INVITED PAPER

Wireless Network Security: Building on Erasures

In wireless settings, careful exploitation of packet loss, multipath propagation, and ACK/NACK feedback can lead to broadcast erasure channels, for which strong secrecy results are easier to establish using maximum distance separable codes. This paper shows how these results lead to secure messaging and secret-key agreement.

By Christina Fragouli, *Senior Member IEEE*, Vinod M. Prabhakaran, *Member IEEE*,
László Czap, *Student Member IEEE*, and Suhas N. Diggavi, *Fellow IEEE*

ABSTRACT | One of the most widely-known techniques for securing a message from eavesdropping, is the famous one-time pad. Although the one-time pad offers unconditional security, it has limited applicability today, because it requires that to communicate, two parties already share a key that is not known by the eavesdropper and has size equal to the message. In this review paper we present a line of work that explores how we can efficiently share keys securely from an eavesdropper, and thus communicate using a one-time pad like approach. The basic idea is to exploit new opportunities that wireless networks offer, such as the fact that we have multiple paths, the fact that we have packet losses, and the availability of ACK/NACK feedback.

KEYWORDS | Cryptography; erasure networks; information theoretic security; message security; network coding; oblivious transfer; wireless network security

I. INTRODUCTION

Securing a message from eavesdropping is a fundamental security problem. Today, we achieve security using cryptographic techniques, that build on the fact that the adversary has limited computational capabilities; for instance, cannot perform fast large integer factorization. In this review paper we describe a new, alternative method, that makes no assumptions on the computational power of the adversary. Instead, it exploits properties of networks to implement one-time pad encryption techniques that offer perfect secrecy.

The idea behind one-time pad encryption is simple. Alice wants to send a message M (a binary vector) to Bob, securely from a passive eavesdropper, Eve. Assume that Alice and Bob share a secret key K (a random binary vector, of length equal to the message), about which Eve has no information. Alice can encrypt M by performing bit by bit binary addition of M with the key K (i.e., `xor` the message M and E to create $M \oplus E$), and send the encrypted message to Bob. Bob, with both the key and the encrypted message at his disposal, can retrieve M. Eve, with no knowledge of K, cannot.

The technique has a long history, and can offer strong security, but has practical limitations. One-time pad encryption traces back to the nineteenth century, and was also reinvented during World War I. Quoting from [1]: "The 'pad' part of the name comes from early implementations where the key material was distributed as a pad of paper, so that the top sheet could be easily torn off and destroyed after use." Shannon proved in the 1940's that, provided the key has the same length as the message, and that it is used exactly once and then thrown away, the scheme offers perfect security [2].[1] That is, even if Eve had unlimited processing capabilities, her best estimate of the message would be just a random guess—observing the encrypted message does not help her. But then the question becomes, how do we efficient create the "pad," how

[1] Shannon also showed that without a secret key at least as long as the secret message, it is impossible to keep the message completely secret from an eavesdropper who can observe the cipher-text (and is not computationally limited).

Manuscript received February 1, 2015; revised May 8, 2015; accepted May 9, 2015. Date of publication August 19, 2015; date of current version September 16, 2015. This work was supported in part by the U.S. NSF under Grant 1321120.

C. Fragouli and **S. N. Diggavi** are with the Department of Electrical Engineering, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: christina.fragouli@epfl.ch; suhas@ee.ucla.edu).
V. M. Prabhakaran is with the School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai 400005, India (e-mail: vinodmp@tifr.res.in).
L. Czap is with IBM, Budapest 1117, Hungary (e-mail: laszlo.czap@epfl.ch).

Digital Object Identifier: 10.1109/JPROC.2015.2438312

do we create shared secret keys of size equal to the messages between Alice and Bob. This challenge may create the impression that this is a scheme of mainly theoretical interest, with limited applicability in practice.

We here review our recent line of work, that exploits wireless network properties to create "pads" efficiently.[2] We exploit the fact that in networks, Alice and Bob may be connected through multiple paths, and Eve may be limited to observe a subset of them; the fact that channels are noisy, leading to packet losses that are random; that in wireless, we have broadcasting and interference; and the existence of feedback, that is today part of all wireless standards. By linking these properties together, we show that we can create large shared secrets in polynomial time, which indicates that one-time pad encryption, implemented appropriately, may be a scheme not only of theoretical but also of practical interest.

This paper focuses on developing a theoretical understanding of these ideas. To do so, we model the wireless medium as a broadcast packet erasure channel, with some erasure probability towards each receiver. Erasure networks, although simpler than general networks, still capture the intricacies and possibilities of operating in a wireless network environment (broadcasting, multipath, channel variability, feedback). At the same time, analyzing erasure networks is a more feasible task than general noisy networks—indeed, we were able to derive secure message capacity characterizations in a number of cases. And although we do not focus on the practical aspects in this paper, we have first evidence that these ideas can translate to practical networks: in Section VI we briefly discuss a method that enables to create erasure channels out of physical wireless channels, and have shown in first experiments that it is possible to create secret keys at tens of Kbits per sec. The basic idea that enables translation to practice is to bound the amount of information Eve receives by artificially creating channel variability in a controlled manner, so that, no matter where Eve is located within a space, we ensure that she misses a certain fraction of the packets.

Interestingly, in all cases where we were able to characterize the secrecy capacity, the optimal scheme utilizes two phases: a key-sharing phase and a message-transmission phase. That is, Alice first exploits the network properties to create a shared key with Bob; and then encrypts her message with the shared key and transmits it, like in the one-time pad approach. The technical design comes into selecting, for each network, how to create keys and how much key to use to encrypt the message. Note that the optimality of the two-phase protocol is proved through an impossibility result that makes no assumptions on the structure of the secrecy protocol, i.e., does not restrict schemes to such two-phase protocols.

In the paper, we start from the simplest network—a single link—and gradually progress to larger networks, indicating in each case what are the new ideas that are needed to optimally create shared keys and achieve the secure message capacity. It turns out that we need to vary our designs, depending on the network structure and parameters. We develop linear programming (LP) formulations, that enable to efficiently select the optimal parameter values; and we prove that the identified solution achieves the secrecy capacity, by providing an outer bound, again in the form of an LP. We also discuss how these techniques extend to multiuser networks, where the network resources need to be shared, and where we may need protection not only against Eve, but also against honest but curious users who participate in the communication protocol.

The paper is organized as follows. Section II describes the communication and adversary models; Section III develops ideas for secure unicast transmission, starting from point-to-point communication to multihop networks; Section IV discusses multiuser secrecy, including private message broadcast, group key generation and oblivious transfer; Section V gives a brief overview of related work; and Section VI concludes the paper with a discussion and open questions.

II. MODELS, DEFINITIONS, AND BACKGROUND

A. Communication Model

In our communication scenarios we have one sender (S), one or more receivers (D_i) and adversaries. We often refer to the sender as Alice, to the receiver(s) as Bob (and Calvin) and to the adversary generically as Eve. In all our settings we consider one particular channel model: a discrete memoryless erasure channel with feedback in terms of ACKs (packet received) and NACKs (packet lost) from the legitimate receivers; this feedback is completely overheard by the adversaries. In particular, we assume that the state of the channel (erased or not erased) toward the *legitimate* receivers is available strictly causally to all parties. Note that the state of the adversary (Eve) is not known to the legitimate parties.

1) *Erasure Channel With ACK/NACK Feedback:* The input to the channel consists of a L-length[3] *packet* with symbols from a finite field $\mathbb{F}_q$. Let $\mathcal{X}$ denote the input alphabet of the channel, then $\mathcal{X} = \mathbb{F}_q^L$. Besides all input symbols, the channel output alphabet $\mathcal{Y}$ contains also an erasure symbol $\perp$: $\mathcal{Y} = \mathcal{X} \cup \{\perp\}$. We denote $X_i \in \mathcal{X}$ the input of the

[2] There are other works that use wireless properties to create security; see Section V for a sampling of these ideas.

[3] From an information theory perspective, we could use $L = 1$; from a practical perspective, we would like to use a value of L that makes the overhead of the packet header negligible.

channel in the ith time slot. We use the notation $X^n = (X_1, \ldots, X_n)$. We apply the same shorthand also for other vectors.

In the *broadcast* setting there are $M \geq 2$ receivers and the collection of channel outputs is $(Y_{1,i}, \ldots, Y_{M,i}) \in \mathcal{Y}^M$, where $Y_{j,i}$ is the observation of receiver j. The adversary is a subset of the receivers. We use Y_i as a shorthand for $(Y_{1,i}, \ldots, Y_{M,i}) \in \mathcal{Y}^M$, which is the channel output at time i. We assume that all erasures are independent and identically distributed (i.i.d.), with erasure probabilities $\delta_1, \ldots, \delta_M$

$$\Pr\{Y_i|X^iY^{i-1}\} = \prod_{j=1}^{M}\{Y_{j,i}|X_i\}$$
$$\Pr\{Y_{j,i}|X_i\} = \begin{cases} 1-\delta_j, & Y_{j,i} = X_i \\ \delta_j, & Y_{j,i} = \perp \end{cases}, \forall j \in \{1, \ldots, M\}. \quad (1)$$

In several situations (especially in Section III), we would have $M = 2$ receivers with one legitimate receiver ("Bob") and one eavesdropper ("Eve"). In this case, we use the special notation of δ denoting Bob's and δ_E denoting Eve's erasure probabilities respectively. We call this a *point-to-point* setting.

