Secure Wireless Multicast for Delay-Sensitive Data via Network Coding

Tuan T. Tran, *Member, IEEE,* Hongxiang Li, *Senior Member, IEEE,* Guanying Ru, *Student Member, IEEE,* Robert J. Kerczewski, *Member, IEEE,* Lingjia Liu, *Member, IEEE,* and Samee U. Khan, *Member, IEEE*

***Abstract*—Wireless multicast for delay-sensitive data is challenging because of the heterogeneity effect where each receiver may experience different packet losses. Fortunately, network coding, a new advanced routing protocol, offers significant advantages over the traditional Automatic Repeat reQuest (ARQ) protocols in that it mitigates the need for retransmission and has the potential to approach the min-cut capacity. Network-coded multicast would be, however, vulnerable to false packet injection attacks, in which the adversary injects bogus packets to prevent receivers from correctly decoding the original data. Without a right defense in place, even a single bogus packet can completely change the decoding outcome. Existing solutions either incur high computation cost or cannot withstand high packet loss. In this paper, we propose a novel scheme to defend against false packet injection attacks on network-coded multicast for delay-sensitive data. Specifically, we propose an efficient authentication mechanism based on null space properties of coded packets, aiming to enable receivers to detect any bogus packets with high probability. We further design an adaptive scheduling algorithm based on the Markov Decision Processes (MDP) to maximize the number of authenticated packets received within a given time constraint. Both analytical and simulation results have been provided to demonstrate the efficacy and efficiency of our proposed scheme.**

***Index Terms*—Wireless multicast, network coding, Markov decision process, security, denied-of-service attack.**

I. Introduction

MULTICASTING delay sensitive data in wireless networks is of great interest in many applications. For example, in a cellular or WiFi network, multiple users may watch a National Basketball Association (NBA) live show or subscribe to the realtime stock market information updated through the same base station or access point. In these applications, a common requirement is that the information must be delivered to as many receivers as possible within certain time constraint.

In general, multicasting delay-sensitive data in a lossy environment is challenging. In particular, wireless channels are lossy in nature where packets may be lost during transmissions due to many reasons, such as radio frequency (RF) interference or channel fading. Additionally, at any given time point, each receiver may experience different packet losses. Traditional Automatic Repeat reQuest (ARQ) protocols are generally not suitable for lossy wireless multicast due to excessive retransmissions and high latency [1].

Network coding is a promising communication paradigm, and it has been proved that random network coding can approach the *min-cut* capacity of a multicast/broadcast session [2]. Simply put, using random network coding, the transmitter encodes a batch of original data packets and transmits random linear combinations to all the receivers. Any receiver can reconstruct the original data packets as long as it receives a sufficient number of linear independent coded packets. There is no need of retransmission; thus, it maximizes the number of receivers which successfully recover the original information within given time interval.

However, network coding is vulnerable to false packet injection attacks in a hostile environment. In particular, the adversary may inject bogus coded packets to prevent receivers from decoding the original data. Without any defense in place, the effect of this attack can be devastating. For example, because a set of coded packets is used to decode the original packets, e.g., using Gaussian elimination, even a single bogus coded packet can completely change the decoding outcome, rendering other genuine coded packets meaningless. The approach to let the transmitter digitally sign every packet would not resolve this issue as it incurs significant transmission bandwidth and computation overhead at the receivers. In addition, the adversary may also send many bogus packets or replay intercepted packets to force receivers into wasteful packet processing so as their limited energy and storage buffer are quickly deplete.

Notably, some schemes have been proposed to enable efficient stream authentication by amortizing one signature over multiple packets [3]–[6], which, however, cannot withstand high packet losses. In addition, Wang *et al.* [7] and Sun *et al.* [8] proposed group key management techniques

Manuscript received October 8, 2012; revised April 2, 2013; accepted May 28, 2013. The associate editor coordinating the review of this paper and approving it for publication was T. Hou.

This work was supported by US National Science Foundation (1156395 & 1228071) and Kentucky NASA EPSCoR 2012 Research Infrastructure Development Grant. This paper was presented in part at the IEEE International Conference on Communications (ICC), June, 2012.

T. T. Tran was with the Department of Electrical and Computer Engineering, J. B. Speed School of Engineering, University of Louisville when he performed this work. He is currently with InfoBeyond Technology LLC, Louisville, KY 40223 USA (e-mail: ttran10@asu.edu).

H. Li and G. Ru are with the Department of Electrical and Computer Engineering, J. B. Speed School of Engineering, University of Louisville, Louisville, KY 40292 USA (e-mail: {h.li, g0ru0001}@louisville.edu).

R. J. Kerczewski is with the NASA Glenn Research Center, Cleveland, OH 44135 USA (e-mail: RKerczewski@nasa.gov).

L. Liu is with the Department of Electrical Engineering and Computer Science, the University of Kansas, Lawrence, KS 66045 USA (e-mail: lingjialiu@ittc.ku.edu).

S. U. Khan is with the Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND 58102 USA (e-mail: Samee.Khan@ndsu.edu).

Digital Object Identifier 10.1109/TWC.2013.062413.121557

for cellular-like networks to reduce the communication cost. Further, [9] investigated the use of data embedding for key transmission by adding one more security layer. To access to the transmitted messages, adversaries must successfully break the key governing the data embedding rule. Using a different approach, Sarkar *et al.* [10] investigated a physical layer security scheme to protect the transmitted data over Rayleigh fading channels in wireless multicast. In another work, Canetti *et al.* [11] presented a taxonomy of secure multicast methods on the Internet from a practical viewpoint. Further, using a graphical approach, Wong *et al.* [12] proposed some strategies to manage the keys for group communications. Along in a different research line, packet authentication in network coded system has recently been investigated in the context of pollution attack. Most of the existing solutions, e.g., [13]–[15], are based on expensive homomorphic hash computation, which are vulnerable to denial-of-service (DoS) attacks.

In this paper, we study secure wireless multicast and devise a novel technique that enables efficient packet authentication at the receiver side, while maximizing the number of authenticated packets at each receiver within given time constraints. Specifically, our proposed technique has two main components. (1) Inspired by null space properties originally proposed to defend against pollution attack in wired-coded networks [16], the first component lets the transmitter generate and transmit a set of null key packets besides coded packets. On receiving the null key packets, each receiver can detect any bogus packet in the buffer with high probability. We further design an efficient authentication scheme to significantly reduce the computational cost incurred by null key packet authentication. (2) The second component aims at maximizing the authentication rate (defined later) at each receiver. Particularly, we design a novel efficient scheduling algorithm to adaptively transmit the null key packets based on the framework of MDP.

Our main contributions in this paper are summarized as follows:

1) We propose a novel scheme to enable efficient authentication of transmitted data at the receivers.
2) We show that without a careful transmission scheduling, the network performance would be detrimental.
3) We then propose an MDP-based scheduling technique to maximize the authentication rate across the receivers.
4) To leverage the time complexity of the MDP-based scheduling algorithm, we further propose a simulation-based scheduling to approximate the optimal scheduling.
5) The effectiveness of the proposed techniques is corroborated through both theoretical analysis and extensive simulations.

The organization of the paper is as follows. We first provide a review on secure data multicast and secure network coded systems in Section II. In Section III, we describe the system model, attack model, and security objectives. We then present our proposed authentication scheme in Section IV. Section V demonstrates the effects of transmission scheduling on the network authentication rate. We then investigate in detail a variety of transmission techniques and analyze their corresponding network recovery performance in Section VI. In Section VII, we formulate and solve the authentication scheduling problem by using the framework of the Markov Decision Processes. Simulation results and discussions are presented in Section VIII. Finally, a conclusion is drawn in Section IX.

II. Related Work

In this section, we review related work on secure data multicast and secure network coded systems.

Secure Multicasting. The problem of secure multicasting data over lossy channels has been investigated in many research papers such as [3],[5],[6],[11],[17]. In the work of [3],[5], the authors proposed efficient schemes to authenticate the information of finite and infinite streams by dividing the data into small blocks and utilizing the signature chain. Particularly, the transmitter embeds authentication information in the stream itself and signs the first block of the data. The signature verification is propagated to the rest of the stream through the embedded authentication information. However, the shortcoming of the proposed schemes is that they were not designed to defense against pollution attacks. In addition, the network performance will significantly decrease when the transmission channels are heterogeneous. Similarly, the work of [6] divided the transmitted information into small blocks and amortized a signature over many packets for verification. It showed substantial performance gain over the schemes of [3]. However, this scheme was originally designed for the Internet multicast; thus, its performance significantly drops when applying to heterogeneous wireless multicast over lossy channels. Alternatively, the work of [11] presented a taxonomy of multicasting on the Internet and proposed keying technique such that it allows source authentication and key revocation. The main focus of this work is efficient key management which reduces the data latency and overhead. Again, this technique cannot be applied to the data multicasting over lossy wireless channel. Notably, the work of [17] proposed an efficient time-based stream authentication technique over lossy channel, named TESLA. The proposed scheme can cope with the losses of transmissions. However, the main drawback of this technique is that it requires clock synchronization between transmitter and receivers, which is practically difficult to achieve in a wireless multicast.

Secure Network Coding. In a different avenue, several aspects of network security in the network coded systems have been studied such as [18]–[25]. In particular, Dong *et al.* have proposed solutions [26]–[28] to defense against pollution attacks in intra-flow network coding for wireless mesh networks. In these systems, the attackers inject corrupted packets into the network which will gradually propagate over the network, depleting the network resources and decreasing the network throughput. Ho *et al.* [23] derived an information theoretic approach to detect Byzantine adversaries in random network coding multicast networks. The analytical result showed an tradeoff among the detection probability, the data redundancy, the coding field size and the amount of information observed by the adversary. Similarly, Jaggi *et al.* [22] proposed algorithms which are information-theoretically secure and rate optimal for Byzantine adversary attacks. Performance of the proposed algorithms depends on the strength of the adversaries on observing the transmitted information.

In a different context, the work of [16],[18] showed approaches to defense against the malicious attacks in the peer-to-peer (P2P) based content distribution networks. The main idea of those approaches is to exploit the orthogonal properties of the span and null spaces generated by the coded packets to detect the attack assuming that the attackers can only observe portion of the transmitted data. Similarly, [29] also provided a TESLA-based scheme to defense against pollution attacks in P2P systems with network coding. However, this scheme required a time synchronization between the sender and receivers. In another work [30], the author proposed a technique combined between homomorphic and traditional hash function to defense against the pollution attacks in the Internet.

In the context of wireless inter-flow network coding, Newell *et al.* proposed Split Null Keys (SNK) scheme to protect transmitted data from pollution attacks. Using a different approach, Le *et al.* proposed a cooperative strategy to defense against pollution attacks in network coding using homomorphic MAC (SpaceMac) [31]. The proposed scheme requires a cooperation between the parents and children. When a centralized coordinator is available, [32] proposed a technique to locate the attackers. Additionally, [25] proposed a framework to defense against entropy attacks where an attacker creates and injects noninnovative packets into the network (i.e., packets that contain information already known by the system). The defense capabilities of the proposed schemes have also provided.

We also recognized that there are some excellent papers, e.g., [33],[34], on reviewing the threats and challenges in the network coded systems. Several security goals and design challenges in achieving security for network coding systems have been discussed. Those papers are viewed as a cautionary note pointing out the frailty of current network coding-based wireless systems and a general guideline in the effort of achieving security for network coding systems.

