams.

A. Network Assumptions

We consider a network abstraction where the source and intermediate nodes have access to the identifiers of the sinks (e.g. the IP addresses). This way, our schemes can be easily adapted to the many classes of networks that share this characteristic, in particular networks with no centralized knowledge of the network topology or of the encoding functions. Our scheme also requires an encryption mechanism for which the ciphertext is of the same size of the plaintext (e.g. AES in stream cipher mode).

B. Threat Model

We consider the threat posed by an 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;
4) he can perform active eavesdropping attacks to carry out known-plaintext attacks.

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

C. *Secure Practical Network Coding (SPOC)*

The basis of the schemes presented is SPOC (Secure Practical Network Coding) [15]. 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 intermediate nodes to run their network coding operations on substitute "unlocked" coefficients which provably do not compromise the hidden data. A security evaluation of SPOC [17] shows that the payload of the packets is in fact protected with information-theoretic security, in the sense that without the encoding matrix the attacker is unable to perform statistical attacks on the payload, predicated on efficient source coding.

III. Secure Network Coding for Video Streaming

A. *Vanilla scheme*

The Vanilla scheme is the basis for our work and considers the encryption of coefficients at the source, while providing a mapping that allows for immediate decoding strategies, described as follows.

1) Immediate decoding strategies: In traditional network coding protocols, the sinks must wait until they receive a full-rank matrix to decode. Since streaming applications have real-time requirements, large delays can cause severe penalties in the quality of the streaming data. However, when the knowledge of the content of the buffers of neighboring nodes is available, an intermediate node can choose the packets in the buffer so as to perform a linear combination that optimizes the intermediate and overall decoding delay. If the global encoding vector is encrypted, intermediate nodes cannot infer the packets that are combined in a given packet. The only approach for using this type of feedback is to obtain a unique mapping between the unlocked and locked coefficients that does not compromise security. This can be achieved by generating locked coefficients in the configuration of a lower diagonal matrix, as shown in *Figure 1*. A non-zero unlocked coefficient in column i corresponds to the combination of packets $i \dots 1$ inside the corresponding packet.

$$\overset{🔓}{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}} \rightarrow \overset{🔒}{\begin{pmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{pmatrix}}$$

Fig. 1. Mapping between unlocked and locked coefficients. If an intermediate node receives a packet with a vector of unlocked coefficients such that all positions are zero except for position 2, the packets that are combined are packets p_1 and p_2, while if an intermediate node receives a packet with a vector such that all positions are zero except for position 3, the packets that are combined are p_1, p_2 and p_3.

2) Precoding strategy: The security of protecting the encoding coefficients can be fine-tuned by pre-encoding at the source: in fact, if the encoding matrix is uniformly distributed among all elements of a finite field, while the plaintext is uniformly distributed among non-zero elements of the same field, the mutual information [18] between the payload and the plaintext is null and the mutual information between the payload and the encoding matrix is strictly greater than zero (that is, statistical attacks are possible on the encoding matrix but not on the payload) [17]. On the other hand, by using non-zero coefficients and zeros in the plaintext, we obtain dual results for the mutual information.

In this case, since we require the encoding matrix to be diagonal, all positions below (or above) the diagonal must be non-zero, which imposes an a-priori restriction on the encoding matrix. This restriction implies pre-coding the plaintext such that all symbols are non-zero and such that dependencies can be mitigated. This technique can be easily illustrated in the following extreme example. Suppose that the plaintext in one use of the encoding matrix is entirely composed of zero symbols. The output of the multiplication of a matrix composed exclusively of non-zero symbols by a zero vector is clearly the zero vector, and thus, an attacker could directly infer that the plaintext is the zero vector, since this is the most likely option yielding this result. Precoding the plaintext can be easily achieved by mapping elements of $\mathbb{F}_q$ into $\mathbb{F}_{q-1}$, thus incurring a negligible rate penalty of $(q-1)/q$.

B. *Strawberry scheme*

The Strawberry scheme builds on top of the Vanilla scheme to prevent known-plaintext attacks, as well as to prevent statistical dependencies in the payload of the packets without the need to perform the mapping proposed in the last scheme.