In a *network* setting there are more than one (ℓ) channels and maybe some intermediate nodes in addition to the sender and the receivers. Every channel operates as defined in the point-to-point setting, independently of each other. In this case we use the indices of the erasure probabilities and the first indices of variables to denote the index of the channel, e.g., $Z_{3,i}$ denotes the output symbol for Eve on the 3rd channel in the ith time slot, which is an erasure with probability δ_3. We again use Y_i as a shorthand for $(Y_{1,i}, \ldots, Y_{\ell,i})$ and Z_i as a shorthand for $(Z_{1,i}, \ldots, Z_{\ell,i})$. The context clarifies whether $Y_{j,i}$ denotes the output of receiver j or that of channel j.

B. Adversary Model and Security Notions

We consider an eavesdropping adversary, Eve, who aims to learn the message that Alice sends to another receiver. In the setting of Section III, Eve is passive, she does not transmit any signal. However, in Section IV-A, where the adversary could be one of the receivers, Eve participates in the protocol and therefore could be an "active" adversary. Next, we describe the security guarantees that we seek. We do not make any assumptions on the computational power of Eve. Instead, we only assume that the channel through which she observes the communication is not perfect; erasures occur on her channel also. In a network setting, the adversary can select a subset of channels to eavesdrop on. We assume that the maximum number of eavesdropped channels is known, but the actual subset of eavesdropped channels (which we sometimes call the *location of Eve*) is not.

1) *Information Theoretic Security Notions:* Information theoretic secrecy (also known as unconditional secrecy) is the security when there is no assumption on the computational power of the adversary. Formally, it is defined in terms of mutual information between the message and the observations of the adversary. It is common to distinguish *perfect*, *strong*, and *weak* secrecy [3]. Let $W \sim \text{Unif}\{1, 2, \ldots, 2^{nR}\}$ denote a message to be secured and E^n all observations that Eve has access to after n transmissions of the protocol. In this paper we use the notion of "strong secrecy" which effectively means that the confidential message and the observations of Eve are (asymptotically) independent

$$\lim_{n\to\infty} I(W; E^n) = 0$$
$$\lim_{n\to\infty} H(W|Y^n) = 0 \quad (2)$$

where Y^n are the observations of any legitimate recipient of W. Strong secrecy can be interpreted as requiring that the information leak toward Eve is negligible, while being able to reliably send the message to the legitimate recipients. The supremum of rates R for which these conditions can be met is called the *secure message capacity*.

A similar notion can be defined when two (or more) parties want to agree on a common key K, which is kept secure from adversaries whose observations are E^n after n transmissions of the protocol. In this case, for secure key agreement we need

$$\lim_{n\to\infty} I(K; E^n) = 0$$
$$\lim_{n\to\infty} \mathbb{P}(\hat{K} \neq K) = 0$$
$$\lim_{n\to\infty} H(K) - nR = 0, \text{ where } K \in \{0,1\}^{nR} \quad (3)$$

where $\hat{K}$ is the key of a legitimate participant created from its observations Y^n, and K is the common key that all the participants agree on.

III. SECURE UNICAST TRANSMISSION

A. Broad Approach

In this section we discuss unicast transmissions where there is a single destination for each message. All the schemes we describe in this section are optimal (capacity achieving), and share the following two-phase structure and design principles:

- *Phase I.* In this phase we create a shared key, between Alice and Bob, and potentially between

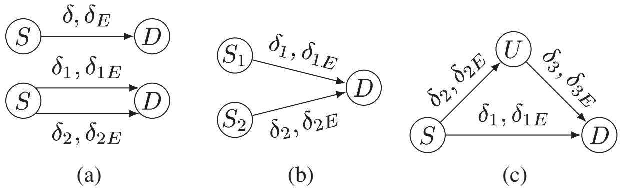

Fig. 1. *Our networks. Causal channel state feedback are sent over a separate noiseless public channel (not shown). (a) Direct channels. (b) V-network. (c) Triangle network.*

other nodes in the network which is secret from the adversaries. To create a shared key between two network nodes, we send random packets in the network so that, we maximize the amount of packets that the legitimate nodes share, while at the same time minimize the amount of the packets that Eve has also observed. Once two nodes share random packets, we estimate how many of these Eve has also overheard. We use this estimate to create a shared key using a maximum distance separable (MDS) code, as we describe in Section III-B.

- *Phase II*. In this phase, we use the shared key to encrypt the message, and transmit the message from Alice to Bob. In some cases, we may only use an end-to-end key, that Alice and Bob share. In other cases, we may also use link-by-link keys that we apply and peel off throughout the network. The overall goal is to use the least amount of key required so that the message is protected from Eve.

We analyze a sequence of networks, starting from the point-to-point single channel network [see Fig. 1(a) (top)], and progressing to a network with two parallel channels [see Fig. 1(a) (bottom)], then to a network with two sources with limited common randomness [V-network, Fig. 1(b)], building to the triangle network. In each of the steps, we introduce new techniques and ideas on how to efficiently share random packets and achieve the message capacity.

Interestingly, it turns out that all the achievability schemes can be expressed through LPs that require new techniques—which introduce new constraints to the LPs–as the network grows, yet also reuse the techniques (constraints) of the previous steps. Moreover, the LPs include as a special case the classical information flow LP without secrecy over the same configurations; essentially we augment them with (non-trivial) secrecy-related constraints. The solution of the LP schemes we present achieves the capacity. To prove this, we developed outer bounds also as linear programs, and showed that the inner and outer bound LPs have the same optimal value (see Section III-H).

B. A Common Tool: MDS Coding to Create Keys

Throughout this paper we use the following observation, that helps to efficiently create shared keys. Assume Alice and Bob share N random packets, and Eve has overheard (any) N' of these; we can then create N–N' linear combinations of the N random packets, such that Eve knows nothing about them. Using MDS codes enables us to efficiently create such linear combinations [4]. Let's think of packets as variables; essentially, what MDS coding ensures is that, no matter which N' variables Eve has observed, if we take the N–N' linear equations and substitute zero for the variables that Eve has, the resulting equations (in the remaining N–N' variables) are linearly independent. That is, we can create N–N' linearly independent packets that Eve knows nothing about, and these can serve as Alice-Bob keys.[4]

As an illustration, the previous observation enables us to implement the information theoretical key-agreement scheme of Maurer [6] and Ahlswede-Csiszár [7], using linear complexity and simple ACK/NACK feedback. Indeed, assume that we have an ideal erasure broadcast channel, where the erasure probability of the Alice-Bob channel is δ and of the Alice-Eve channel δ_E. Suppose Alice transmits n random packets. For large enough n, the erasure channel ensures that $N \approx (1-\delta)n$ packets are received by Bob, of which $N' \approx (1-\delta_E)(1-\delta)n$ are also received by Eve and leaving about $N - N' \approx \delta_E(1-\delta)n$ packets received by Bob and not Eve. Clearly, neither Alice or Bob know what are the packets Eve received, but only their cardinality. First, Alice learns which N packets Bob received, through simple ACK/NACK feedback. Alice and Bob establish a secret key by performing privacy amplification: they create a common key by forming about $n\delta_E(1-\delta)$ appropriately chosen linear combinations of the N packets received by Bob. By choosing these linear combinations through a MDS code [4], we can generate about $n\delta_E(1-\delta)$ bits of key that are completely secure from Eve. For the erasure channel, this corresponds to the optimal key-generation rate [6], [7].

C. Point-to-Point Secret Message Capacity

Consider the erasure channel depicted in Fig. 1(a) (top), where δ is the erasure probability to D and δ_E the erasure probability to Eve. Our aim is to maximize R, the secret message rate at which S can send a message securely to D. We use the following three ideas, that we next summarize, and discuss in more detail when we describe the scheme:

(I1). Time-sharing between two phases: in a first phase—for a time fraction k—we create a secret key between S

[4] The size of the MDS code affects the size of the finite field over which we need to operate. However, instead of MDS coding, for large values of n we can also use random coding over a finite field of fixed size (eg binary), and still asymptotically in n achieve the same properties [5].

and D; in the second phase—for a time fraction m—we use the created key to encrypt and send the message.

(I2). The amount of key we need to secure the message equals the amount of the message transmissions that Eve will actually observe (which is *smaller than the message size*).

(I3). We use ARQ for the encrypted message transmission.

Scheme Description and Discussion [8], [9]: In the first phase, Alice simply sends i.i.d. random packets generated from her private randomness; Bob publicly acknowledges the packets he has received. If the first phase lasts for a time fraction k, then Alice and Bob, using the MDS coding we previously described, can create a key at rate $k\delta_E(1-\delta)$.

The amount of key we need, is smaller than the length of the message (unlike the one-time pad requirement). Indeed, Eve is only going to receive a fraction of the packets intended for Bob in the second phase and thus we only need to create an amount of key that allows us to protect against this fraction. As a specific example, say Alice sends 10 encrypted messages, and because of erasures, Eve receives (some) 3 of them. To protect the 10 messages, it is sufficient that we use 3 key packets—and not 10. We simply take 3 key packets, create 10 linear combinations of them such that any three are linearly independent, and use these 10 linear combinations for encryption, by XOR-ing them with the 10 messages.