Different from all these techniques, we proposed a novel technique to defend against pollution attacks on network coded multicast for delay-sensitive data. The data transmissions in our model is constrained by a deadline beyond that the received information is useless. The investigated model can be used for many practical transmission scenarios such as multimedia streaming or time-sensitive applications. Furthermore, in our paper we assume transmission channels are heterogeneous, each receiver may experience different erasure packet rates. This creates significant challenges in scheduling data transmissions. Finally, we assume coordinated attacks in our model where the attackers coordinate with each other to pollute the received data of the receivers. We propose an efficient coding and adaptive scheduling algorithm to mitigate the bogus data at the receivers while maximizing the data transmission rate. Both theoretical analysis and intensive simulations are provided.

III. System, Attack Model, and Security Objectives

A. System Model

We consider a single-hop wireless network, where a transmitter securely multicasts delay-sensitive data to M users. We assume that the system is time-slotted and each slot length corresponds to one packet transmission. Furthermore, we assume that a block of $m-1$ data packets arrives at the transmitter periodically, and each block is associated with a deadline of T time slots, $T \geq m$. That is, to be useful at the receivers, a packet needs to be received and authenticated by the deadline. Additionally, we assume that the transmitter employs random network coding to generate coded packets over a large finite field $\mathbb{F}_q$. Further, we assume that the links between the transmitter and receivers are lossy, i.e., each packet transmitted to receiver R_j is subject to an erasure with probability p_j. The erasure probabilities are independent across the receivers and independent and identically distributed ($i.i.d$) across time slots. We assume that the network implements the distributed coordination function (DCF) [35] for medium access control (MAC). In our paper, all data processing functions are performed at the application layers.

B. Attack Model

We consider a computationally bounded adversary consisting of both *external* attackers (i.e., outsiders) and *internal* attackers (i.e., insiders). The model studied in our paper is coordinated attack where insiders cooperate with outsiders to launch the attack. Outsiders do not belong to the target network, but they are capable of overhearing packet transmissions, injecting bogus packets, and replaying intercepted packets. On the other hand, insiders are compromised yet undetected nodes which are fully controlled by the adversary. In fact, our attack model with attack contention probability ρ is analogous to the adversarial queuing model in [36], where adversary injects a rate of ρ into the network over a time window w, (w, ρ). As an example, to perform the attack, the insiders content, jam or block the legitimate transmitter while the outsiders (transmitter rogue), with the same IP and MAC addresses, spoofs transmitter's data to deliver bogus data to the receivers. Without any defense in place, the receivers may not be able to decode the received data. In addition, following the conventional assumption, we assume that the transmitter cannot be compromised and that non-compromised receivers are always the majority.

Among the many attacks the adversary can launch, this paper focuses on tackling the following two types of attacks on network coding based multicast.

- The attackers may inject bogus coded packets to pollute the receiver's buffer to prevent receiver from successfully decoding the original packets.
- The attackers may send many bogus packets or replay intercepted packets to force receivers into wasteful packet processing so as to quickly deplete their limited energy and/or memory buffer.

In addition, we assume that the transmitter has a public and private key pair K/K^{-1} and the public key K is publicly known.

C. Security Objectives

In the view of the two aforementioned attacks, our objectives are two folds:

- Packet integrity: Any bogus packet should be detected with high probability.

- DoS attack resilience: Packet authentication should be efficient.

IV. Packet Authentication

In this section, we present our proposed scheme that enables efficient packet authentication for multicasting delay-sensitive data. Particularly, our scheme is inspired by the null space originally proposed for defending against pollution attack in wired networks [16]. To enable efficient coded packet authentication, we let the transmitter generate a set of null key packets to be transmitted besides the coded packets. Upon receiving the null key packets, each receiver can authenticate all the coded packets it has received so far and detect bogus packets with high probability, if any. In addition, we propose an efficient scheme for null key packet authentication at the receivers. In what follows, we detail the design of our scheme, including *data packet preprocessing*, *null key packet generation*, and *packet authentication* at the receivers.

A. Data Packet Preprocessing

As standard, we assume that network coding is applied to a batch of $m-1$ incoming packets, denoted by $\{\mathcal{N}_i\}_{i=1}^{m-1}$. For secure transmission, each batch needs a signature packet, denoted as

$$\mathcal{S} = K^{-1}(h(\mathcal{N}_1||\cdots||\mathcal{N}_{m-1}))\ , \tag{1}$$

where $K^{-1}(\cdot)$ denotes the transmitter's signature using its private key, $h(\cdot)$ is a good hash function (e.g., Secure Hash Algorithm (SHA-2)), and $||$ denotes concatenation operator. For convenience, we subsequently treat the signature packet as a normal packet, i.e., $\mathcal{S} = \mathcal{N}_m$.

The transmitter interprets each packet $\mathcal{N}_i$ as an n-dimensional vector $(N_{i,1}, \cdots, N_{i,n})$ over the finite field $\mathbb{F}_q$ and appends a unit vector of length m to the vector $\mathcal{N}_i$ to create m augmented vector $\hat{\mathcal{N}}_i$ as

$$\hat{\mathcal{N}}_i = \langle \mathcal{N}_i, \underbrace{0, \cdots, 0, \overbrace{1}^{i}, 0, \cdots, 0}_{m} \rangle. \tag{2}$$

The jth coded packet is generated from the m original packets, in the form of

$$\begin{aligned} \mathcal{C}_j &= \sum_{i=1}^{m} \alpha_{j,i} \hat{\mathcal{N}}_i \\ &= \langle C_{j,1}, \cdots, C_{j,n}, \alpha_{j,1}, \cdots, \alpha_{j,m} \rangle\ , \end{aligned} \tag{3}$$

where $C_{j,i} = \sum_{i=1}^{m} \alpha_{j,i} N_{i,j}$; $\alpha_{j,i}$ and $N_{i,j}$ are coding coefficients and original data packets, respectively. The coefficients $\alpha_{j,i}$ are selected randomly from a sufficient large finite field $\mathbb{F}_q$ to ensure that all generated coded packets are independent. In order to allows the receivers to recover the original data, the coefficients are included within each coded packet. The graphical representation of the coding process is illustrated in Fig. 1. We note that using network coding, each coded packet will create some overhead compared with the original packet. However, the augmentation part has a size of $m \times$ symbol size (bits) (m is the generation size (order of ten), which is small, about 3% [37]), compared with the packet payload, order of Kbytes; therefore, it is negligible.

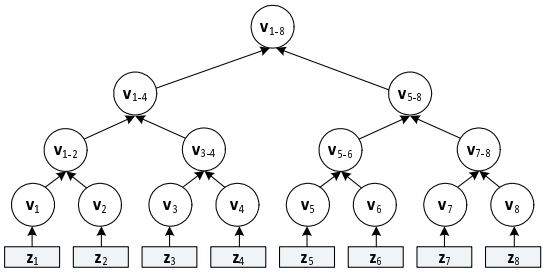


Fig. 2. An example of construction Merkle hash tree over $\{\mathbf{z}_1, \cdots, \mathbf{z}_n\}$, where $n = 8$.

B. Null Key Packet Generation

The adversary may inject bogus data packets to prevent receivers from successfully decoding the original packets. Recall that the transmitter signs the original packets, which can be verified after successful decoding. One challenge here is that if the receiver cannot differentiate genuine coded packets from bogus ones, it may be difficult to identify the sets of packets to correctly decode the original packets. For example, assume that a receiver has received m genuine and b bogus packets. To find the correct m packets for decoding, the receiver needs to examine possibly $\binom{m+b}{m}$ combinations, a number that grows exponentially in b.

To generate null key packets, we adopt the technique proposed in [16]. The transmitter then optimally sends the null key packets to the receivers to filter out bogus packets (detail of the optimal transmission scheduling is described in section VII.) The main idea of our bogus packet detection is based on the orthogonal property of the null space and subspace spanned by the original packets, i.e., any valid coded packet multiples with a null key packet will equal to zero. We next show how the null key packets are computed.

Let $\mathbf{N}$ denote the $m \times (m+n)$ matrix whose ith row is $\hat{\mathcal{N}}_i$. Thus, the subspace of matrix $\mathbf{N}$, i.e., $span(\mathbf{N})$, which is spanned by $\{\hat{\mathcal{N}}\}_{i=1}^{m}$, will contain all coded packet $\mathcal{C}_j$ (cf. equation (3)). In addition, the null space of $\mathbf{N}$, denoted as $null(\mathbf{N})$, which consists of the set of all vectors $\mathbf{z}$ for which $\mathbf{Nz} = 0$. Furthermore, let the dimension of $null(\mathbf{N})$ be $nullity(\mathbf{N})$, based on the rank-nullity theorem [38], we then have $rank(\mathbf{N}) + nullity(\mathbf{N}) = m+n$. Because $rank(\mathbf{N}) = m$, we obtain $nullity(\mathbf{N}) = n$. The basis vectors $\{\mathbf{z}_1, \cdots, \mathbf{z}_n\}$, which span $null(\mathbf{N})$, can be efficiently computed by using Singular Value Decomposition (SVD) [39]. The null key packets are any combination of the basis vectors $\{\mathbf{z}_i\}_{i=1}^{n}$, which can be used to verify the validity of the coded packets, i.e., $\mathcal{C}_j \mathbf{z}_i = 0$, for $\forall j, i$.

However, the adversary may also inject bogus null key packets. Without an appropriate authentication mechanism in place, genuine data packets may be misidentified as bogus ones. Further, null key packet authentication must be efficient, which may otherwise be exploited by the adversary to inject a large number of bogus packets to force the receivers to perform expensive signature verification.

We leverage a combination of Merkle hash tree [40] and digital signature to realize efficient authentication for null key packets. Specifically, let $n = 2^d$ for some integer d. To authenticate $\{\mathbf{z}_1, \cdots, \mathbf{z}_n\}$, the transmitter builds a Merkle

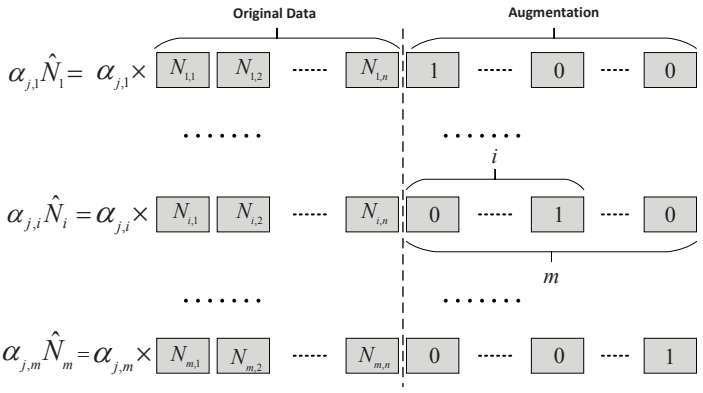


Fig. 1. Encoded packet is generated by linearly combining augmented vectors. Coding coefficients are included within the payload.

hash tree of depth d on top of $\{\mathbf{z}_1, \cdots, \mathbf{z}_n\}$, as illustrated in Fig. 2. In particular, the transmitter computes $\mathbf{v}_i = h(\mathbf{z}_i)$, for all $i \in [1, n]$, and builds a binary tree by computing each internal node as the hash of its adjacent children nodes. For example, $\mathbf{v}_{3-4} = h(\mathbf{v}_3||\mathbf{v}_4)$ and $\mathbf{v}_{1-4} = h(\mathbf{v}_{1-2}||\mathbf{v}_{3-4})$ in Fig. 2.

The transmitter signs the root of the Merkle hash tree, e.g., $\mathbf{v}_{1-8}$ using its private key K^{-1}. Given the Merkle hash tree, the base station constructs one null key packet for each $\mathbf{z}_i$, which consists of $\mathbf{z}_i$ itself and its authentication information, i.e., all the siblings of the nodes in the path from $\mathbf{v}_i$ to the root of the Merkle hash tree. For example, we have

$$\mathcal{K}_2 := \langle \mathbf{z}_2, \mathbf{v}_1, \mathbf{v}_{3-4}, \mathbf{v}_{5-8}, \mathbf{v}_{1-8}, K^{-1}(h(\mathbf{v}_{1-8}))\rangle$$

C. Packet Authentication at the Receiver

We defer the scheduling at the transmitter and introduce the following operation at the receivers in this subsection.