In fact, the Vanilla scheme presented is prone to known-plaintext attacks. An attacker in possession of n^2 packets and the full plaintext corresponding to those packets is able to recover the encoding matrix by solving a system with n^2 equations and n^2 unknowns. This attack can be easily counteracted in SPOC by adding an extra key, which can be deemed as a map for performing a random mix or reordering of the information symbols. The attacker would then be forced to test a prohibitive number of combinations. However, streaming applications require packets to arrive in the original order, and hence this technique is unfeasible for this goal. We now propose a technique for counteracting known-plaintext attacks against the Vanilla scheme. The key idea is to encrypt one symbol for each use of the encoding matrix by using the pre-shared keys between the source and the sinks. In the example below, even though the attacker has access to b_2 and b_3, he cannot obtain $E(b_1)$ and thus cannot form a solvable system of equations for recovering a_{ij}, as shown in *Figure 2*. It is worthwhile to mention that this technique also mitigates the effect of statistical dependencies occurring across reuses of the encoding matrix.

$$\begin{cases} \boxed{a_{11}\ \mathbf{E}(\mathbf{b_1})} + 0 + 0 = \gamma_1 \\ \boxed{a_{21}\ \mathbf{E}(\mathbf{b_1})} + \boxed{a_{22}}\, b_2 + 0 = \gamma_2 \\ \boxed{a_{31}\ \mathbf{E}(\mathbf{b_1})} + \boxed{a_{32}}\, b_2 + \boxed{a_{33}}\, b_3 = \gamma_3 \end{cases}$$

Fig. 2. Prevention of known-plaintext attacks in Strawberry scheme by encrypting one symbol per use of the encoding matrix. The symbols in the dashed boxes represent what the attacker aims at recovering, while the others represent the symbols the attacker has access to.

C. Chocolate Sundae scheme (Security layers)

The goal of this scheme is to allow 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. The key idea is to encrypt each line of the encoding matrix using a different key, as illustrated by the example in *Figure 3*.

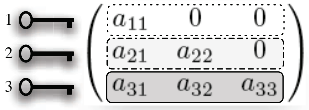

Fig. 3. Illustration of the encryption of the locked coefficients in the Chocolate scheme. The first layer corresponds to the first line of the matrix and is encrypted with the key for layer 1. The remaining locked coefficients are encrypted line by line according to a similar mechanism.

Unlocked coefficients allow for standard network coding operations to be performed on packets even though distinct levels are encrypted with different keys. Even if packets from different layers are combined, reverting the operations through the use of unlocked coefficients reverts all combinations of different layers, so that the original information can be recovered.

To prevent statistical dependencies, the technique of encrypting one of the symbols for each use of the matrix can be used as well; however, in this case, the symbol should be encrypted with the key for the lowest level in the network (that is, K_1), so that all legitimate intervenients in the protocol can decrypt the locked symbol, as shown in *Figure 4*. In this case, the base layer is also protected. If layer 1 is to be accessible by all nodes in the network, other symbols can be encrypted, as shown in *Figure 5*.

$$\begin{cases} \boxed{a_{11}\ \mathbf{E}(\mathbf{b_1},\mathbf{K_1})} + 0 + 0 = \gamma_1 \\ \boxed{a_{21}\ \mathbf{E}(\mathbf{b_1},\mathbf{K_1})} + \boxed{a_{22}}\, b_2 + 0 = \gamma_2 \\ \boxed{a_{31}\ \mathbf{E}(\mathbf{b_1},\mathbf{K_1})} + \boxed{a_{32}}\, b_2 + \boxed{a_{33}}\, b_3 = \gamma_3 \end{cases}$$

Fig. 4. Layered protection for the Chocolate scheme. In this example, all layers are protected from potential eavesdroppers.

$$\begin{cases} \boxed{a_{11}}\, b_1 + 0 + 0 = \gamma_1 \\ \boxed{a_{21}}\, b_1 + \boxed{a_{22}\mathbf{E}(\mathbf{b_2},\mathbf{K_2})} + 0 = \gamma_2 \\ \boxed{a_{31}}\, b_1 + \boxed{a_{32}\mathbf{E}(\mathbf{b_2},\mathbf{K_2})} + \boxed{a_{33}}\, b_3 = \gamma_3 \end{cases}$$

Fig. 5. Layered protection for the Chocolate scheme. In this example, the base layer of the video is accessible to the entire network.

D. Scheme comparison

A summary of the proposed schemes is shown in *Table I*. Operations that are common to all schemes are presented first, followed by the specifics for each method, in the order of execution. All schemes start with key distribution, a step that can be performed once, since keys can be reutilized. The source then generates the unlocked and locked coefficients. In the case of the Vanilla and Strawberry schemes, the source encrypts the encoding matrix using the shared group key; in the case of the Chocolate scheme, the source encrypts each line of matrix $\mathbf{A}$ using a different key. Pre-coding of the plaintext is performed according to the considered scheme. In the Vanilla scheme, the plaintext should not contain zeros (which can be easily achieved by using mapping) and should be compressed. In the Strawberry and Chocolate schemes, one symbol should be encrypted for each use of the encoding matrix. The packets are then composed by applying the matrix successively to the symbols in the information to be sent. At the intermediate nodes, all operations are performed according to the rules of standard RLNC protocols. In the case of the Chocolate scheme, packets are combined with packets of the same or lower layers. It is possible to obtain the layer of the packet by using the direct mapping to the unlocked coefficients. It is worthwhile to mention that any immediate decoding algorithms can be applied by using the same mapping.