Note that how many encrypted message packets Eve receives, depends on the strategy Alice uses to convey the encrypted message to Bob. Alice is connected to Bob through an erasure channel; to transmit efficiently, she needs to use a capacity achieving technique. For instance, both ARQ and erasure coding are capacity achieving techniques and would lead to the same performance for unsecure message transmission over erasure channels. However, when we are interested in secure message sending, if we were to transmit the encrypted message using erasure coding, we would get a worse performance than with ARQ. This is because, with ARQ Eve may receive the same packet and thus the same information twice; while with erasure coding, *every* packet Eve receives gives her *new* information about the message.

In summary: in the second phase, the optimal scheme uses ARQ and transmits the message for a time fraction of m; Eve receives $m(1-\delta)((1-\delta_E)/(1-\delta\delta_E))$ packets. To be secure, we need to have created a key equal to this amount, and expand it to size m to encrypt the message.

Capacity: The secure message capacity equals [9]

$$R_{SM} = (1-\delta)\delta_E \frac{1-\delta\delta_E}{1-\delta\delta_E^2} L \log q. \tag{4}$$

LP 1 Point-to-Point LP

Input: Erasure probabilities δ and δ_E
Output: Secure message rate R

$$\begin{aligned} &\max\ R \\ &\text{s.t.}:\ R \le (1-\delta)m && (5)\\ &m+k \le 1 && (6)\\ &m(1-\delta)\frac{1-\delta_E}{1-\delta\delta_E} \le k\delta_E(1-\delta) && (7)\\ &m,k,R \ge 0 && (8)\end{aligned}$$

LP Formulation: Although we have a closed form expression for the capacity, we also provide an associated LP formulation, because the LPs for larger networks follow the structure introduced here. The solution of LP leads to the optimal rate R_{SM} in equation (4) [9].

We have three variables: R, the secret message rate that we maximize, k, the fraction of time that we use to create a secret key from private randomness, and m the fraction of time we use to send the message. Constraint (5) is a capacity constraint: using the erasure channel at a fraction m of the time enables a message rate at most $m(1-\delta)$. Constraint (5) expresses that we timeshare between key generation and message sending. Constraint (6) ensures that the amount of message that Eve receives is smaller than the amount of key we have created and thus the key is sufficient to securely encrypt the message.

D. Parallel Channels

Consider the setting displayed in Fig. 1(a) (bottom) where there are two parallel independent erasure channels, with erasure parameters δ_1, δ_{1E} (channel 1) and δ_2, δ_{2E} (channel 2). We assume that Eve wiretaps any one of the two channels (and we do not know which one). Clearly, on both channels we can apply the scheme we have previously described for a single channel and achieve the sum rate. This is not optimal, because we do not exploit the fact that we have multiple paths and Eve is not present in all of them. We need the following idea.

(I4). All the random packets that Alice sends through channel 1 and Bob successfully receives, can be used as a secret key on channel 2 (and symmetrically). For instance, if Eve is on channel 1, any packet received through channel 2 by Bob is completely secret from Eve, and can be contribute to the key used on channel 1.

Short Scheme Description: We divide the message into two parts and send them at rates m_1 and m_2 through channels 1 and 2 respectively; to protect them, we use a different key for each channel. Over each channel we implement essentially the same scheme as before (i.e., time-sharing between key-generation and a message sending phase, sending the encrypted messaged with ARQ, etc.); the difference is that we more efficiently generate keys using the idea (I4).

LP 2 Two Parallel Channels LP

Input: Erasure probabilities δ_i and δ_{iE}
Output: Secure message rate R

$$\max\ R$$

$$\text{s.t.}:\ R \leq (1-\delta_1)m_1 + (1-\delta_2)m_2 \tag{9}$$

$$m_1\frac{(1-\delta_{1E})(1-\delta_1)}{1-\delta_1\delta_{1E}} \leq k_2(1-\delta_2) + k_1\delta_{1E}(1-\delta_1) \tag{10}$$

$$m_2\frac{(1-\delta_{2E})(1-\delta_2)}{1-\delta_2\delta_{2E}} \leq k_1(1-\delta_1) + k_2\delta_{2E}(1-\delta_2) \tag{11}$$

$$m_1 + k_1 \leq 1 \tag{12}$$

$$m_2 + k_2 \leq 1 \tag{13}$$

$$R, m_1, m_2, k_1, k_2 \geq 0 \tag{14}$$

LP Formulation: We follow the structure of LP 1, with now five variables: R is the rate we maximize, k_1, k_2, and m_1, m_2 the fraction of time we use to send generate keys and send messages on channels 1 and 2, respectively. Constraints (12) and (13) express the timesharing between key generation and message sending. Constraint (9) is the capacity constraint across the two channels. In the secrecy constraints (10) and (11) besides the key generation terms as seen for the single channel, the terms $k_2(1-\delta_2)$ and $k_1(1-\delta_1)$ also appear; these capture the key generation packets received through the second (first) channel and used as secret keys on the other channel. The solution to this LP is the capacity (see Section III-H).

E. V-Network

Consider the V-network in Fig. 1(b): two sources S_1 and S_2 are connected to a common destination D through independent erasure channels. S_1 and S_2 have a common message and can generate unlimited amounts of private randomness, but have access to only a rate limited common random source Ψ. Essentially, this is very similar to two parallel channel setup, with the difference that we split Alice into two parts: both parts have the message, but they have limited common random packets to send. We look at this setup as an intermediate step to studying multihop networks, where now intermediate network nodes may share limited common randomness that they acquired from a common source.

The common randomness between the two sources is a valuable resource as it affects the key generation rate achievable by approach (I4) (packets received by D through one of the channels and used as a key on the other). To avoid wasting the common randomness we introduce two techniques:

(I5). We send part of the common random packets using ARQ (ensuring that these packet are definitely received).

(I6). We send another part of the common packets using approach (I4), with a twist: sources S_1 and S_2 transmit linear combinations of common random packets, so that, the set of received packets (either by the destination, or the eavesdropper, or both) are independent. Using coding makes this possible without knowing Eve's channel state.

Short Scheme Description and Discussion [10]: Our scheme combines several key generation methods. The common randomness Ψ is divided into three independent parts: r_1, r_2, and c. S_1 sends rate r_1 independent random packets using ARQ. These packets contribute to the key of S_1 with rate $r_1(\delta_{1E}(1-\delta_1)/(1-\delta_1\delta_{1E}))$; moreover, as they are known by S_2, but not an eavesdropper on the second channel, they also contribute to the key of S_2 with rate r_1. Source S_2 uses the rate r_2 packets from the common randomness in the same way. Part c is used as in I5: S_1 sends rate c_1 packets while S_2 sends rate c_2 packets; these packets are not necessarily independent, but are always innovative for D and Eve (taken together), i.e., they convey new information to D and Eve. This enables a key rate $c_1\delta_{1E}(1-\delta_1) + c_2(1-\delta_2)$ for S_1 and $c_2\delta_{2E}(1-\ \delta_2) + c_1(1-\delta_1)$ for S_2. Finally, each source sends random packets using private randomness at a time fraction p_1 and p_2; these packets contribute only to the key of the source that transmitted them.

The intuition behind (I6) is the following. In the point-to-point case, the optimal key-generation scheme has Alice generate uniform at random packets and send these to Bob; this has the advantage that packets that Eve receives and Bob does not, give no information to Eve about the packets that Bob receives. (I6) mimics this more efficiently: when Alice sends uniform at random packets, there exist some packets that neither Bob nor Eve receive; thus in a sense these packets do not serve any purpose. To avoid this, Alice could simply expand the k random packets to $k/(1-\delta\delta_E)$ packets, so that, when Alice transmits the expanded packets, Eve cannot learn anything about the packets that Bob receives, from the packets that only she (and not Bob) has collected. This is exactly what (I6) does across S_1 and S_2. The LP we provide next selects what fraction of the packets to send using MDS, and what fraction to send using ARQ, separately for each edge. ARQ has the advantage that it preserves all random packets, and the disadvantage that Eve collects more of them.

LP Formulation: Our parameters are now: the rate R; m_1 and m_2, the message transmitting fractions; r_1 and r_2, the fraction of time we use ARQ for common randomness; c_1 and c_2, the fraction of time we transmit linear combinations of common randomness; and p_1 and p_2, the fraction of time we use private randomness for key generation. As before, (15) is a min-cut constraint; (16) and (17) ensure we create sufficient key to protect the message; (21) and (22) ensure time-sharing; and the new constraints (18), (19), and (20) ensure sufficient rate in the limited common randomness.