Upon receiving a data packet $\mathcal{C}_j$, the receiver marks it as an unverified packet and stores it in the receiver's buffer. When a new null key packet arrived, the receiver takes the following steps:

1) Step 1: check if a valid root of the Merkle hash tree of the current batch, e.g., $\mathbf{v}_{1-8}$, has been received before. If not, verify the signature $K^{-1}(h(\mathbf{v}_{1-8}))$ using the transmitter's public key K. Otherwise, check if it matches the one in new null key packet. The null packet is dropped if either $\mathbf{v}_{1-8}$ does not match or the signature is invalid .
2) Step 2: verify the authenticity of the received null key $\mathbf{z}_i$ using the authentication information along the path to the root of the Merkle hash tree. For example, for null key packet $\mathcal{K}_1$, the receiver checks if
$$\mathbf{v}_{1-8} = h(h(h(h(\mathbf{z}_1)||\mathbf{v}_2)||\mathbf{v}_{3-4})||\mathbf{v}_{5-8}) \ .$$
3) Step 3: using the received null key to check all the received data packets to detect bogus packets, if any.
4) Step 4: repeat Step 1 to Step 3 until it reaches the deadline. If the number of coded packets passed the authentication is larger or equal to m, decode the original data packets $\{\mathcal{N}_i\}_{i=1}^m$. If the number of authenticated coded packets is less than m, say m', then randomly choose a $m - m'$ packets from unverified packets to proceed to the decoding.
5) Step 5: verify if $\mathcal{N}_m$ is the valid signature of $\mathcal{N}_1||\cdots||\mathcal{N}_{m-1}$ (cf. Eq. (1)). If so, the batch is considered authenticated, and dropped otherwise.

Note that, the recovery of the transmitted data succeeded only if sufficient coded packets have been received and the receiver chooses a correct set of the coded packets for decoding.

D. Security Analysis

In what follows, we outline the security analysis of the proposed authentication scheme.

1) The integrity of null key packet: The transmitter signs the root of the Merkle hash tree using its private key K^{-1}, which is included in every null key packet. All the receivers know the public key of the transmitter, and thus can verify the signature. That said, all the receivers can authenticate the root of the Merkle hash tree, whereby to authenticate all the null keys. Therefore, any forged null key packet can be detected.

2) The integrity of coded packet: Each coded packet must pass the authentication using all received null key packets. Without loss of generality, assume that the transmitter have released $g < n$ null key packets, say $\{\mathbf{z}_i\}_{i=1}^g$. In the worst case, assume they have also been received by the adversary. The adversary then generates a bogus packet $\mathcal{B}$ such that $\mathcal{B}\mathbf{z}_i = 0$ for $\forall i \in [1, g]$. Because the adversary cannot predict $\mathbf{z}_{i+1}$, the probability that $\mathcal{B}$ is also orthogonal to the next null key $\mathbf{z}_{i+1}$ is $1/q$, where q is the underlying field size. In other words, any bogus coded packet can be detected with probability at least $1/q$ before being used to decode the original packets.

V. Time-delay Null Key Release

We observe that when the transmitter sends a null key packet to the receivers, it likely will be overheard and intercepted by the adversary who then can generate bogus data to pass the authentication. Therefore, to defend against the false injection attacks we need to carefully schedule transmissions. Finding an optimal scheduling boils down to how the transmitter chooses which packet to transmit at each time slot. Let us illustrate the intuition via the following example.

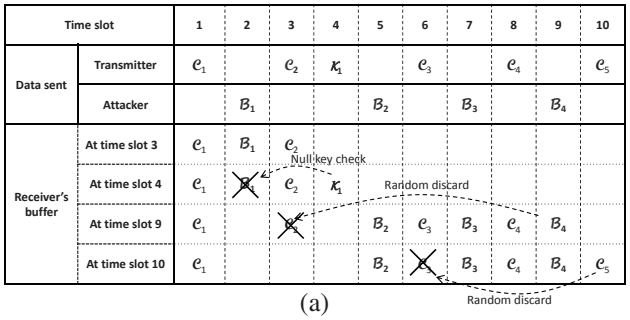
Example 1: Consider a single-source single-receiver wireless network with one attacker. The transmitter wishes to deliver a delay-sensitive data file consisting of $m = 4$ packets

within a deadline of $T = 10$ time slots. For now, let us assume that the receiver has small buffer size $|Q| = 6$ packets. When the receiver buffer is full and a new packet arrived, one of the received packets will be discarded uniformly at random. For the sake of exposition, we assume that there is no loss in transmissions. Further, at the beginning of each time slot, all users who want to send data over the network contend the transmission medium. Assume we have a realization in which the transmitter obtains the channel for transmissions at time slots $\{1, 3, 4, 6, 8, 10\}$, and the attacker obtains the channel for transmissions at time slots $\{2, 5, 7, 9\}$ as shown in Fig. 3. We let $\mathcal{C}_i$, $\mathcal{K}_i$, and $\mathcal{B}_i$ denote the coded packets, null keys, and bogus packets, respectively. We also assume that all packets have the same size and each packet is fitted into one time slot.

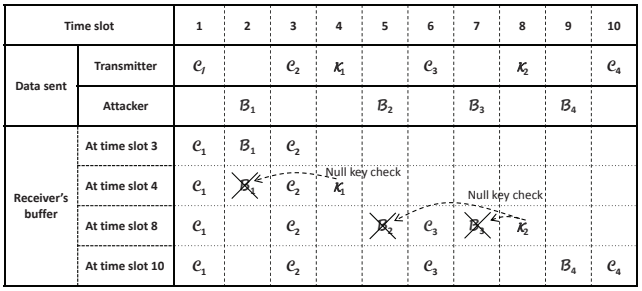
In the first scheme shown in Fig. 3(a), after time slot ts_4 the receiver obtains four packets $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{B}_3$, and $\mathcal{K}_1$. By using null key $\mathcal{K}_1$, the receiver is able to detect and discard the bogus packet $\mathcal{B}_1$. The receiver then also discards null key packet $\mathcal{K}_1$ to save space because with high probability the attacker might also receive $\mathcal{K}_1$, thus, it can generate next bogus packets that satisfy the checking conditions of $\mathcal{K}_1$. At time slot ts_9, the attacker obtains the transmission medium and sends another bogus packet $\mathcal{B}_4$. For illustration, we assume that the receiver implements random discard policy when the buffer is full. However, the problem is unchanged under any other queuing policies, e.g., random early discard, tail drop, etc., because of the random packet arrivals. There are more coded packets in the buffer; thus, with high chance one of them will be discarded, assumed $\mathcal{C}_2$. Hence, after nine time slots, the receiver has six packets $\{\mathcal{C}_1, \mathcal{B}_2, \mathcal{C}_3, \mathcal{B}_3, \mathcal{C}_4, \mathcal{B}_4\}$. Assume that at the last time slot ts_{10}, the transmitter sends another coded packet $\mathcal{C}_5$. Upon receiving it, the receiver also selects one of the received packets uniformly at random for discard. It may happen that another coded packet will be discarded, say $\mathcal{C}_3$. In the end, the receiver obtains only three coded packets and cannot recover the transmitted data.

We now show a better transmission strategy in Fig. 3(b). As shown, all the transmissions are scheduled the same way as before except for time slot ts_8. Instead of transmitting coded packet $\mathcal{C}_4$, the transmitter releases another null key packet $\mathcal{K}_2$. Upon receiving $\mathcal{K}_2$, the receiver uses it to detect and eliminate bogus packets $\mathcal{B}_2$ and $\mathcal{B}_3$. Using this transmission strategy, after ten time slots, the receiver obtains four coded and one bogus packets; thus, with high probability it is able to recover the transmitted data.

We observe that without careful scheduling, the network performance may be devastating. For example, sending too many null key packets will help the receiver filter out all bogus packets; however, it also decreases the number of time slots for coded packet transmission. On the other hand, sending only a few null key packets will leave many bogus packets in the receivers' buffer, making the decoding expensive. The network performance is maximized when the transmitter knows "when to send null key packets" and "how many null key packets will be sent". The framework of adaptive transmission scheduling is investigated in the next section.



(a)



(b)

Fig. 3. An example of two different time-delay null key releases for the case of one transmitter, one receiver, and one attacker with $m = 4$, $|Q| = 6$, and $T = 10$.

VI. Transmission Techniques: Performance Analysis

In this section, we will study network performance of some NC transmission techniques using different scheduling strategies. The network performance is measured by the recovery probability, i.e., the probability that all receivers in the network can recover (decode and authenticate) the transmitted data. Particularly, we focus on three techniques: network coding, network coding with randomly distributed null keys, and MDP-based adaptive null key scheduling. Due to the adaptive scheduling of the proposed MDP-based technique, there is no closed form to express its performance, and we describe how the transmitter adaptively selects a packet for transmission at every time slot in Section VII. For all techniques, we assume that the transmitter has m data packets consisting of $m - 1$ data and one signature packets that need to be delivered to M receivers in T time slots. Further, we assume that the erasure probabilities from the transmitter and attacker to receiver R_j are p_j and ϵ_j, respectively. For the sake of clarity, we assume that all the receivers have the same buffer size $|Q|$ and denote ρ to be the transmitter contention probability.

A. Network Coding (NC)

In this technique, the transmitter does not generate null keys and sends out only coded packets at every transmission opportunity. In order to recover the transmitted data, a receiver needs to obtain at least m coded packets[1] and choose a correct set of the coded packets among all the received ones. We have the following result.

Theorem 6.1: The recovery probability of a network consisting of M receivers is given by

[1]We assume that the operating finite field $\mathbb{F}_q$ is large so that all coded packets are independent.