At the sink nodes, Gaussian elimination according to the standard rules of RLNC protocols is applied using the unlocked coefficients. The locked coefficients can then be recovered (by using the single shared key in the case of the Vanilla and Chocolate schemes and multiple keys in the case of the Chocolate scheme), followed by the recovery of the sent information by means of forward substitution.

IV. Performance

A. Encryption volume

Figure 6 compares the volume of data to be encrypted according to the size of the plaintext for our schemes and traditional encryption, for typical packet sizes of 500 bytes (for video packets in cellular networks) [19], 1000 bytes (for example, for video over wifi networks) and 1500 bytes (the

TABLE I

SUMMARY OF PROPOSED SCHEMES

Initialization (source nodes):	
All	- A key management mechanism is used to exchange either 1 key or L shared keys with the sink nodes, depending on whether the scheme is for 1 or multiple layers, respectively. - The source node generates a $L \times L$ lower diagonal 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 $L \times L$ identity matrix are added to the header of each coded packet. These correspond to the *unlocked* coefficients.
Vanilla, Strawb.	- Each line of the matrix $\mathbf{A}$ is encrypted with the shared keys and also placed in the header of each packet. These coefficients correspond to the *locked* coefficients;
Vanilla	- The source node applies the matrix $\mathbf{A}$ to the packets $\{p_1, p_2, ..., p_N\}$ to be sent, forming packets $\{w_i = \text{join}(\mathbf{A}(b_{hi}, \ldots, b_{h(i+1)})^T \text{ for } i \in \{0, \ldots N-h\})\}$ and places them it in its memory.
Strawberry	- Each line of the matrix $\mathbf{A}$ is encrypted with the shared keys and also 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 $\{p_1, p_2, ..., p_N\}$ to be sent, forming packets $\{w_i = \text{join}(\mathbf{A}(b_{hi}, \ldots, \mathbf{E}(\mathbf{b_{h(i+1)}}, \mathbf{K})^T \text{ for } i \in \{0, \ldots N-h\})\}$ and places them it in its memory.
Chocolate	- Each line i of the matrix $\mathbf{A}$ is encrypted with shared key K_i and also 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 $\{p_1, p_2, ..., p_N\}$ to be sent, forming packets $\{w_i = \text{join}(\mathbf{A}(b_{hi}, \ldots, \mathbf{E}(\mathbf{b_{h(i+1)}}, \mathbf{K})^T \text{ for } i \in \{0, \ldots N-h\})\}$ and places them it in its memory.
Initialization (intermediate nodes):	
Chocolate	- Each node initializes L buffers, one for each layer in the network.
Operation at intermediate nodes:	
Vanilla and Strawberry	- When a packet is received by a node, the node stores the packet in its buffer; - To transmit a packet on an outgoing link, the node produces a packet by forming a random linear combination of the packets in its buffer, according to the rules of standard RLNC based protocols.
Chocolate	- 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):	
All	When *sufficient packets are received:* - The sink nodes perform Gaussian elimination using the unlocked coefficients thus obtaining the original locked coefficients and coded packets;
Vanilla, Strawb.	- The receiver then decrypts the locked coefficients using shared key K;
Chocolate	- The receiver then decrypts the locked coefficients using the corresponding keys K_i for level i;
All	- The receiver performs Gaussian elimination on the packets using the locked coefficients (triangular matrix) to recover the original packets;
Strawb, Choc.	- The receiver decrypts symbols $b_h, b_{2h}, \ldots, b_{N/h}$ to form the original plaintext.

typical IP packet size). In the case of 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 schemes substantially reduce 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. The Chocolate Sundae and Strawberry schemes require a higher number of encryption operations due to the extra operations (one symbol per use of the encoding matrix). 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 schemes as shown in *Figure 6*.