LP 3 V-network LP

Input: Erasure probabilities δ_i and δ_{iE}, randomness Ψ
Output: Secure message rate R

$$\max\ R \text{ such that :}$$

$$R \leq (1-\delta_1)m_1 + (1-\delta_2)m_2 \tag{15}$$

$$m_1\frac{(1-\delta_{1E})(1-\delta_1)}{1-\delta_1\delta_{1E}} \leq r_2 + r_1\frac{\delta_{1E}(1-\delta_1)}{1-\delta_1\delta_{1E}} + c_2(1-\delta_2) + (c_1+k_1)\delta_{1E}(1-\delta_1) \tag{16}$$

$$m_2\frac{(1-\delta_{2E})(1-\delta_2)}{1-\delta_2\delta_{2E}} \leq r_1 + r_2\frac{\delta_{2E}(1-\delta_2)}{1-\delta_2\delta_{2E}} + c_1(1-\delta_1) + (c_2+k_2)\delta_{2E}(1-\delta_2) \tag{17}$$

$$H(\Psi) \geq c + r_1 + r_2 \tag{18}$$

$$c \geq (1-\delta_1\delta_{1E})c_1 + (1-\delta_2)c_2 \tag{19}$$

$$c \geq (1-\delta_2\delta_{2E})c_2 + (1-\delta_1)c_1 \tag{20}$$

$$1 \geq p_1 + m_1 + c_1 + \frac{r_1}{1-\delta_1} \tag{21}$$

$$1 \geq p_2 + m_2 + c_2 + \frac{r_2}{1-\delta_2} \tag{22}$$

F. Triangle Network

Consider now the triangle network in Fig. 1(c), where S (Alice) is connected to D (Bob) through a relay U. In this case as well, the optimal scheme first generates sufficient amounts of key for each channel and then employs it for message encryption. Note that the relay U and the source S will share limited rate common randomness, from random packets that S sends to U through the $S-U$ channel, which is very similar to the V-network setup. Yet there is also a significant difference with the V-network: the relay U does not have the message; it can only receive it by consuming channel $S-U$ resources. To overcome this, we need to use two more ideas, that essentially reduce the amount of message we need to send to U.

(I7). The encrypted message packets that travel on the S–U–D path can potentially be encrypted with three types of key: key that only S–U share (say K_{SU}); key that only U–D share (say K_{UD}), and key that S and D share (say K_{SD}). The source can send to U messages encrypted with K_{SU} and K_{SD}; U will need to remove K_{SU} from these encrypted messages (since D does not have it), yet it does not need to remove K_{SD}. That is, *U does not need to completely decrypt the message*, but can essentially assume that K_{SD} provides part of the randomness needed to secure the message over the UD link, and use the key K_{UD} to provide the remaining randomness.

(I8). Assume that S creates a packet P from the common randomness as we did in (I5). Instead of sending P as previously, S now combines P with a message packet W and sends $P' = P \oplus W$ to D. Note that D cannot yet decode W. Packet P is available for U and is transmitted on the $U-D$ channel using ARQ. This transmission is utilized in two ways. First, note that $W = P' \oplus P$, thus they allow D to decode a message packet. Thus, as far as D is concerned, transmission of P from U can be considered as part of the message sending phase, but there is no need to further encrypt them, because P is independent from the message. Second, as far as Eve on the $U-D$ channel is concerned, these are random key generation packets forwarded using ARQ, so they also contribute to the key K_{UD} on the $U-D$ channel. We refer the interested reader for the detailed description of the scheme to [11].

LP 4 Triangle Network LP

Input: Erasure probabilities δ_i and δ_{iE}
Output: Secure message rate R

$$\max\ R \text{ such that:}$$

$$R \leq (1-\delta_1)m_1 + (1-\delta_3)m_3 \tag{23}$$

$$m_1(1-\delta_1)\frac{1-\delta_{1E}}{1-\delta_1\delta_{1E}} \leq (k_1+c_1)\delta_{1E}(1-\delta_1) + r_3 + c_3(1-\delta_3) \tag{24}$$

$$m_2(1-\delta_2)\frac{1-\delta_{2E}}{1-\delta_2\delta_{2E}} \leq k_2\delta_{2E}(1-\delta_2) + k_1(1-\delta_1) \tag{25}$$

$$m_3(1-\delta_3)\frac{1-\delta_{3E}}{1-\delta_3\delta_{3E}} \leq (k_3+c_3)\delta_{3E}(1-\delta_3) + (k_1+c_1)(1-\delta_1) + r_3\delta_{3E}\frac{1-\delta_3}{1-\delta_3\delta_{3E}} \tag{26}$$

$$k_2(1-\delta_2) \geq c + r_3 \tag{27}$$

$$c \geq c_1(1-\delta_1\delta_{1E}) + c_3(1-\delta_3) \tag{28}$$

$$c \geq c_3(1-\delta_3\delta_{3E}) + c_1(1-\delta_1) \tag{29}$$

$$(1-\delta_3)m_3 \leq (1-\delta_2)m_2 + c_1(1-\delta_1) \tag{30}$$

$$1 \geq k_1 + m_1 + c_1 \tag{31}$$

$$1 \geq k_2 + m_2 \tag{32}$$

$$1 \geq k_3 + m_3 + c_3 + \frac{r_3}{1-\delta_3} \tag{33}$$

LP-Formulation: LP 4 has very similar format to the previous cases, with the difference that we now also use (I7) and (I8) for efficiency (see [11]). Briefly, variables m_1, m_2, m_3 correspond to the message phase on each channel, while the other variables correspond to various key generation methods. Inequality (23) is a rate constraint arising from the length of message sending phases on the S–D and U–D channels. The next three constraints (31)–(33) formulate the time-sharing constraints on each channel. Inequalities (24)–(26) ensure that each channel has sufficient amount of key at its disposal to secure against the eavesdropper on the given channel in the message sending phase. Inequalities (24) and (25) come directly after accounting for the corresponding key rates. Since U uses keys only in the last step of the message sending phase, (26) is not that straightforward, but a short calculation shows that (26) guarantees security on the U–D channel [11]. Constraints (27)–(29) correspond to the amount and the use of common randomness that S and U share, but with the difference that now the key generation packets that U receives constitute the common randomness. Inequality (30) limits the length of the message sending phase on the U–D channel, the constraint follows from the fact that U does not have access to W.

G. Extension to Arbitrary Networks

A natural question is, can we extend these results to arbitrary networks. Unfortunately, it is clear from the previous sections that, as the size of the network increases, so does the complexity of the secure message capacity achieving scheme. For the small networks we discussed we could derive secure message capacity characterizations, yet as soon as we go to a network with more than one hop and multiple nodes, there exists an exponential number of subsets of nodes that can generate randomness, or exploit shared randomness, create shared keys, and apply or peel off keys. Indeed, finding the secrecy capacity of a general network is as hard as determining the capacity region of multiple unicast network coding, which is a long-standing open problem [12], [13]. Thus exact secrecy capacity characterizations seem challenging.

On the positive side, we can easily extend the ideas we have developed to design new achievability schemes for arbitrary networks, that leverage erasures and feedback to achieve secrecy rates that are in some cases multiple times higher than the best alternatives in the literature.

To illustrate, consider the secure network coding setup, introduced by Cai and Yeung [14]–[16]: a source wants to securely send a message to a set of receivers over a packet network with error-free unit-rate edges, in the presence of a passive eavesdropper, Eve; the min-cut to each receiver equals h and Eve wiretaps any z edges of the network. By exploiting the fact that there will exist at least $h - z$ paths towards each receiver that Eve will not overhear, it is possible to securely communicate at rate $h - z$.

Now instead of the unit-rate error-free edges that secure network coding assumes, consider the simplest generalization, independent erasure networks with the same erasure probability δ in each channel. We assume that there exist h edge-disjoint paths from the source to each receiver, and that our eavesdropper, Eve, observes any z channels in the network. To apply secure network coding in this setup, we need to first apply channel coding to transform the erasure channels to error-free links; channel coding followed by secure network coding would lead to rate $(h - z)(1 - \delta)$. Clearly, this is suboptimal, as we do not exploit the erasures for secrecy.

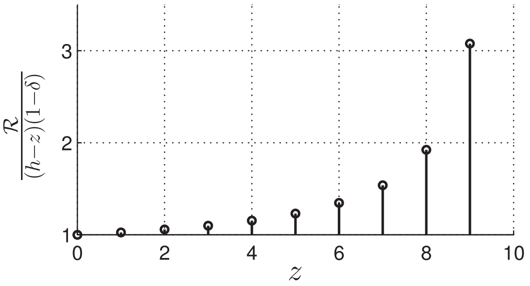

Fig. 2. ***Advantage of exploiting erasures and feedback, as a function of the number of eavesdropped edges z, when h = t = 10, $\delta = \delta_E = 0.3$. Note that this figure applies for all network topologies with these parameters.***

To improve upon this scheme, we first observe that secure network coding can also be expressed as two phase scheme through a modified construction [17]. The original construction in [14] simultaneously sends keys and messages encrypted with the keys to the receiver; the receiver solves linear equations to retrieve at the same time both the keys and the messages. Our modified construction explicitly utilizes a key-sharing phase, where the source establishes secret keys with the receivers and a message sending phase, that uses these established secret keys to encode and securely send messages. Although there is no difference in achievable rates between the original and our modified approach, by explicitly separating the two phases enables to optimize each phase on its own. Compared to error-free networks, by exploiting erasures and feedback we gain benefits in two ways: 1) we can create keys at higher rates during the first phase; and 2) we need create a smaller amount of keys to protect the message-sending during the second phase, since Eve observes less relevant information over erasure networks [17], [18].

Fig. 2 shows the ration of the rate R one of our algorithms achieves, divided with the rate $(h - z)(1 - \delta)$ that channel coding followed by secure network coding achieves, as a function of the number of eavesdropped edges, z.