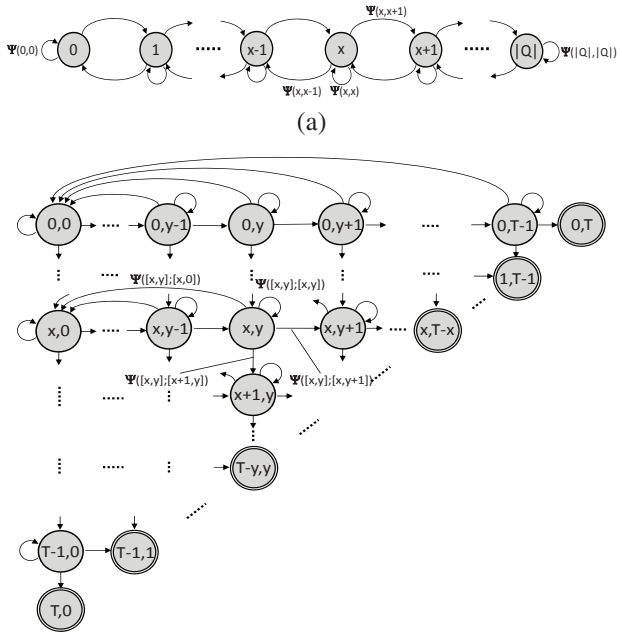


Fig. 4. Markov chain model for the revolution of the receiver's buffer: a) Network coding without null keys for $|Q| \leq T$; b) Network coding with randomly distributed null keys for $T \leq |Q|$. The final states are indicated with double circles.

$$\mathbb{P}_{\text{dec}} = \prod_{j=1}^{M} \mathbb{P}_{\text{dec}}(j), \tag{4}$$

where:

1) *Case 1*: $T \leq |Q|$

$$\mathbb{P}_{\text{dec}}(j) = \sum_{T_1=m}^{T} \sum_{l=m}^{T_1} \sum_{k=0}^{T_2} \frac{\binom{l}{m}}{\binom{l+k}{m}} \binom{T}{T_1} \rho^{T_1} (1-\rho)^{T_2} \times \binom{T_1}{l} (1-p_j)^l p_j^{T_1-l} \binom{T_2}{k} (1-\epsilon_j)^k \epsilon_j^{T_2-k}. \tag{5}$$

2) *Case 2*: $T > |Q|$

$$\mathbb{P}_{\text{dec}}(j) = \sum_{T_1=m}^{T} \sum_{l=m}^{\min\{T_1,|Q|-1\}} \sum_{k=0}^{|Q|-1-l} \frac{\binom{l}{m}}{\binom{l+k}{m}} \binom{T}{T_1} \rho^{T_1} (1-\rho)^{T_2} \times \binom{T_1}{l} (1-p_j)^l p_j^{T_1-l} \binom{T_2}{k} (1-\epsilon_j)^k \epsilon_j^{T_2-k} + \sum_{h=|Q|}^{T} \sum_{n=0}^{|Q|} \sum_{u=n}^{h} \frac{\binom{j}{m}}{\binom{|Q|}{m}} \binom{T}{u} \rho^u (1-\rho)^{T-u} \binom{u}{n} p_j^{u-n} (1-p_j)^n \times \binom{h-u}{|Q|-n} \epsilon_j^{h-u-|Q|+n} (1-\epsilon_j)^{|Q|-n} \Psi_j^{T-h}(n, \geq m), \tag{6}$$

where $\Psi_j^t(x,y)$ represents the transition probability from state x to state y in t steps and $T_2 = T - T_1$.

Proof: Consider receiver R_j and let $Z_c(t)$ and $Z_b(t)$ respectively be the number of coded and bogus packets received after t time slots. The theorem is proved in two cases.

1) *Case 1*: $T \leq |Q|$

In this case, there is no received packet discarded as $T \leq |Q|$. Let T_1 and T_2 respectively be the number of time slots the transmitter and attacker attained for transmissions; thus, $T_1 + T_2 = T$. Let $\mathbb{P}_{\text{dec}}(j)$ denote the probability that receiver R_j can recover the transmitted data. Because the transmitter does not use the null keys to filter out the bogus packets, the recovery probability is computed as the probability that receiver R_j correctly selects a set of m coded packets among all the received packets. We have that

$$\begin{aligned}\mathbb{P}_{\text{dec}}(j) &= \sum_{k=0}^{T_2} \frac{\binom{l}{m}}{\binom{l+k}{m}} \mathbf{Pr}\left(Z_c(T) = l \geq m, Z_b(T) = k\right) \\ &= \sum_{T_1=m}^{T} \sum_{k=0}^{T_2} \sum_{l=m}^{T_1} \frac{\binom{l}{m}}{\binom{l+k}{m}} \binom{T}{T_1} \rho^{T_1} (1-\rho)^{T_2} \\ &\times \binom{T_1}{l} (1-p_j)^l p_j^{T_1-l} \binom{T_2}{k} (1-\epsilon_j)^k \epsilon_j^{T_2-k}.\end{aligned} \tag{7}$$

The explanation is as follows. The probability $\mathbb{P}_{\text{dec}}(j)$ is obtained by adding all possible combinations of which R_j receives and selects a correct full set of coded packets out of the received packets (which may have bogus data packets). For example, the first term in equation (7) represents the probability that the receiver can pick a correct set of coded packets out of all received packets, i.e., $\binom{l}{m} \setminus \binom{l+k}{m}$, where l and k is the number of coded and bogus packets, respectively, and m is the number of coded packets requires for decoding. The second term indicates the probability that at least full set of coded data ($\geq m$) is received after T time slots, i.e., $P(Z_c(T) = l \geq m, Z_b(T) = k)$. This probability is further computed using the geometry distribution based on the contention probability ρ, data packet error rate p_j, and bogus packet error rate ϵ_j. The term $\binom{T}{T_1}\rho^{T_1}(1-\rho)^{T_2}$ computes the probability that the transmitter attain $T_1 \geq m$ time slots and attacker attains $T_2 = T - T_1$ time slots, out of total T time slots. The term $\binom{T_1}{l}(1-p_j)^l p_j^{T_1-l}$ accounts for the probability that at least a full set of coded packets is received, i.e., $l \geq m$. Finally, the last term represents the probability that the R_j receives k bogus packets, i.e., $\binom{T_2}{k}(1-\epsilon_j)^k \epsilon_j^{T_2-k}$. Combining all the terms we obtain equation (7).

2) *Case 2*: $|Q| < T$

Let the binary random variable $F = 1$ denote the full-buffer event and the random variable T^* denote the "*dropping threshold*", i.e., $T^* \triangleq \min\{t \in \mathbb{N}^+ : Z_c(t) + Z_b(t) = |Q|\}$, where $\mathbb{N}^+$ is the set of non-negative integer. We note that in this case some received packets might be dropped when $T^* < T$. Thus, the recovery probability of receiver R_j can be written as

$$\begin{aligned}\mathbb{P}_{\text{dec}}(j) &= \mathbf{Pr}_j(Z_c(T) \geq m, F = 0) \\ &+ \mathbf{Pr}_j(Z_c(T) \geq m, F = 1).\end{aligned} \tag{8}$$

The first term in equation (8) represents the case where the receiver is able to recover the transmitted data without filling the buffer during the transmission. This probability can be computed as

$$\mathbf{Pr}_j(Z_c(T)\geq m, F=0) = \sum_{T_1=m}^{T} \sum_{l=m}^{\min\{T_1,|Q|-1\}} \sum_{k=0}^{|Q|-1-l} \frac{\binom{l}{m}}{\binom{l+k}{m}} \times \binom{T}{T_1}\rho^{T_1}(1-\rho)^{T_2}\binom{T_1}{l}(1-p_j)^l p_j^{T_1-l} \times \binom{T_2}{k}(1-\epsilon_j)^k \epsilon_j^{T_2-k}. \quad (9)$$

The second term in equation (8) corresponds to the case where the receiver can recover the transmitted data with a full-buffer event occurred. The change of the received packets is more complicated than the previous case as it has random packet discarded. To characterize the change of the received data packets, we model it as the state transition of a Markov chain. Each state corresponds to a number of coded packets received. The state space is then written as $\mathcal{S} = \{0, 1, \ldots, |Q|\}$. The recovery probability with a full-buffer event is written as

$$\begin{aligned}&\mathbf{Pr}_j(Z_c(T)\geq m, F=1)\\ &=\sum_{l=0}^{|Q|}\sum_{k=|Q|}^{T}\mathbf{Pr}(Z_c(k)=l, Z_b(k)=|Q|-l)\\ &\times \Psi_j^{T-k}(Z_c(k)=l, i\geq m)\\ &=\sum_{l=0}^{|Q|}\sum_{k=|Q|}^{T}\sum_{u=l}^{k}\sum_{i\geq m}\frac{\binom{i}{m}}{\binom{|Q|}{m}}\binom{T}{u}\rho^u(1-\rho)^{T-u}\binom{u}{l}p_j^{u-l}(1-p_j)^l\\ &\times \binom{k-u}{|Q|-l}\epsilon_j^{k-u-|Q|+l}(1-\epsilon_j)^{|Q|-l}\Psi_j^{T-k}(Z_c(k)=l,i),\end{aligned} \quad (10)$$

where $\Psi_j^t(x,y)$ denotes the transition probability from state x to state y in t steps as illustrated in Fig. 4(a). In particular, for a state $0 < x < |Q|$, the valid transitions are only to states $x-1, x+1$ or itself x. For the extreme cases where the buffer is empty or full, states 0 and $|Q|$ can only transit to itself or states 1 and $|Q|-1$, respectively. The transition probabilities are specified as

$$\Psi_j(x,y) = \begin{cases}(1-\rho)(1-\epsilon_j)\frac{x}{|Q|+1}, & \text{if } y = x-1 > 0\\ \rho(1-p_j)\frac{x+1}{|Q|+1} + (1-\rho)\epsilon_j + \rho p_j & \\ +(1-\rho)(1-\epsilon_j)\frac{|Q|-x+1}{|Q|+1}, & \text{if } y = x\\ \rho(1-p_j)\frac{|Q|-x}{|Q|+1}, & \text{if } y = x+1 < |Q|\\ 0 & \text{otherwise.}\end{cases} \quad (11)$$

The first equation in (11) accounts for the case where the number of received coded packets decreases by one. This occurs only when the receiver correctly receives a bogus packet and randomly discards a coded packet. The second equation accounts for the case where the number of received coded packets is unchanged. It happens when new packet transmission is unsuccessful, or correctly received but the same type of the packet is discarded. The third case models the system after a successful coded packet transmission with a bogus packet discard afterward. That said, based on (11), we can write down the transition probability matrix $\mathbf{\Psi}$ and compute the transition probability from any state x to any state y in an arbitrary time steps.

The network recovery probability is computed based on (7), (9), (10), and noticing that the received data at the receivers is independent. We have that

$$\mathbb{P}_{dec} = \prod_{j=1}^{M} \mathbb{P}_{dec}(j). \quad \blacksquare$$

B. Network Coding with Randomly Distributed Null Keys (NC-RDNK)

In this technique, when the transmitter successfully attains the channel, it randomly selects a packet among the coded and null key packets for transmission. If the receiver correctly receives the transmitted null key packet, it then can filter out all bogus packets having been stored in its buffer with probability at least $1/q$ (cf. Section IV-D). For the sake of clarity, in our analysis we assume that the field size is large so that the probability that a null key fails to detect a bogus packet is negligible. Similarly, we model the change of the number of the received packets as the revolution of a Markov chain. In particular, we represent each network state by a vector $s = [n_c, n_b]$, where n_c and n_b respectively are the number of received coded and bogus packets. We have the following results.

Theorem 6.2: The recovery probability of a network consisting of M receivers is given by

$$\mathbb{P}_{\text{dec}} = \prod_{j=1}^{M} \mathbb{P}_{\text{dec}}(j), \quad (12)$$

where:

1) *Case 1*: $T \leq |Q|$

$$\mathbb{P}_{\text{dec}}(j) = \sum_{u\geq m}\frac{\binom{u}{m}}{\binom{u+v}{m}}\Psi_j^T([0,0];[u,v]), \quad (13)$$

2) *Case 2*: for $|Q| < T$,

$$\mathbb{P}_{\text{dec}}(j) = \sum_{u\geq m; u+v<|Q|}\frac{\binom{u}{m}}{\binom{u+v}{m}}\Psi_j^T([0,0];[u,v]) + \sum_{u\geq m}\sum_{k=|Q|}^{T}\sum_{x+y=|Q|}\frac{\binom{u}{m}}{\binom{u+v}{m}}\Psi_j^k([0,0];[x,y])\Phi_j^{T-k}([x,y];[u,v]), \quad (14)$$

and $\Psi_j^k([x,y];[u,v])$ and $\Phi_j^{T-k}([x,y];[u,v])$ are transition probabilities from state $[x,y]$ to state $[u,v]$ in k and $T-k$ steps, respectively.

Proof: The theorem is also proved in two cases corresponding to the order of the transmission window and the queue size.