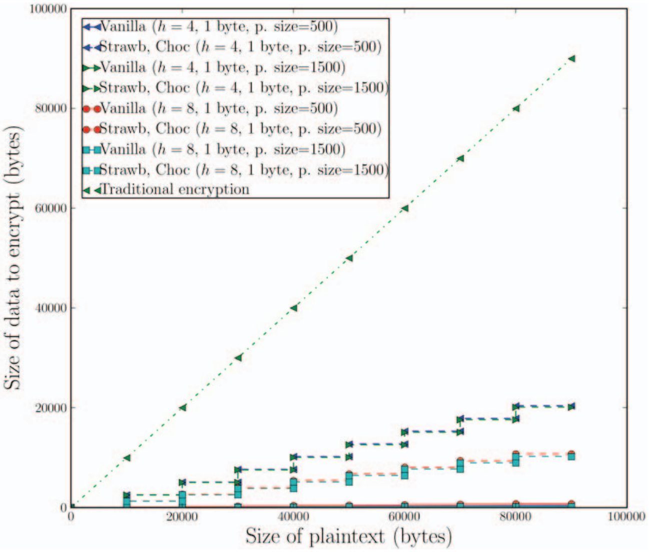

Fig. 6. Size of data to be encrypted, for Vanilla scheme and Strawberry/Chocolate schemes versus traditional encryption (encryption of the whole data).

B. Communication and Computational overhead

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 %

The ability to reduce the volume of data to be encrypted, for all the proposed schemes, comes at the cost of including locked coefficients in the data packet. In *Table II* we show the overhead introduced by our schemes 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. All schemes incur the additional communication overhead of key exchange, which is analyzed in *Section IV-C*.

Due to the inclusion of an extra set of coefficients (the locked coefficients), our schemes require 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 IV-A*	
Intermediate 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 IV-A*	$O(n^2)$
	Forward-substitution using recovered locked coefficients	$O(n^2)$	
	Decrypt one encrypted symbol per use of the encoding matrix (Strawberry and Chocolate)	See *Section IV-A*	

C. Key distribution

Our schemes require shared keys between the sinks and the sources. For the Vanilla and Strawberry schemes, one key can be shared between the source and all the valid recipients (commonly denominated as group keys), thus incurring a smaller overhead than the Chocolate Sundae scheme, in which several keys (one for each level) must be shared among all legitimate recipients in the network. Several mechanisms can be used for the exchange of shared keys, such as an offline mechanism for pre-distribution of keys [20], an authentication protocol such as Kerberos or a Public Key Infrastructure (PKI). The situation in which keys are shared among several legitimate nodes in a network occurs frequently in multicast scenarios and is commonly denominated as broadcast encryption or multicast key distribution [21]. We provide a generic overhead analysis based on the number of group keys that should be kept in the network. In the case of the Vanilla and Strawberry schemes, one key must be shared between the source and the t legitimate recipients, and thus, $t+1$ keys must be exchanged and are kept in the network. In the case of the Chocolate Sunday scheme, nodes at level l should keep l keys, one for each level, 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 at level l in the network and L the total number of levels in the network.

V. CONCLUSIONS AND FURTHER WORK

We presented a set of practical schemes for secure scalable media streaming that exploit the algebraic characteristics of RLNC. On the one hand our proposals ensure differentiated levels of security for distinct users. On the other hand, the multipath properties of the network coding paradigm assure the resilience to packet losses over wireless channels. Our work was focused on eavesdropping attacks, however network pollution attacks can be dealt with using the techniques in [22] 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. Finally, it is important to mention that these techniques can be generalized to other multi-resolution streaming architectures, including those based on the principles of fountain coding.

REFERENCES

[1] AS Tosun and W.C. Feng, “Efficient multi-layer coding and encryption of MPEG video streams,” *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, vol. 1, 2000.

[2] A.G. Dimakis, Jiajun Wang, and K. Ramchandran, “Unequal growth codes: Intermediate performance and unequal error protection for video streaming,” *MMSP 2007. IEEE 9th Workshop on Multimedia Signal Processing, 2007*, pp. 107–110, Oct. 2007.

[3] R. Ahlswede, N. Cai, S.Y.R. Li, and RW Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[4] A.G. Dimakis, P.B. Godfrey, M.J. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 2000–2008, May 2007.

[5] 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.

[6] 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.

[7] Christina Fragouli and Dina Katabi and Athina Markopoulou and Muriel Medard and Hariharan Rahul, “Wireless Network Coding: Opportunities & Challenges,” in *MILCOM 2007*, Orlando, FL, October 2007.

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

[9] 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.

[10] 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.

[11] J. MacLaren Walsh and S. Weber, “A concatenated network coding scheme for multimedia transmission,” *Fourth Workshop on Network Coding, Theory and Applications, 2008 (NetCod 2008)*, pp. 1–6, Jan. 2008.

[12] 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.

[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] 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.

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

[17] 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.

[18] T.M. Cover, J.A. Thomas, J. Wiley, and W. InterScience, *Elements of Information Theory*, Wiley-Interscience New York, 2006.

[19] TIA/EIA IS-707-A-2.10, *Data Service Options for Spread Spectrum Systems: Radio Link Protocol Type 3*, Jan. 2000.

[20] 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, Sept. 2008.

[21] 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.

[22] 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.