H. Brief Outline of Outer Bounds

We can show that the schemes in Section III-C to F are optimal. To illustrate the proof technique, consider the parallel channels from Section III-D. We derive a number of general inequalities and then treat the different entropy and mutual information terms as arbitrary nonnegative variables. This turns the information inequalities into linear constraints resulting in another linear program, which we call the outer bound LP. Specifically, we show in [18] that the secret message transmission capacity of two parallel channels is upper-bounded by the value to the following LP:

$$\max\ R$$

$$\text{s.t.}: R \leq (1-\delta_1)m_1 + (1-\delta_2)m_2 \tag{34}$$

$$m_1 \frac{(1-\delta_{1E})(1-\delta_1)}{1-\delta_1\delta_{1E}} \leq k_2(1-\delta_2) + k_1\delta_{1E}(1-\delta_1) \tag{35}$$

$$m_2 \frac{(1-\delta_{2E})(1-\delta_2)}{1-\delta_2\delta_{2E}} \leq k_1(1-\delta_1) + k_2\delta_{2E}(1-\delta_2) \tag{36}$$

$$m_1 + k_1 \leq 1 \tag{37}$$

$$R, m_1, m_2, k_1, k_2 \geq 0. \tag{39}$$

In our converse, the variables m_1, m_2, k_1, k_2 in the LP are in fact the following:

$$m_j = \frac{1}{n}\sum_{i=1}^{n} I(X_{j,i}; W | Y^{i-1}, F^{i-1})$$

$$k_j = \frac{1}{n}\sum_{i=1}^{n} H(X_{j,i} | Y^{i-1}, F^{i-1}, W)$$

for $j = 1, 2$, where n is number of channel uses, W is the secret message, X_j is the input to the jth channel, Y is the (vector of) outputs of the two channels, and F is the public feedback. We may interpret k_j as the fraction of time Alice devotes to key generation on channel-j and m_j as the fraction she devotes to sending (encrypted) message over that channel. The constraint (34) comes via Fano's inequality from Bob's probability of error in decoding the secret message being small. Secrecy implies the constraints (35) and (36). The constraint (35), for instance, can be interpreted as a balance between the amount of secret key available for securing message transmission on channel 1 (RHS) and the amount of secret key required for securing the message sent over channel 1 from an eavesdropper on that channel (LHS). The rest of the constraints are just regularity conditions on the variables; (37) and (38) come from the total fraction of time not exceeding 1, and (39) from fractions of time and rate being nonnegative. In this case the outer bound LP matches the inner bound LP from Section III-D proving the optimality of the scheme there.

For the triangle network (Section III-F), the capacity of any cut of the network is an obvious outer bound. The triangle network has two cuts, the S–UD and the SU–D cut. However, using a similar technique as above we can show [11] that the capacity of the triangle network is not the minimum of the cut values. In other words, the min-cut value of the network is not achievable, in general. Our outer bound LP in this case turns out not to be identical to the inner bound LP. Still we can show through a number of reduction steps that the optimal value of the outer bound LP is the same as the optimal value of the inner bound LP proving the optimality of the scheme. For details we refer the reader to [19].

IV. MULTIUSER NETWORKS

In this section we highlight some ideas in *multiuser* secrecy over erasure broadcast channels. The distinguishing theme is that we need protocols that are secure even when untrusted users actively participate in them. For most ideas in this section, the security is provided against users who are honest but curious; i.e., follow the protocol correctly, but would like to extract the secret of the other users.[5] In Section IV-A, we focus on confidential individual message broadcast to multiple users. In Section IV-B we develop protocols for group key generation motivated by the need for subset of participants to share a common and secure key, which is kept secret from another (nonintersecting) set of participants as well as eavesdroppers. Finally, in Section IV-C, we discuss preliminary ideas of getting new functionalities like oblivious transfer (OT) between two or more untrusting parties in the presence of eavesdroppers.

A. Secure Private Message Broadcast

Consider an erasure broadcast channel with M receivers $D_1, \ldots, D_M$, and a sender Alice, who has a private message for each of the receivers. The goal is to find the set of all achievable M-tuples $(R_1, \ldots, R_M)$, so that, the private message to each receiver is kept secret from the other receivers, even if they all collude. We will again use a two-phase (key-creation, message-sending) protocol, where we now add the following ideas.

(I9). In Phase I, Alice broadcasts *the same* random packets towards all users, and exploits the fact that every user D_i successfully receives a different subset of these packets, to create private keys K_i with each user.

(I10). Even if users are malicious and *dishonestly* acknowledge which packets they have received, we can in some cases exploit coding to build a protocol that still guarantees to the honest users the optimal secure message rate.

Scheme Description and Discussion: We distinguish between honest and malicious users.

[5] We also briefly mention cases where we can get security against users that maliciously do not follow the protocol.

• *Honest but curious users*. In the first phase, Alice sends i.i.d. uniform packets, and all receivers publicly acknowledge the packets they receive. The packets received exclusively by receiver D_k is the secret key between that receiver and Alice. To create the secret key between Alice and receiver D_k, we use ideas (I1) and (I2), assuming that all the other receivers collude, and jointly play the role of Eve.

In the second phase, these keys are used to encrypt the private messages. To deliver the encrypted messages efficiently, we need to use a capacity-achieving strategy for insecure message transmission (like we used ARQ in the point-to-point channels, that is capacity achieving for insecure message transmission over erasure channels). In this case, we adapt the capacity achieving scheme of [20] and [21] for nonsecure message transmission over erasure broadcast channels. That scheme achieves all rate-tuples $(R_1, \ldots, R_M)$ satisfying

$$T_2(R_1, \ldots, R_M) := \max_{\pi} \sum_{i=1}^{M} \frac{R_{\pi_i}}{1 - \prod_{k=1}^{i} \delta_{\pi_k}} \leq 1 \qquad (40)$$

where the maximization is taken over all permutations π of $\{1, \ldots, M\}$. This is optimal for $M \leq 3$ or for a symmetric channel with $M > 3$ [20], [21]. $T_2(R_1, \ldots, R_M)$ is the proportion of time taken by the second phase of our protocol (as a fraction of the duration of the entire protocol).

We adapt this for our second phase as follows. In a first subphase, Alice repeatedly sends each encrypted data packet until it is received by at least one of the receivers. If the packet is received exclusively by the intended receiver the key is reused. In the second subphase, Alice sends linear combinations of the encrypted packets that were not received by the intended receiver in the first subphase. This allows her to take advantage of the side-information she receivers in the form of packets "incorrectly" delivered during the first subphase. A property of the scheme is that an encrypted packet forms part of a transmitted linear combination only until the channel delivers such a packet to the intended receiver (see [22] for more details).

To compute the amount of secret key required to encrypt packets to D_j (cf. idea (I2) of Section III-C), we may treat the rest of the users as one super-eavesdropper with an erasure probability of $\prod_{k \neq j} \delta_k$. Notice that an encrypted packet meant for D_j is repeated over the channel, either on its on or as part of a linear combination (whose other elements are known to the super-eavesdropper), only until it is received by the intended receiver D_j. Hence, by the same reasoning as in Section III-C, the rate of the secret key between Alice and D_j is $R_j((1 - \prod_{k \neq j} \delta_k)/(1 - \prod_{k=1}^{M} \delta_k))$. Since the secret-key generation rate for Alice and D_j is $(1 - \delta_j) \prod_{k \neq j} \delta_k$, this translates to the proportion of time taken by the first phase to be at least

$$T_1(R_1, \ldots, R_M) := \max_{j} R_j \frac{1 - \prod_{k \neq j} \delta_k}{1 - \prod_{k=1}^{M} \delta_k} \frac{1}{(1 - \delta_j) \prod_{k \neq j} \delta_k}.$$

Thus all rate-tuples $(R_1, \ldots, R_M)$ satisfying the following are achievable:

$$T_1(R_1, \ldots, R_M) + T_2(R_1, \ldots, R_M) \leq 1.$$

Using a similar converse technique as in Section III-H, we can show the optimality of this scheme whenever (40) is optimal for the nonsecure message transmission problem (e.g., for $M \leq 3$ or for a symmetric channel with $M > 3$ [22]).

• *Malicious users*. The above scheme depended on all users honestly reporting ACK/NACK's. However, it turns out that for the special case of $M = 2$, the same secure message rates can be supported even if a user deviates from the protocol by maliciously misreporting ACK/NACK's. For this, we make the following modifications:

In the first phase, when Alice and a receiver, say, D_1 want to generate a secret key, rather than rely on the feedback from D_2, they use privacy amplification as in Section III-B on the packets received by D_1. In the second phase, an expanded secret key is used, again as in Section III-C. Care is taken to ensure that no user is pretending he received significantly less packets than what expected from the typical behavior of its channel. When such atypical behavior is detected, that user is treated as an eavesdropper and no further attempts are made to deliver its message; from then on, the protocol proceeds along the lines of the point-to-point case with eavesdropper in Section III-C. Since the security of our scheme from Section III-C only depends on ensuring that an adversary does not obtain more than its typical share of eavesdropped packets about another user, the modified scheme ensures security even against malicious adversaries. See [22, Section VI] for details.