1) *Case 1*: $T \leq |Q|$

As we notice that, in this case there is no received packet dropped as the the transmission window T is less than the queue size $|Q|$. The transition probabilities from non-final state $[x,y]$ to its neighboring state $[u,v]$ are illustrated in Fig. 4(b). Particularly, there are four valid transitions corresponding to the success or failure of the transmission of coded, null key, or bogus packet. We note that there are $T+1$ final states, indicated with double circles, representing the termination of the

network operation. The transition probabilities of a non-final state are specified as

$$\Psi_j([x,y];[u,v]) = \begin{cases} \frac{1}{2}\rho(1-p_j) & \text{if } u=x, v=0 \\ \rho p_j + (1-\rho)\epsilon_j & \text{if } u=x, v=y \\ \frac{1}{2}\rho(1-p_j) & \text{if } u=x+1, v=y \\ (1-\rho)(1-\epsilon_j) & \text{if } u=x, v=y+1 \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

The first equation accounts for the case where the transmitter attains the transmission channel and successfully delivers a null key packet to the receiver. The receiver then uses the received null key to filter out all the received bogus packets in its buffer, i.e., $v = 0$. The multiply factor $1/2$ indicates the fact that the transmitter uniformly randomly selects a packet among null key and coded packets for each transmission. The second equation accounts for a loop transition where there is no change in the number of received packets. This represents the scenario where either the transmitter or attacker fails to deliver their packets to the receiver. The last two equations correspond to the scenarios where the transmitter and attacker successfully deliver a coded and bogus packets to the receiver, respectively. Based on these observations, we can write down the transition probability matrix $\mathbf{\Psi}_j$ between two arbitrary states of receiver R_j. Therefore, the recovery probability of the transmitted data of receiver R_j after T time slots can be computed as

$$\mathbb{P}_{\text{dec}}(j) = \sum_{u \geq m} \frac{\binom{u}{m}}{\binom{u+v}{m}} \Psi_j^T\left([0,0];[u,v]\right). \tag{16}$$

2) *Case 2*: $|Q| < T$
In this case, some received packets might be discarded when the buffer is full and a new transmission arrives. We characterize the transition probability matrix $\mathbf{\Psi}_j$ of receiver R_j in two scenarios:

- When the start state $[x,y]$ with the total number of received packets less than the queue size, i.e., $x + y < |Q|$, the transition probabilities from state $[x,y]$ to its neighboring state $[u,v]$ are specified in (15).
- In the other case, the start state $[x,y]$ satisfies the condition $x + y = |Q|$, when a new packet arrives, one of the received packets will be discarded. The transition probabilities from state $[x,y]$ to its neighboring state $[u,v]$ are written as

$$\Phi_j([x,y];[u,v]) = \begin{cases} \frac{x}{2(|Q|+1)}\rho(1-p_j) & \text{if } u=x-1, v=0 \\ \frac{y}{2(|Q|+1)}\rho(1-p_j) & \text{if } u=x, v=0 \\ \rho p_j + \frac{x+1}{2(|Q|+1)}\rho(1-p_j) & \\ +(1-\rho)\epsilon_j + \frac{y+1}{|Q|+1} & \\ \times(1-\rho)(1-\epsilon_j) & \text{if } u=x, v=y \\ \frac{1}{2}\rho(1-p_j)\frac{y}{|Q|+1} & \text{if } u=x+1, v=y-1 \\ (1-\rho)(1-\epsilon_j) & \\ \times\frac{x}{|Q|+1} & \text{if } u=x-1, v=y+1 \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

In this formula, the first and second equations account for the fact that a new null key packet is received correctly, then, a coded and bogus packets are discarded at random, respectively. The null key is then used to filter out all the bogus packets in the buffer, i.e., $v = 0$. The third equation represents the case where the state loops to itself. This occurs when either the new transmission fails to reach to the receiver or the same type of packets is discarded after a new successful transmission. The last two equations correspond to two contradictory scenarios where a successful coded packet transmission with a bogus packet discard and a successful bogus packet transmission with a coded packet discard. Therefore, the data recovery probability of receiver R_j can be written as

$$\begin{aligned} \mathbb{P}_{\text{dec}}(j) = & \sum_{u \geq m; u+v<|Q|} \frac{\binom{u}{m}}{\binom{u+v}{m}} \Psi_j^T([0,0];[u,v]) \\ & + \sum_{u \geq m} \sum_{k=|Q|}^{T} \sum_{x+y=|Q|} \frac{\binom{u}{m}}{\binom{u+v}{m}} \Psi_j^k([0,0];[x,y]) \\ & \times \Phi_j^{T-k}([x,y];[u,v]), \end{aligned} \tag{18}$$

From (16) and (18), and using the fact that received data at the receivers is independent, we then can write the network recovery probability as

$$\mathbb{P}_{dec} = \prod_{j=1}^{M} \mathbb{P}_{dec}(j). \qquad \blacksquare$$

VII. The Proposed Adaptive Transmission Scheduling

As discussed, at every attained time slot, the transmitter can send either coded or null key packets to the receivers. Each transmitted packet additively affects to the receivers' authentication rate at the end of the transmission period, depending on whether it is able to be decoded. That said, without careful transmissions scheduling, the overall network performance can be detrimental. In this section, we investigate our proposed adaptive transmission scheduling that maximizes the network authentication rate.

A. MDP-based Scheduling

In this subsection, we show how the transmitter adaptively schedules transmissions (data packets and null keys), based on the network state, to maximize the network authentication rate. For now, we assume that at the beginning of each time slot the transmitter reliably receives one-bit feedback messages from the receivers to indicate whether the previous transmitted packet has been received successfully. The network dynamics is modeled as an MDP, and at each time slot, the transmitter chooses an action from the action set to maximize the accumulative reward at the end of the transmission period. In particular, we specify the network dynamics by a six-tuple $(\mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{r}, T, \gamma)$, where boldface letters refer to matrix or vector.

1) State space $\mathbf{S}$: A state s is defined by a matrix

$$s = \begin{bmatrix} c_1 & b_1 \\ c_2 & b_2 \\ \dots & \dots \\ c_M & b_M \end{bmatrix}, \tag{19}$$

where the first and second columns of the ith row $[c_i, b_i]$, respectively, represent the numbers of coded and bogus packets received by receiver R_i.
2) The action set $\mathbf{A}$ consists of i) sending a coded packet, ii) sending a null key, and iii) sending nothing (at time slots the transmitter cannot attain the channel).
3) The transition distribution $P(s_{t+1}|s_t, a_t)$ is computed based on the network current state, s_t, action taken, a_t, and the packet erasure probability of each receiver. For example, in the case of one receiver with coded and bogus packet erasure probabilities respectively are p and ϵ, and assume that the transmitter attains the channel, the probability $P(s_{t+1} = [00]|s_t = [00], a_t =$ "sending a coded packet") $= p$, or $P(s_{t+1} = [10]|s_t = [00], a_t =$ "sending a coded packet") $= 1 - p$, corresponding to the scenarios where the transmitted coded packet is received successfully or lost.
4) The immediate reward matrix $\mathbf{r}(s, a)$ is computed based on the future-dependent reward function, which is defined by

$$\mathbf{r}(s'|s, a) = \begin{cases} f(.) & \text{if } s' \text{ is a data-recoverable state} \\ 0 & \text{otherwise,} \end{cases} \tag{20}$$

where $f(.)$ is a non-negative function, and it is designed to reflect whether the receivers are able to recover the transmitted data and how easy the decoding process is performed. This is analogous to the delay reward assignment, i.e., the intermediate states, in which the receivers cannot recover the transmitted data, should have a zero-immediate reward. Detail of the design is illustrated via the example in Section VIII.
5) T is the number of stages or time slots associated with the current data batch.
6) Discount factor γ is in $[0, 1)$. When γ is close to zero, the transmitter tends to consider only immediate reward, and when γ is close to one, the transmitter prefers future reward with higher weight.

Once the parameters are specified, we use the backward induction algorithm (BIA), summarized in Algorithm 1, to find an optimal policy. The main idea of this algorithm is that it computes the immediate rewards and expected rewards backwardly from the final stage to the initial stage. The optimal policy is the one that results in the maximum expected reward.

Remark 1: We note that all the transition probabilities discussed above are conditional probability. Specifically, the transmission is performed only if the transmitter has attained the transmission medium. If we let a binary random variable $\Gamma = 1$ denote the event that the transmitter attains the transmission medium, the transition probability is computed as

$$\begin{aligned} P(s'|s, a) &= P(s', \Gamma = 1|s, a) + P(s', \Gamma = 0|s, a) \\ &= \rho P(s'|s, a, \Gamma = 1) \\ &\quad + (1 - \rho) P(s'|s, \text{"sending a bogus packet"}, \Gamma = 0), \end{aligned}$$

where ρ is the contention probability of the transmitter.

Algorithm 1 : Backward Induction Algorithm (BIA)

Input: $\mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{r}, T, \gamma$.
Output: $\pi^* = \{d(s_1), d(s_2), \dots, d(s_T)\}$

Initialize: $t = T$, set $V^*(s_T) = 0$ for all $s_T \in \mathbf{S}$
for $t = T - 1$ to 0 **do**
  $r(s_t, a_t) = \sum_{s_{t+1} \in \mathbf{S}} r(s_{t+1}|s_t, a_t) P(s_{t+1}|s_t, a_t)$
  $V'(s_t) = \gamma \sum_{s_{t+1} \in \mathbf{S}} P(s_{t+1}|s_t, a_t) V^*(s_{t+1})$
  $V^*(s_t) = \max_{a_t \in \mathbf{A}} \{r(s_t, a_t) + V'(s_t)\}$
  $d(s_t) = \arg\max_{a_t \in \mathbf{A}} \{V^*(s_t)\}$
end for

B. Approximation for MDP-ATS

As discussed above, when the network parameters such as the number of receivers, receivers' buffer size, and transmission window increase, the MDP-ATS technique is subject to the curse of dimensionality as state space size increases exponentially; its time complexity is given as $\mathcal{O}(\mathbf{T}|\mathbf{S}|^2|\mathbf{A}|)$. To tackle this problem, we use the simulation-based method [41] to approximate the optimal policy. In particular, our algorithm combines the simulation-based method with backward induction (SBIA) to reduce the time complexity to $\mathcal{O}(\mathbf{T}\Delta|\mathbf{S}||\mathbf{A}|)$ with $\Delta \ll \mathbf{S}$. Pseudocode of the SBIA algorithm is summarized in Algorithm 2.

Algorithm 2 : Simulation-based using Backward Induction Algorithm (SBIA)

Input: $\mathbf{S}, \mathbf{A}, \mathbf{T}, \gamma$.
Output: $\pi^* = \{d(s_1), d(s_2), \dots, d(s_T)\}$

Initialize: $t = T$, set $V^*(s_T) = 0$ for all $s_T \in \mathbf{S}$
for $t := T - 1$ to 1 **do**
  for each state $s_t \in \mathbf{S}$ **do**
    try all actions $a_t \in \mathbf{A}$ for Δ iterations, and compute
    $\widehat{V}^*(s_t, a_t) = \frac{1}{\Delta} \sum_{\Delta} [\mathbf{r}(s_{t+1}|s_t, a_t) + \gamma V^*(s_{t+1})]$
    $V^*(s_t) = \max_{a_t \in \mathbf{A}} \left\{\widehat{V}^*(s_t, a_t)\right\}$
    $d(s_t) = \arg\max_{a_t \in \mathbf{A}} \left\{\widehat{V}^*(s_t, a_t)\right\}$
  end for
end for

Remark 2: So far, we have solved the problem with perfect feedback from the receivers. However, in practice feedback messages are subject to losses and errors; thus, the transmitter observes only partial of the states. Fortunately, the problem can be formulated and solved as a Partially Observable Markov Decision Process (POMDP) and the POMDP problem can also be solved by using the same SBIA algorithm as

$$\widehat{V}^*(o_t, a_t) = \frac{1}{\Delta} \sum_{\Delta} [\mathbf{r}(o_{t+1}|o_t, a_t) + \gamma V^*(o_{t+1})], \tag{21}$$

where o_t is an observed state at time step t. This equation is similar to the formula in line 5 of Algorithm 2 (SBIA). However, in this case, the transmitter schedules its transmission (selects an action) based on its observed state, o_t, which may be different from the actual network state, s_t. The reward value is performed via the simulation-based approach where Δ is the number of iterations.