B. Group Keys and Network Coding

Unlike the previous sections, where we looked at secure message sending, we give here an example of creating a *shared key* among a group of wireless users. The users may utilize this key in a variety of ways, possibly to bootstrap, for instance, cryptographic protocols; we here focus only on the shared key creation. Thus clearly, we only have a key-creation phase.

We consider n wireless nodes $T_0, \ldots, T_{n-1}$ who want to share a group secret key which is secure from another set of nodes denoted by $\mathcal{E}$, which shares the wireless medium. All users can broadcast through an erasure broadcast channel and also have access to an insecure public channel (which is reliable but can be heard by all participants including eavesdroppers). For simplicity we assume that only T_0 (like a wireless access point) can transmit over the broadcast erasure channel to all the users.[6] We use the following idea.

(I11). T_0 creates a pairwise secret key $\mathcal{S}_i$ of size k_i with each terminal T_i, which is completely secure from $\mathcal{E}$ eavesdroppers, using the broadcast erasure channel. Then T_0 uses the public (insecure) channel and coding to create a *a common shared key* among all users that has size $\min_i k_i$.

Scheme Description: The group key generation occurs in the following steps: First, T_0 broadcasts a set of random packets, which get independently erased at each node T_i and in nodes in $\mathcal{E}$. The (legitimate) nodes then respond through ACK/NACK enabling T_0 to create pairwise keys K_i with every user $T_i, i = 1, \ldots, m-1$, that are secure from the eavesdropper $\mathcal{E}$, using the ideas (I1) and (I2). In the second step, using principles of network coding, T_0 broadcasts, using the insecure public channel, maximally useful linear combinations of the pairwise keys $\{K_i\}$ to reconcile them among the users, creating a shared randomness $(K_1, \ldots, K_{m-1})$. Finally, as this broadcast leaks some information, a secure key K can then be extracted from the (partially insecure) key $(K_1, \ldots, K_{m-1})$ which is a common knowledge among all legitimate nodes.

For example, suppose we have three terminals T_0, T_1, T_2, which we call Alice, Bob and Calvin; along with an eavesdropper Eve. Suppose Alice broadcasts a set of packets $\{x_i\}$ and suppose Bob received x_1, x_2, x_3, x_4, x_5, and Eve received x_1, x_2, x_3, x_7, x_8 and did not receive x_4, x_5, x_6. Then in phase 1, Alice and Bob construct $K_1 = \{x_1 \oplus x_3 \oplus x_5, x_2 \oplus x_4\}$, which is completely secure from Eve. In a similar manner suppose Calvin has received x_2, x_4, x_6, x_7, x_8 and therefore Alice-Calvin can create $K_2 = \{x_2 \oplus x_7 \oplus x_4, x_2 \oplus x_4\}$, which is again completely secure from Eve. Now, let us call $y_1 = x_1 \oplus x_3 \oplus x_5, y_2 = x_2 \oplus x_4, y_3 = x_2 \oplus x_7 \oplus x_4$, which is (K_1, K_2) and are completely secure from Eve. To reconcile the information, Alice needs to send just one transmission using network coding; in this case as Bob knows $\{y_1, y_2\}$ and Calvin knows $\{y_2, y_3\}$, by sending $y_1 \oplus y_3$ over the public channel, Alice can ensure that Bob and Calvin know $\{y_1, y_2, y_3\}$. However, Eve now knows $y_1 \oplus y_3$. Therefore to create a group secret that Eve does not know anything about, Alice-Bob-Calvin can create $K = \{y_1 \oplus y_2 \oplus y_3, y_2 \oplus y_3\}$. Note that the linear combinations should be chosen with care so that they are secure from Eve.

Interestingly, in the above example, the overall shared secret K has the same size as K_1 and K_2. In fact this observation is true in general for erasure broadcast channels. It can be shown that this protocol yields the group key is of the same size as the minimum size of secure pairwise key that can be generated between T_0 and terminals $\{T_i\}$. For channels which have independent erasures, this can also be shown to be the largest group key that can be created; i.e., that this protocol is in fact optimal [8], [23]. Some preliminary experimental observations and ideas to make this protocol implementable in a real wireless testbed are given in Section VI.

C. Oblivious Transfer

In secure multiparty computation mutually distrusting users compute functions of their data in such a fashion that they learn nothing more about other users' data than the output of the functions, i.e., in contrast with previous sections where secrecy is required only against eavesdroppers, here secrecy is desired even against the users engaged in the computation. It is well-known that, in general, such computations are not possible with information theoretic security guarantees (against all users) if the users only have access to private/common randomness and noiseless communication [24]. However, in a pair of seminal papers, Ben-Or, Goldwasser, and Wigderson [25] and Chaum, Crépeau, and Damgård [26] showed that if the number of colluding users is guaranteed to be strictly smaller a certain fraction of the number n of users computing, it is possible to (information theoretically) securely compute any function. This fraction is $n/2$ for the honest-but-curious model where users faithfully follow the given protocol, but at the end of the execution of the protocol they may collude to obtain additional information from everything they have observed in the course of executing the protocol, and it is $n/3$ in the malicious model where colluders may arbitrarily deviate from the protocol. When the number of colluders cannot be guaranteed to satisfy such thresholds, additional stochastic resources such as noisy channels can be used to obtain secure computation. Crépeau and Kilian [27] showed that two users with access to a noisy channel can securely compute any function. The approach was to obtain a primitive secure computation called oblivious transfer (OT) from the noisy channel and then use a reduction of the two-user secure computation to OT due to Kilian [28].

Oblivious Transfer: is a secure two-user computation. In 1-of-2 string OT, one of the users, say, Alice, has two bit strings B_0, B_1 each of length nR. The other user, say, Bob, wants to learn exactly one of the two strings. Let $U \in \{0,1\}$ represent Bob's choice bit, i.e., Bob requires B_U. Bob does not want Alice to find out U, while Alice wants to ensure that Bob does not learn anything about $B_{\overline{U}}$, where $\overline{U} = U \oplus 1$. As in previous sections we consider a broadcast erasure channel from Alice to Bob and an eavesdropper Eve, and let n be the number of channel uses. In addition, we assume the availability of a noiseless public

[6] Some of these ideas can also be extended when each user can also broadcast with different broadcast erasure channels.

discussion channel over which Alice and Bob may exchange messages which are observed by Eve as well[7]. In addition to privacy of B_0, B_1, and U from Eve, we will also require privacy for Alice and for Bob against the other user in collusion with Eve (2-privacy).

Let A_n, B_n, C_n denote what Alice, Bob, and Eve, respectively have access to at the end of the protocol. The goal is to characterize the largest R such that Bob's output has vanishing probability of error and the following privacy goals are met

$$\lim_{n\to\infty} I(C_n; B_0, B_1, U) = 0$$
$$\lim_{n\to\infty} I(A_n, C_n; U|B_0, B_1) = 0$$
$$\lim_{n\to\infty} I(B_n, C_n; B_{\overline{U}}|U, B_U) = 0.$$

For noisy channels without an eavesdropper, Nascimento and Winter [29] obtained a lower bound on the OT capacity of noisy channels and distributed sources in the honest-but-curious model. Ahlswede and Csiszár [30] characterized the honest-but-curious OT capacity for generalized erasure channels. Building on [29], [30] the OT capacity of the binary erasure channel with an eavesdropper was characterized for the honest-but-curious model in [31]; we briefly describe a protocol below which shows the achievability of capacity $C = \delta_E \min(\delta, 1-\delta)$. This was extended to obtain inner and outer bounds on the honest-but-curious OT capacity region of the two-user binary erasure broadcast channel in [32].

Protocol: Alice sends a sequence X^n of independent and uniformly distributed bits over the channel. Based on the Y^n received, Bob forms two disjoint and equal sized subsets $L_0, L_1 \subset \{1, \ldots, n\}$ of indices as follows: L_U is picked uniformly at random from among the unerased indices of Y^n, while $L_{\overline{U}}$ is chosen uniformly at random from the erased indices. Roughly, these equal-sized subsets are of lengths $n\min(\delta, 1-\delta)$. Bob sends the sets L_0, L_1 to Alice. Alice creates two secret keys K_0 and K_1 using the bits in X^n at indices given by K_0 and K_1 respectively; these are each of length about $n\delta_E \min(\delta, 1-\delta)$. Thus, both of secret keys are secret from Eve and only one of it, namely K_U, is known to Bob. The lengths of B_0 and B_1 are the same as the sizes of K_0 and K_1. Alice sends the encrypted strings $B_0 \oplus K_0$ and $B_1 \oplus K_1$ over the public channel. Since Bob knows the secret key K_U, he is able to recover B_U. Eve has no significant information about the two secret keys and hence she is unable to learn anything about B_0, B_1. Alice and Eve acting together will still be unable to determine U since they have no information about the erasure process of Bob's channel. Furthermore, even if Bob and Eve collude, they are unable to learn $K_{\overline{U}}$ and hence $B_{\overline{U}}$ (see [31] and [32] for a formal description of the protocol, proof of achievability, and converse).

V. RELATED WORK

We briefly overview work related to the topic of this paper. This is not meant to be comprehensive, but just gives a flavor of the many interesting ideas developed with a close relationship to ideas reviewed in this paper.

Secure transmission of messages using noisy channel properties was pioneered by Wyner [33], who characterized the secret message capacity of wiretap channels. This led to a long sequence of research on information-theoretic security on various generalizations of the wiretap channel [34], [35]. Notably, when the eavesdropper and legitimate channel are statistically identical, then the wiretap framework yields no security. The fact that feedback can give security even in this case was first observed for secret key agreement by Maurer [6] and further developed by Ahlswede–Csiszár [7]—but secure key agreement is not the same as secure transmission of *specific* messages. Group key generation from multiuser channels using public discussion was studied by Csiszár and Narayan [36]. The wiretap channel with secure feedback and its variants for message security have been studied in [37] and [38]; some conclusive results are developed in special cases when there is a secure feedback inaccessible to the eavesdropper. Security of private message broadcasting *without feedback* has been studied in [39], where some conclusive results have been established. The use of feedback and broadcast for private message transmission, *without* security requirements has been studied in [20] and [21]. The study of information theoretic security over (arbitrary) relay networks has been studied in [40]–[42] and references therein. We believe that the line of work we reviewed offers the first conclusive results that use insecure (and very limited) feedback for information-theoretic security of (multiple individual) messages.

Cai and Yeung proposed secure network coding [14] which studies secure message transmission over noiseless networks where an adversary can eavesdrop on any subset of edges of the network of a certain cardinality. Several extensions including resilience to Byzantine adversaries have been studied [43]. However, the general problem remains open [44].

Implementable protocols for creating pairwise shared secrets in wireless networks have been explored using channel reciprocity to create common randomness which can be made secure against eavesdroppers (see [45] and references therein. However, they need very rapid channel changes, accurate channel measurements and strong reliance on reciprocity extract a much lower rate of secret bits [46]. Pairwise secret keys using erasures and feedback have also been explored in [47] through an implementation in a WiFi setting.

VI. DISCUSSION AND OPEN QUESTIONS

In this paper we reviewed some of our recent work that exploits properties of erasure broadcast channels along with feedback to create secure message transmission. In this section, we discuss implications, extensions and open questions.

Connecting the Erasure Model to Wireless: Our work assumed erasure broadcast channels with ACK/NACK state feedback. This model fits well with packet-based wireless networks where packet losses correspond to erasures, however, to translate the ideas we developed into a practical security mechanism, there are still several challenges to address, including (i) what if an eavesdropper does not use commodity hardware and can snoop the physical layer signal obtaining more leaked information? (ii) does there exist enough time-variation to create a rich enough set of packet erasures, or can we enforce it? (iii) how does one know the parameters and statistics of the erasure channel?

To address (i), we propose to use two-layer wiretap codes to create erasures out of SNR variations [33], [34]. If Alice transmits using such a wiretap code, then whenever the reception signal-to-noise ratio (SNR) is higher than a threshold (say SNR_1) the information is received correctly, while if the SNR is below SNR_2), then no information is leaked; this effectively creates an information-theoretic erasure channel. Practical implementations of wiretap codes are reported in [48].

For (ii), one can create time and space-varying behavior artificially through either random beam-forming by Alice or through time-varying random interference (noise) insertion by infrastructure wireless nodes. This can effectively create situations where all nodes which can overhear Alice's transmissions experience time-varying (and independent) reception SNRs. Therefore in conjunction with wiretap coding, this potentially creates the broadcast erasure channel modeled in this paper. In order to learn the parameters of such an erasure channel, legitimate nodes can collect statistics and with sufficient richness in calibration, one could estimate the (erasure) channel statistics, partially addressing (iii). These questions were partially explored in [49] and [50], which demonstrated a preliminary test-bed implementation for group and multiple pairwise keys for erasure (wireless) networks, generating in some cases several thousand bits per second of secret keys.[8]

An implicit idea is a *layered* approach to wireless network security. We suggested using physical channel signaling (through wiretap codes) to create a broadcast erasure channel. Using the same physical layer, one can create several different functionalities at higher layers, like secure unicast message transmission (Section III), confidential individual message broadcast (Section IV-A), group key generation (Section IV-B) or oblivious transfer (Section IV-C). This is akin to a layering of functionalities often used in networking systems [51].

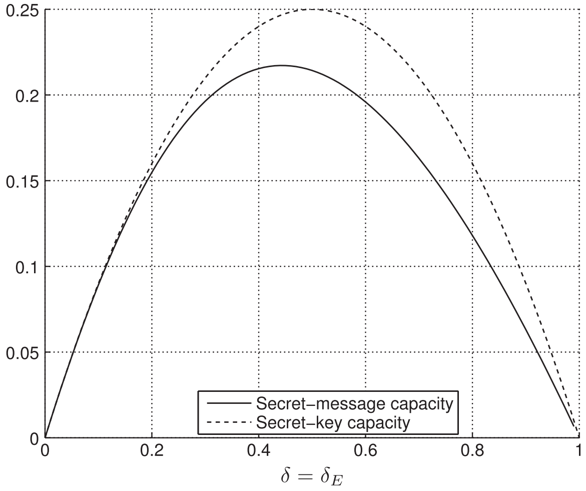

Fig. 3. ***Secret message and secret key capacities for point-to-point channel with $\delta = \delta_E$.***

Secret Messages Versus Secret Keys: Most of this paper, except for Section IV-B and C, was about secure message transmission. However, as part of secure message sending, we also developed optimal secret key sharing mechanisms (Phase I of our algorithms). We underline the critical difference between message transmission and key agreement. In message transmission a *specific* message is to be secured, whereas in key agreement the ultimate key is not prespecified. Because our protocols have feedback, they enable to exploit this flexibility in key-agreement, and create keys at higher rates than securing messages, as demonstrated in Fig. 3.

Other Security Metrics and Adversary Models: In this paper we focused on strong secrecy as defined in Section -II-B. However, many ideas of this paper can also be carried out for *semantic security*[9] [53], as shown in [22]. This also leads to questions of security against an active adversary, who maliciously does not follow protocol. Partial answers to security against active attacks are given in [22] for confidential individual message broadcast of Section IV-A.

Some Open Questions: A number of open questions arise naturally from this line of work. A natural theoretical question would be to explore how much of the ideas

[8] In comparison earlier implementations such as in [45] (and references therein) produce a few 10 s of bits per second of secret key.

[9] As introduced in [52], "informally, a system is semantically secure if whatever an eavesdropper can compute about the cleartext given the ciphertext, he can also compute without the ciphertext."

developed for erasure channels can be extended to general discrete-memoryless networks. Even in the case of erasure networks, several questions remain unanswered, such as: the question of secret message transmission for arbitrary networks, explored in Section III-G remains unresolved; developing new protocols and/or better converses would be essential to the solve the confidential individual messages broadcast of Section IV-A to erasure channels with arbitrary number of users and parameters, along with passive eavesdroppers; Section IV-B and C used an insecure public discussion channel which was available for free, but, if one takes a rate penalty for creating such a channel, the optimality of the protocols in those sections need to be re-examined; developing testbed implementation of these ideas might give rise to many new questions both in security systems and in theory.

Conclusion: We presented what we believe is a promising line of work, exploiting wireless network properties to create security. Given the encouraging preliminary results, we hope that these ideas can offer an alternative, complementary source of security to cryptosystems, and can eventually help future networks become more efficient, reliable and secure. ■

REFERENCES

[1] [Online]. Available: URL: http://en.wikipedia.org/wiki/One-time_pad

[2] C. Shannon, "Communication theory of secrecy systems," *Bell Syst. Techn. J.*, vol. 28, pp. 656–715, Oct. 1949.

[3] M. Bloch and J. Barros, *Physical-Layer Security: From Information Theory to Security Engineering*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[4] F. MacWilliams and N. Sloane, *The Theory of Error Correcting Codes*. Amsterdam, The Netherlands: North-Holland Publishing Company, 1977.

[5] D. J. C. MacKay, *Information Theory, Inference, Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[6] U. M. Maurer, "Secret key agreement by public discussion from common information," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 733–742, Apr. 1993.

[7] R. Ahlswede and I. Csiszár, "Common randomness in information theory and cryptography—I: Secret sharing," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1121–1132, Apr. 1993.

[8] M. Jafari Siavoshani, S. Diggavi, C. Fragouli, U. K. Pulleti, and K. Argyraki, "Group secret key generation over broadcast erasure channels," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, 2010, pp. 719–723.

[9] L. Czap, V. Prabhakaran, C. Fragouli, and S. Diggavi, "Secret message capacity of erasure broadcast channels with feedback," in *Proc. Inf. Theory Workshop (ITW)*, 2011, pp. 65–69.

[10] L. Czap, V. M. Prabhakaran, S. Diggavi, and C. Fragouli, "Exploiting common randomness: A resource for network secrecy," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2013.

[11] L. Czap, V. M. Prabhakaran, S. Diggavi, and C. Fragouli, "Triangle network secrecy," in *Proc IEEE Int. Symp. Inf. Theory (ISIT)*, 2014, pp. 781–785.

[12] M. Langberg and M. Médard, "On the multiple unicast network coding conjecture," in *Proc. 47th Annu. Allerton Conf. Commun., Contr., Comput.*, 2009, pp. 222–227.

[13] W. Huang, T. Ho, M. Langberg, and J. Kliewer, "On secure network coding with uniform wiretap sets," presented at the IEEE Int. Symp. Netw. Coding (NetCod), 2013.