VIII. Simulation Results and Discussions

In this section, we will demonstrate the effectiveness of the proposed technique via simulations. In particular, we focus on comparing the network authentication rate of the following techniques: network coding (NC), network coding with randomly distributed null keys (NC-RDNK), network coding with greedy null key scheduling (NC-GNKS), MDP-based adaptive transmission scheduling (MDP-ATS), and the approximation for MDP-ATS technique (AMDP-ATS). We use the well-known TESLA scheme [17] for authentication of multicast stream over lossy channels as the benchmark in our comparison. In our simulation, TESLA is implemented with the key chain approach to tolerate packet losses during the transmission. In NC-GNKS technique, the transmitter sends out a null key packet only when existing a buffer overflow. This technique is of interest because it has low time complexity and may result in a good performance. We start with the basic setup.

A. Basic Setup

In our simulation, we assume that each delay-sensitive data batch consists of $m = 10$ layers that correspond to $m-1$ original data packets and one signature packet. The signature packet needs to be available at the receivers in order to authenticate the received data. This scenario is well suited to model the transmission of a Group of Picture (GOP) of the MPEG standard [42]. In addition, we assume that the erasure channels between the transmitter and receivers are independent. We consider two scenarios corresponding to the cases where the buffer size $|Q|$ is smaller or larger than the deadline T. For all simulations, we set the number of iterations for our simulation-based scheme $\Delta = 5$, discount factor $\gamma = 0.8$, and packet erasure probabilities $p_i = 10\%$ for $\forall i \in \{1, \ldots, M\}$ and $\epsilon = 20\%$. Different attack models are simulated by varying the value of ρ. In our simulation, the network authentication network rate is determined as the mean of the network authentication rates across all receivers defined in Eq. (22) over $10,000$ trials.

Definition 8.1: Assume that the transmitter has a batch of $m-1$ data packets $\{\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_{m-1}\}$ to be delivered to M receivers within a deadline of T time slots. A receiver R_j achieves an authentication rate of η_j if there exists a set of η_j packets which are decoded and authenticated by the deadline T. The average network authentication rate across all the receivers per unit time is defined as

$$\eta = \lim_{\tau \to \infty} \frac{1}{\tau} \sum_{\tau} \frac{\sum_{j=1}^{M} \eta_j}{MT}, \tag{22}$$

where τ is the operating time of the network.

We use the average network authentication rate as our performance measure to compare different techniques. A technique X is better than a technique Y if $\eta_X > \eta_Y$. We start the simulation by a simple example to demonstrate how the reward function is assigned in the MDP-ATS.

B. Reward Function: An Illustrative Example

We now demonstrate how the MDP-based scheduling works in a simple transmission scenario consisting of one transmitter, one receiver, and one attacker. Let us assume that the batch size $m = 2$, and the buffer size $|Q| = 3$. Since there is only one receiver in this example, a state can be represented by a vector, and there are total ten states as shown in Fig. 5. At each time slot, based on the feedback message from the receiver, the transmitter adaptively decides which packet will be transmitted.

To compute the transition probability matrix, let us denote p the packet erasure probabilities of the channels between the transmitter and receiver. In Fig. 5(a), we show the conditional transition probability associated with the action "sending a coded packet" given that the transmitter already attained the transmission channel. Each cell (i, j) in the matrix represents the probability of moving to state j from the current state i taking the action "sending a coded packet", $P(j|i, \text{"sending a coded packet"}, \Gamma = 1)$, where $\Gamma = 1$ denotes the event that the transmitter attains the transmission channel. For example, probability of transition from state $s_t = [10]$ to state $s_{t+1} = [20]$ is $1-p$, corresponding to the scenario where the receiver correctly obtains the transmitted packet. We recall that in our paper, we use uniformly random discard when the buffer overflows, for instance, the transition probability from state $s = [21]$ to state $s = [30]$ is $(1-p)/4$. This accounts for the case where the receiver receives the coded packet successfully, with probability $1-p$, and it randomly discards the bogus packet, with probability $1/4$ (three stored and one new arrived packets). Carrying out the same procedure, one can compute all the other transition probabilities associated with the action "sending a coded packet" as shown in Fig. 5(a).

We now compute the transition probabilities associated with the action "sending a null-key packet". Note that when the receiver successfully receives a null-key packet, it first stores the packet in its buffer, then uses it to filter out the bogus packets in the buffer. We emphasize that each null-key packet is authenticated separately, thus, the receiver can distinguish it from the other packets, i.e, coded and bogus packets. As a result, when the buffer overflows, the transmitter discards only either coded or bogus packets. For instance, the transition probability from state $s_t = [02]$ to state $s_{t+1} = [00]$ is $1-p$, corresponding to a scenario where the receiver receives the null-key packet successfully and uses it to filter out all the bogus packets. Similarly, one can compute the transition probabilities associated with the action "sending a null key packet" as shown in Fig. 5(b).

Fig. 5(c) shows a reward assignment that satisfies the conditions imposed in Eq. (20). In particular, if the receiver can decode and authenticate the transmitted data with probability one, then the transmitter obtains a reward of $r > 0$ units.

S_t \ S_{t+1}	[00]	[01]	[02]	[03]	[10]	[11]	[12]	[20]	[21]	[30]
[00]	p	0	0	0	1-p	0	0	0	0	0
[01]	0	p	0	0	0	1-p	0	0	0	0
[02]	0	0	p	0	0	0	1-p	0	0	0
[03]	0	0	(1-p)/4	p	0	0	3(1-p)/4	0	0	0
[10]	0	0	0	0	p	0	0	1-p	0	0
[11]	0	0	0	0	0	p	0	0	1-p	0
[12]	0	0	0	0	0	0	p+(1-p)/2	0	(1-p)/2	0
[20]	0	0	0	0	0	0	0	1	0	0
[21]	0	0	0	0	0	0	0	0	p+3(1-p)/4	(1-p)/4
[30]	0	0	0	0	0	0	0	0	0	1

(a)

S_t \ S_{t+1}	[00]	[01]	[02]	[03]	[10]	[11]	[12]	[20]	[21]	[30]
[00]	1	0	0	0	0	0	0	0	0	0
[01]	1-p	p	0	0	0	0	0	0	0	0
[02]	1-p	0	p	0	0	0	0	0	0	0
[03]	1-p	0	0	p	0	0	0	0	0	0
[10]	0	0	0	0	1	0	0	0	0	0
[11]	0	0	0	0	1-p	p	0	0	0	0
[12]	(1-p)/3	0	0	0	2(1-p)/3	0	p	0	0	0
[20]	0	0	0	0	0	0	0	1	0	0
[21]	0	0	0	0	2(1-p)/3	0	0	(1-p)/3	p	0
[30]	0	0	0	0	0	0	0	0	0	1

(b)

S_t \ S_{t+1}	[00]	[01]	[02]	[03]	[10]	[11]	[12]	[20]	[21]	[30]
[00]	0	0	0	0	0	0	0	0	0	0
[01]	0	0	0	0	0	0	0	0	0	0
[02]	0	0	0	0	0	0	0	0	0	0
[03]	0	0	0	0	0	0	0	0	0	0
[10]	0	0	0	0	0	0	0	**r**	0	0
[11]	0	0	0	0	0	0	0	0	**r/3**	0
[12]	0	0	0	0	0	0	0	0	**r/3**	0
[20]	0	0	0	0	0	0	0	**r**	0	0
[21]	0	0	0	0	0	0	0	0	**r/3**	**r**
[30]	0	0	0	0	0	0	0	0	0	**r**

(c)

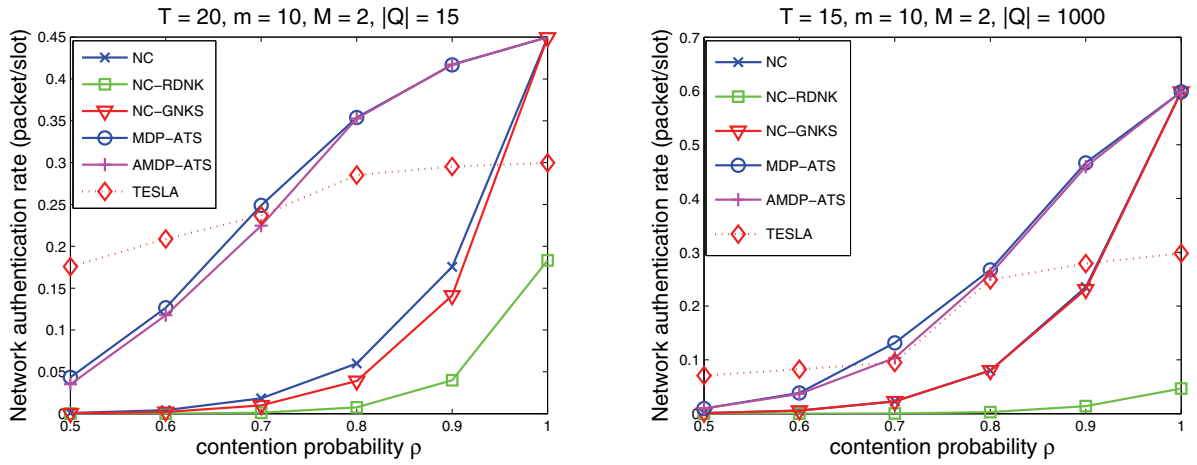
Fig. 5. Conditional transition probability associated with different actions: a) sending a coded packet, b) sending a null-key packet, and c) reward associated with the action of "sending a coded packet" $\mathbf{r}(s'|s,a)$.

On the other hand, with only some probability, the receiver can decode and authenticate the transmitted data, then the transmitter needs to pay a cost and achieves only a portion of the total reward $\zeta.r$ with $\zeta < 1$. In our simulation, the cost is defined by the probability that the transmitter is able to select a correct set of coded packets out of all received packets for decoding. For example, assume the current state is $s_t = [10]$, the transmitter has only one more time slot for transmission, and it decides to take the action "sending a coded packet". If the receiver correctly receives the transmitted packet, the next state will be $s_{t+1} = [20]$, and the transmitter obtains a reward of r units because with probability one, the receiver can decode and authenticate the transmitted data. On the contrary, if the receiver fails to receive the transmitted packet, thus the next state is the same as before, $s_{t+1} = [10]$, hence, it cannot decode and authenticate the transmitted data. The transmitter achieves a reward of zero unit. Another example, if the final state is $s_T = [21]$, then with probability $1/3$ the receiver can select a correct set of the received packets for decoding; thus, the reward achieved by taking this action is $r/3$.

C. Simulation Results

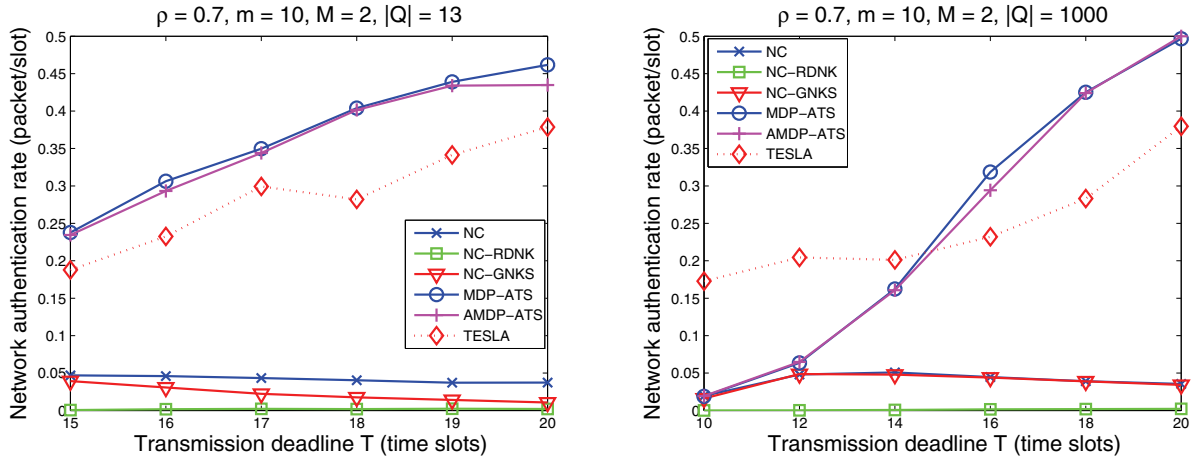
We first investigate the impact of contention probability ρ on the network performance of different techniques. We recall that the value of ρ indicates how severe the attack is to the network. Specifically, the lower value of ρ corresponds to an aggressive attack in which the attacker attains most of the time slots within the transmission deadline T. Figs. 6(a) and (b) show the network authentication rate with respect to ρ in the cases of small and large buffer sizes, respectively. We observe that NC-RDNK has the worst performance, whereas MDP-ATS achieves the best performance of all network coding-based schemes with a substantial gain in the regime of severe and mediate attack. Our explanation is that NC-RDNK randomly selects a packet for each transmission, thus, most of the time the receivers may not be able to collect enough data for the decoding process. On the other hand, MDP-ATS technique that optimizes the scheduling over all the transmission period T and adaptively selects a packet for each transmission in every time slot, consequentially, all the receivers are able to decode the data (i.e., most of the bogus packets are filtered out). In addition, in the case of no attack, i.e., $\rho = 1$, the two techniques MDP-ATS and NC achieve the same performance. Our intuition is that when there is no attacker, MDP-ATS technique will never transmit null key packets; thus, its transmission schedule is equivalent with that of NC. We further observe that the TESLA scheme outperforms all other schemes in the regime of smaller ρ. This is because TESLA uses separate authentication key attached to each data packet. Thus, it allows the receivers to verify a portion of the transmitted data when only partial of the data is received. On the contrary, when ρ increases, the transmitter attains more time slots for data transmission, network coding based schemes outperforms TESLA. This is because when more number of time slots available for data transmission, with high chance the receivers will obtain a whole set of coded packets, which allows the receivers to reconstruct and authenticate all data, resulting in a higher authentication throughput. As expected, the performance of TESLA scheme increases with respect to ρ. However, the increase is confined by the overhead of authentication key attached to each data packet. We observe the same network performance in the scenario of large buffer size as shown in Fig. 6(b). As expected, the two techniques NC and NC-GNKS have an identical performance. This is because when there is no buffer overflow, NC-GDNK technique transmits only coded packets, equivalent to NC. Furthermore, we observe that AMDP-ATS approximates the performance of MDP-ATS in both scenarios while requiring much lower time complexity. Furthermore, all schemes achieve higher performance compared with the small buffer size in Fig. 6(a) as there is no received data packet dropped.

Next, we study the impact of the transmission deadline T on the network performance. In this experiment we set the attack intensity as mediate, i.e., $\rho = 0.7$ and vary the transmission deadline T. Figs. 7(a) and (b) indicate the network performances of different techniques in the cases of small and large buffer sizes, respectively. Several insights are observed. First, as would be expected, NC-RDNK technique has the worst performance due to the lack of received data for the recovering process. Next, the MDP-based techniques, MDP-ATS and AMDP-ATS, achieve the best performances, substantially higher than that of the other techniques, and increase with T. Our intuition is as follows: when more number of time slots available for transmissions, most of the time the receivers are guaranteed to receive at least a full set of the coded packets and more null keys. Therefore, as T increases, the MDP-based techniques result in higher performances. We also observe that the performance of TESLA scheme increases



(a) (b)

Fig. 6. Network authentication rate per time slot vs. the contention probability ρ for a) small buffer size, b) large buffer size.



(a) (b)

Fig. 7. Network authentication rate per time slot vs. transmission deadline T for a) small buffer size, b) large buffer size.

with T, but it is lower than that of MDP-ATS and AMDP-ATS in Fig. 7(a). The reason of this is due to the combined effect of the overhead of the authentication key and the dropped data packet. On the contrary, the network authentication rates of NC and NC-GNKS techniques slightly decrease as T increases. The appealing explanation is that because the greater T implies that more opportunities the attacker can inject the bogus packets into the network, exaggerating the overflowed-dropping effect and preventing the decoding. The same network performance behavior is observed in the case of large buffer size as shown in Fig. 7(b). Particularly, when the number of time slots available for transmissions is equal to the number of coded packets, the four techniques, MDP-ATS, AMDP-ATS, NC, and NC-GNKS, are equivalent, that is, transmit coded packet at every time slot as shown in Fig. 7(b). TESLA offers the ability to allow receivers to authenticate each data packet separately, resulting in higher performance in the case of limited transmission time, i.e., $T \leq 14$. However, we emphasize that TESLA requires clock synchronization between the transmitter and receivers and in the simulation we assume perfect synchronization. However, in practice it is very difficult to achieve this, especially, in the multiple access wireless networks.

Figs. 8(a) and (b) describe the network performances of the scheduling techniques with respect to the erasure packet probability. Similarly, we set the contention probability $\rho = 0.7$ for both cases of large and small buffer sizes. Further, in this experiment we assume that all the receivers experience the same packet erasure probability p, and we vary p from 10% to 20% with a step of 2%. As expected, the network authentication rate of each technique decreases with the packet erasure probability. This is because there are more packet losses due to higher erasure probability. Again, NC-RDNK technique results in the worst performance. It is clear that the MDP-based techniques, MDP-ATS and AMDP-ATS, outperform all other techniques with a considerable gaps when the packet loss rate is less than 12% and 14% in the case of small and large buffer, respectively. Intuitively, the MDP-based techniques optimize their transmission at every time slot, depending on the network state. As a result, most of the time the receivers correctly obtain at least a full set of the coded packets, leading to higher network performance. In the high-erasure regime, TESLA outperforms all the other schemes thanks to its partial authentication capability. However, this gain comes with perfect synchronization between the transmitter and receivers by assumption in our simulation;

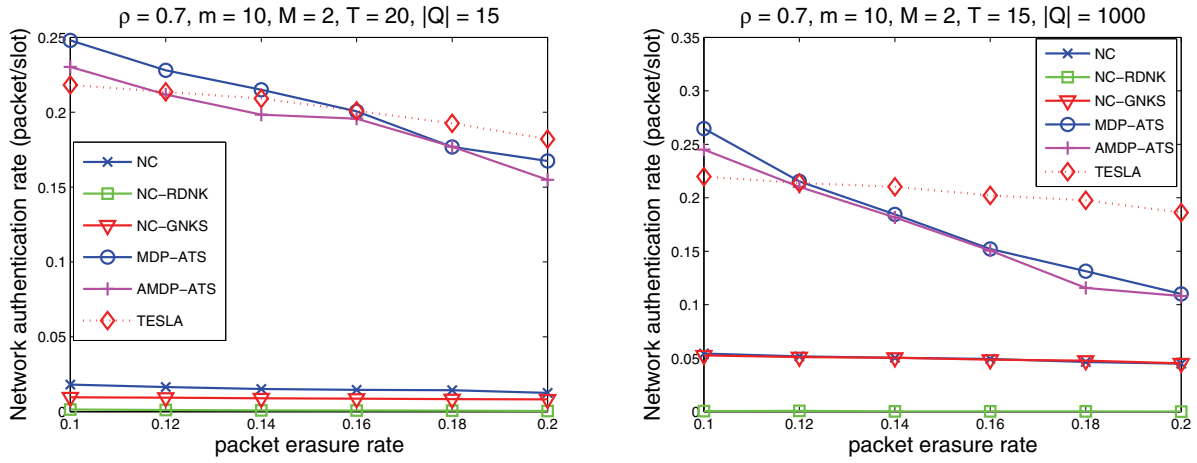


Fig. 8. Network authentication rate per time slot vs. packet erasure probability for a) small buffer size, b) large buffer size.

thus, the actual gain should be much less than that. In addition, we observe that AMDP-ATS technique closely approximates MDP-ATS in the whole range of packet erasure probability under consideration.

Finally, we examine the network authentication rate by varying the number of receivers in Figs. 9. We note that when the number of receivers increases, the state space size of the MDP-base schemes increases as well. Consequentially, it slows down the running time of the MDP-based techniques. When there is only a single receiver, TESLA achieves the best performance because of its partial authentication property. On the other hand, network coding-based schemes require receiving a full set for authentication which is hard to achieve under stressed time constraints. When the number of receivers increases, as expected, the MDP-based scheduling techniques outperform all the other schemes with a substantial gain. Particularly, the performance of TESLA scheme decreases steeply with the increase of receivers. The intuition of this is due to the effect of packet losses, i.e., the transmitter only sends new data packet when all receivers obtain the current transmitted data successfully. Differently, network coding based schemes send innovative information in each transmitted data packet, resulting in higher performance. Further, we observe that the network authentication rate of the AMDP-ATS technique declines slightly when number of receivers increases. The reason for this is that when increasing the number of receivers, the state space increases exponentially. As we keep the number of iterations Δ constant, some of the states will not be explored. Consequently, it results in a suboptimal policy which is slightly lower than that of the MDP-ATS scheme.

IX. Conclusion

In this paper we have studied the impact of the DoS attacks in a one-hop wireless network, where the transmitter wishes to multicast delay-sensitive data to multiple receivers over lossy channels. In particular, to defend against the false packet injection attacks, we proposed an efficient authentication mechanism with small overhead and light verification computation, which is resilient to data loss. One key idea in the authentication method is to exploit the null space properties of network coding combined with adaptive scheduling. To find an optimal transmission strategy, we casted the problem into the framework of the MDP. The BIA was then used to find an optimal solution for the scheduling problem. Further, to reduce the time complexity of the BIA method, we proposed a simulation-based algorithm to approximate the optimal solution. Analysis and simulations have been provided to demonstrate the effectiveness of the proposed techniques.

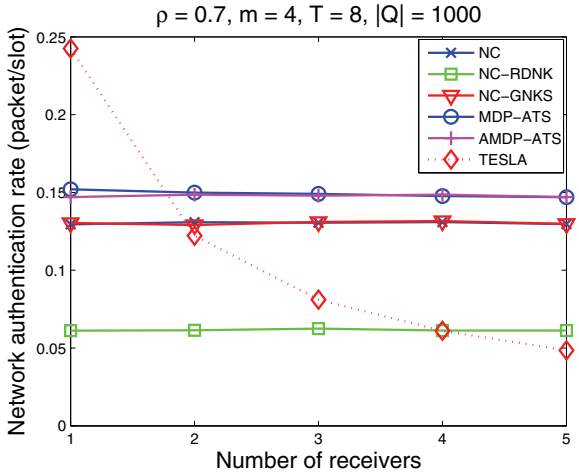


Fig. 9. Average authentication rate per time slot vs. number of receivers M.

Acknowledgment

The authors would like to thank the Associate Editor and anonymous reviewers for their thorough reading and constructive comments which improved the quality of the paper.

References

[1] A. Kumar, "Comparative performance analysis of versions of TCP in a local network with a lossy link," *IEEE/ACM Trans. Netw.*, vol. 6, pp. 485–498, Aug. 1998.

[2] T. Ho, M. Medard, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[3] R. Gennaro and P. Rohatgi, "How to sign digital streams," in *Proc. 1997 International Cryptology Conference on Advances in Cryptology*.

[4] J. Min, P. Edwin, K. Chong, and H. Siegel, "Efficient multicast packet authentication using signature amortization," in *Proc. 2002 IEEE Symposium on Security and Privacy*.

[5] P. Rohatgi, "A compact and fast hybrid signature scheme for multicast packet authentication," in *Proc. 1999 ACM Conference on Computer and Communications Security*.

[6] C. Wong, W. Simon, and S. Lam, "Digital signatures for flows and multicasts," in *IEEE/ACM Trans. Networking*, 1998.

[7] Y. Wang, P. Le, and B. Srinivasan, "Hybrid group key management scheme for secure wireless multicast," in *Proc. 2007 IEEE/ACIS International Conference on Computer and Information*, pp. 346–351.

[8] Y. Sun, W. Trappe, and K. Liu, "An efficient key management scheme for secure wireless multicast," in *Proc. 2002 IEEE International Conference on Communications*.

[9] W. Jie, J. Song, R. Poovendran, and K. J. Liu, "Key distribution for secure multimedia multicasts via data embedding," in *Proc. 2001 IEEE ICASSP*, pp. 1449–1452.

[10] M. Sarkar, T. Ratnarajah, and M. Sellathurai, "Secure wireless multicasting through Rayleigh fading channels—a secrecy tradeoff," in *Cognitive Wireless Systems (UKIWCWS)*, pp. 1–5, 2009.

[11] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast security: a taxonomy and some efficient constructions," in *Proc. 1999 IEEE INFOCOM*.

[12] C. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," *IEEE/ACM Trans. Netw.*, vol. 8, 2000.

[13] S. Agrawal and D. Boneh, "Homomorphic MACs: MAC-based integrity for network coding," in *Applied Cryptography and Network Security*, pp. 292–305, 2009.

[14] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a linear subspace: signature schemes for network coding," in *Proc. 2009 PKC*, vol. 5443, pp. 68–87.

[15] C. Gkantsidis and P. Rodriguez, "Cooperative security for network coding file distribution," in *Proc. 2006 IEEE INFOCOM*, pp. 1–13.

[16] E. Kehdi and B. Li, "Null keys: limiting malicious attacks via null space properties of network coding," in *Proc. 2009 IEEE INFOCOM*.

[17] A. Perrig, R. Canetti, J. Tyagar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," *2000 IEEE Symposium on Security and Privacy*.

[18] F. Zhao, T. Kalker, M. Medard, and K. Han, "Signatures for content distribution with network coding," in *Proc. 2007 International Symposium on Information Theory*.

[19] M. Krohn, M. Freedman, and D. Mazieres, "On the fly verification of rateless erasure codes for efficient content distribution," *2004 IEEE Symposium on Security and Privacy*.

[20] N. Cai and R. Yeung, "Secure network coding on a wiretap network," *IEEE Trans. Inf. Theory*, 2011.

[21] S. Jaggi, M. Langberg, R. Ho, and M. Effros, "Correction of adversarial erros in networks," in *Proc. 2005 International Symposium on Information Theory*.

[22] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Medard, "Resilient network coding in the presence of byzantine adversaries," in *Proc. 2007 IEEE INFOCOM*.

[23] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. Karger, "Byzantine modification detection in multicast networks with random network coding," *IEEE Trans. Inf. Theory*, 2008.

[24] A. Newell and C. Nita-Rotaru, "Split null keys: a null space based defense for pollution attacks in wireless network coding," *2012 IEEE SECON*.

[25] A. Newell, R. Curtmola, and C. Nita-Rotaru, "Entropy attacks and countermeasures in wireless network coding," in *Proc. 2012 ACM Conference on Security and Privacy in Wireless and Mobile Network*.

[26] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks," in *Proc. 2009 ACM Conference on Wireless Network Security*.

[27] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical defenses against pollution attacks in wireless network coding," *ACM Trans. Systems and Inf. Security*, 2011.

[28] J. Dong, R. Curtmola, C. Nita-Rotaru, and D. Yau, "Pollution attacks and defenses in wireless inter-flow network coding systems," *IEEE Trans. Dependable and Secure Computing*, 2012.

[29] A. Le and A. Markopoulou, "Tesla-based defense against pollution attacks in p2p systems with network coding," in *Proc. 2011 International Symposium on Network Coding*.

[30] A. Le and A. Markopoulou, "On detecting pollution attacks in inter-session network coding," in *Proc. 2012 IEEE INFOCOM*.

[31] A. Le and A. Markopoulou, "Cooperative defense against pollution attacks in network coding using spacemac," *IEEE J. Sel. Areas Commun.*, 2012.

[32] A. Le and A. Markopoulou, "Locating Byzantine attackers in intra-session network coding using spacemac," in *Proc. 2010 International Symposium on Network Coding*.

[33] J. Dong, R. Curtmola, R. Sethi, and C. Nita-Rotaru, "Toward secure network coding in wireless networks: threats and challenges," *Proc. 2008 Workshop on Secure Network Protocols*.

[34] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Secure network coding for wireless mesh networks: threats, challenges, and directions," *J. Computer Commun.*, 2009.

[35] "IEEE Standard for Wireless LAN Medium Acess Control (MAC) and Physical Layer (PHY) Specifications," Nov. 1997. P802.11.

[36] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson, "Adversarial queuing theory," in *Proc. 1996 ACM Symposium on Theory of Computing*.

[37] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 2003 Allerton Conf. Communication, Control and Computing Monticello*.

[38] H. Anton and C. Rorres, *Elementary Linear Algebra: Applications Version*. John Wiley & Sons, 2010.

[39] O. Alter, P. Brown, and D. Botstein, "Singular value decomposition for genome-wide expression data processing and modeling," in *Proc. 2000 National Academy of Science of the United States of America*.

[40] R. Merkle, "Protocols for public key cryptosystems," in *Proc. 1980 IEEE S&P*, pp. 122–134.

[41] H. Chang, M. Fu, J. Hu, and S. Marcus, *Simulation-Based Algorithms for Markov Decision Processes (Communications and Control Engineering)* Springer-Verlag, Inc., 2007.

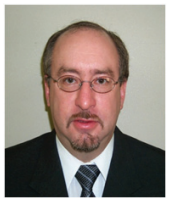
[42] V. Goebel and T. Plagemann, *Interactive Distributed Multimedia Systems and Telecommunication Services*. Springer, 1998.

Tuan T. Tran is currently a research scientist at InfoBeyond Technology LLC. He received his Ph.D. degree from Oregon State University, Corvallis in 2010 and his B.S. degree in Electronics and Telecommunications from Hanoi University of Technology (HUT), Vietnam, in 2000. Prior to joining InfoBeyond, he worked as postdocs at Arizona State University and the University of Louisville from 2010 to 2012. His research interests include network security, network and channel codings, wireless communications, and multimedia networking.

Hongxiang Li is currently an assistant professor with the Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY. Prior to that, he was an assistant professor at North Dakota State University from 2008 to 2011. He received a B.S. degree from Xi'an Jiaotong University, China in 2000, a M.S. degree from Ohio University in 2004, and a Ph.D. degree from University of Washington, Seattle in 2008, all in electrical engineering.

Dr. Li's general research area is wireless communications and networks. He has published over 70 journal and conference articles in the field. He won the Best Paper Award in the IEEE International Conference on Electro/Information Technology (EIT) in 2013. His research has been funded by National Science Foundation (NSF), National Aeronautics and Space Administration (NASA) and Air Force Research Lab (AFRL). In the year of 2008, he received the Chinese Government Award for Outstanding Self-Financed Students Abroad, the University of Washington Outstanding Research Assistant Award and was nominated for the YANG Research Award. He is also the recipient of the 2012 Ralph E. Powe Junior Faculty Enhancement Award.

Guanying Ru is currently pursuing her Ph.D. degree in the Department of Electrical and Computer Engineering at University of Louisville. She visited the University of California, Davis for six months in 2011. She did her Ph.D. studies at North Dakota State University from 2009-2011. She received the B.S. degree and the M.S. degree (both with honors) in telecommunication engineering from Zhengzhou University, China, in 2006 and 2009, respectively. Her current research interests include resource management and capacity analysis for next generation communication systems and heterogeneous networks.



Robert J Kerczewski received the MS degree in Electrical Engineering and Applied Physics from Case Western Reserve University (1987) and Bachelor of Electrical Engineering from Cleveland State University (1982). He has been working for NASA Glenn Research Center since 1986 in the field of space and aeronautical communications systems and applications. During that time he has served as Principle Investigator for interference and telemedicine experiments with the Advanced Communications Technology Satellite, and as Project Manager for the Advanced Communications for Air Traffic Management Project, Space Based Technologies Project, and the Integrated Vehicle Health Management Project. He is currently Deputy Project Manager for the Unmanned Aircraft Systems Integration in the National Airspace System Project, and actively engaged in aeronautical communications research activities for unmanned aerial vehicle command and control communications, air-ground data communications, airport surface wireless communications, and aviation spectrum.

Lingjia Liu received the Ph.D. degree at Texas A&M University in Electrical and Computer Engineering, the B.S. degree with highest honor at Shanghai Jiao Tong University in Electronic Engineering. He is currently working as an Assistant Professor in the Electrical Engineering and Computer Science Department at the University of Kansas (KU). Prior to joining the EECS at KU, he spent more than three years in Samsung Research America Dallas (SRA-D) leading Samsung's work on downlink multi-user MIMO, Coordinated multipoint (CoMP) transmission, and Heterogeneous Networks for 3GPP LTE/LTE-Advanced standards where he has more than 10 essential intellectual property rights (IPRs). His general research interests lie in the areas of wireless communication systems including cooperative communications, energy-efficient communications, multi-user MIMO systems, coordinated multipoint transmissions, and heterogeneous networks.

Lingjia Liu is a recipient of the Texas Telecommunications Engineering Consortium (TxTEC) Fellowship from the Department of Electrical and Computer Engineering at Texas A&M University in 2003 - 2004. He received the Global Samsung Best Paper Award in 2008 and 2010 respectively. He is the best paper finalist for the ICC 2012 Wireless Communication Symposium (5/508). He has also been selected by the National Engineers Week Foundation Diversity Council as New Faces of Engineering 2011 and was recognized during the 2011 National Asian American Engineers of The Year (AAEOY) Awards Banquet in Seattle 2012.

Lingjia Liu has been actively involved in beyond 4G technologies where he has been serving as the Chair of Technical Program Committee (TPC) of GLOBECOM International Workshop on Emerging Technologies for LTE-Advanced and Beyond-4G 2012 and 2013 (http://wcsp.eng.usf.edu/b4g/). He is the TPC co-Chair of the Communication Theory Symposium (CTS) of WCSP 2013 and TPC co-Chair of the Wireless Ad Hoc and Sensor Networks Symposium (WAHS) of ICNC 2014. Lingjia Liu is also serving as an Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and as Associate Editors for *EURASIP Journal on Wireless Communications and Networking* as well as Wiley's *International Journal on Communication Systems*.

Samee U. Khan received a B.S. degree from Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan, and a Ph.D. from the University of Texas, Arlington, TX, USA. Currently, he is Assistant Professor of Electrical and Computer Engineering at the North Dakota State University, Fargo, ND, USA. Prof. Khans research interests include cloud and big-data computing, social networking, and reliability. His work has appeared in over 200 publications. He is a Fellow of the Institution of Engineering and Technology (IET, formally IEE), and a Fellow of the British Computer Society (BCS).