[14] N. Cai and R. Yeung, "Secure network coding on a wiretap network," *IEEE Trans. Inf. Theory*, vol. 57, no. 1, pp. 424–435, Jan. 2011.

[15] N. Cai and R. Yeung, "Secure network coding," in *Proc. Int. Symp. Inf. Theory (ISIT)*, 2005, p. 323.

[16] R. W. Yeung and N. Cai, "On the optimality of a construction of secure network codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2008, pp. 166–170.

[17] L. Czap, V. M. Prabhakaran, S. Diggavi, and C. Fragouli, "Secure network coding with erasures and feedback," presented at the Annu. Allerton Conf. Commun., Contr., Comput., 2013.

[18] L. Czap, C. Fragouli, V. Prabhakaran, and S. Diggavi, "Secure network coding with erasures and feedback," *IEEE Trans. Inf. Theory*, 2015.

[19] L. Czap, V. Prabhakaran, C. Fragouli, and S. N. Diggavi, "Secure network coding with erasures and feedback. [Online]. Available: URL: https://infoscience.epfl.ch/record/198478

[20] L. Georgiadis and L. Tassiulas, "Broadcast erasure channel with feedback—capacity and algorithms," in *Proc. IEEE Workshop Netw. Coding, Theory, Appl. (NetCod)*, 2009, pp. 54–61.

[21] C. Wang, "On the capacity of 1-to-k broadcast packet erasure channels with channel output feedback," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 931–956, Feb. 2012.

[22] L. Czap, V. Prabhakaran, C. Fragouli, and S. Diggavi, "Secret communication over broadcast erasure channels with state-feedback," *IEEE Trans. Inf. Theory*, 2015, to be published.

[23] M. Jafari Siavoshani, S. Mishra, S. Diggavi, and C. Fragouli, "Group secret key agreement over state-dependent wireless broadcast channels," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2011, pp. 1960–1964.

[24] E. Kushilevitz, "Privacy and communication complexity," *SIAM J. Discrete Math.*, vol. 5, no. 2, pp. 273–284, 1992.

[25] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proc. 20th Annu. ACM Symp. Theory Comput.*, 1988, pp. 1–10.

[26] D. Chaum, C. Crépeau, and I. Damgård, "Multiparty unconditionally secure protocols," in *Proc. 20th Annu. ACM Symp. Theory Comput.*, 1988, pp. 11–19.

[27] C. Crépeau and J. Kilian, "Achieving oblivious transfer using weakened security assumptions," in *Proc. 29th Symp. Foundations Computer Sci.*, 1988, pp. 42–52.

[28] J. Kilian, "Founding cryptography on oblivious transfer," in *Proc. 20th Annu. ACM Symp. Theory Comput.*, 1988, pp. 20–31.

[29] A. C. Nascimento and A. Winter, "On the oblivious-transfer capacity of noisy resources," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2572–2581, Jun. 2008.

[30] R. Ahlswede and I. Csiszár, "On oblivious transfer capacity," in *Information Theory, Combinatorics, Search Theory*. Berlin, Germany: Springer-Verlag, 2013, pp. 145–166.

[31] M. Mishra, B. K. Dey, V. M. Prabhakaran, and S. Diggavi, "The oblivious transfer capacity of the wiretapped binary erasure channel," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2014, pp. 1539–1543.

[32] M. Mishra, B. K. Dey, V. M. Prabhakaran, and S. Diggavi, "On the oblivious transfer capacity region of the binary erasure broadcast channel," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2014, pp. 237–241.

[33] A. D. Wyner, "The wire-tap channel," *Bell Syst. Techn. J.*, vol. 54, no. 8, pp. 1355–1387, 1975.

[34] I. Csiszár and J. Körner, "Broadcast channels with confidential messages," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 339–348, Jun. 1978.

[35] Y. Liang, H. V. Poor, and S. Shamai, "Information theoretic security," *Foundations Trends Commun. Inf. Theory*, vol. 5, no. 4–5, pp. 355–580, 2009.

[36] I. Csiszár and P. Narayan, "Secrecy capacities for multiterminal channel models," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2437–2452, Jun. 2008.

[37] L. Lai, H. E. Gamal, and H. Poor, "The wiretap channel with feedback: Encryption over the channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 5059–5067, Nov. 2008.

[38] E. Ardestanizadeh, M. Franceschetti, T. Javidi, and Y.-H. Kim, "Wiretap channel with secure rate-limited feedback," *IEEE Trans. Inf. Theory*, vol. 55, no. 12, pp. 5353–5361, Dec. 2009.

[39] H. D. Ly, T. Liu, and Y. Liang, "Multiple-input multiple-output Gaussian broadcast channels with common and confidential messages," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5477–5487, Nov. 2010.

[40] E. Perron, "Information-theoretic Secrecy for Wireless Networks," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2009.

[41] E. Perron, S. N. Diggavi, and I. E. Telatar, "On cooperative secrecy for discrete memoryless relay networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2010, pp. 2573–2577.

[42] E. Perron, S. N. Diggavi, and I. E. Telatar, "On cooperative wireless network secrecy," in *IEEE INFOCOM*, Apr. 2009, pp. 1935–1943.

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

26th IEEE Int. Conf. Comput. Commun. (INFOCOM 2007), 2007, pp. 616–624.

[44] T. Cui, T. Ho, and J. Kliewer, "On secure network coding with nonuniform or restricted wiretap sets," *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 166–176, Jan. 2013.

[45] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. Mandayam, "Information-theoretically secret key generation for fading wireless channels," *IEEE Trans. Inf. Forens. Security*, vol. 5, pp. 240–254, May 2010.

[46] S. Jana *et al.*, "On the effectiveness of secret key extraction from wireless signal strength in real environments," presented at the ACM MOBICOM Conf., 2009.

[47] Y. Abdallah, M. A. Latif, M. Youssef, A. Sultan, and H. E. Gamal, "Keys through arq: Theory and practice," *IEEE Trans. Inf. Forens. Security*, vol. 6, no. 3, pp. 737–751, Sep. 2011.

[48] A. Thangaraj, S. Dihidar, A. R. Calderbank, S. W. McLaughlin, and J. M. Merolla, "Applications of ldpc codes to the wiretap channel," *IEEE Trans. Inf. Theory*, vol. 53, no. 8, pp. 2933–2945, Aug. 2007.

[49] K. Argyraki *et al.*, "Creating secrets out of erasures," in *Proc. ACM Int. Conf. Mobile Comput. Netw. (MOBICOM)*, 2013, pp. 429–440.

[50] I. Safaka, C. Fragouli, K. Argyraki, and S. Diggavi, "Exchanging pairwise secrets efficiently," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM 2013)*, 2013, pp. 2265–2273.

[51] S. Keshav, *An Engineering Approach to Computer Networking: ATM Networks, the Internet, the Telephone Network*. New York, NY, USA: Addison-Wesley, 1997.

[52] S. Goldwasser and S. Micali, *Probabilistic Encryption*, vol. 28, 1984.

[53] M. Bellare, S. Tessaro, and A. Vardy, "Semantic security for the wiretap channel," in *Proc. Int. Cryptol. Conf. (CRYPTO)*, 2012, pp. 294–311, Springer.

ABOUT THE AUTHORS

Christina Fragouli (Senior Member, IEEE) received the B.S. degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1996, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of California (UCLA), Los Angeles, CA, USA.

She is a Professor at UCLA in the Electrical Engineering Department. She has worked at the Information Sciences Center, AT&T Labs, Florham Park, NJ, USA, and the National University of Athens. She also visited Bell Laboratories, Murray Hill, NJ, USA, and DIMACS, Rutgers University. Between 2006-2007, 2007-2012 and 2012-2015 she was an FNS Assistant Professor, an Assistant Professor, and an Associate Professor, respectively, in the School of Computer and Communication Sciences, EPFL, Switzerland.

She received the Fulbright Fellowship for her graduate studies, the Outstanding Ph.D. Student Award 2000-2001 from the Electrical Engineering Department, UCLA, the Zonta Award 2008 in Switzerland, the Starting Investigator ERC award in 2009, and several paper awards. She served as an Associate Editor for the IEEE Communications Letters, Elsevier's *Computer Communication*, the IEEE Transactions on Communications, the IEEE Transactions on Information Theory, and the IEEE Transactions on Mobile Communications.

Vinod M. Prabhakaran (Member, IEEE) received the M.E. degree from the Indian Institute of Science, Karnataka, India, in 2001, and the Ph.D. degree from the University of California, Berkeley, CA, USA, in 2007.

He was a Postdoctoral Researcher at the Coordinated Science Laboratory, University of Illinois, Urbana-Champaign from 2008 to 2010, and at Ecole Polytechnique Fédérale de Lausanne, Switzerland, in 2011. Since 2011, he has been a Reader at the School of Technology and Computer Science at the Tata Institute of Fundamental Research, Mumbai, India. His research interests include information theory, cryptography, wireless communication, and signal processing.

Dr. Prabhakaran received the Tong Leong Lim PreDoctoral Prize and the Demetri Angelakos Memorial Achievement Award from the EECS Department, University of California, Berkeley, and the Ramanujan Fellowship from the Department of Science and Technology, Government of India.

László Czap (Student Member, IEEE) received the M.Sc. degree in computer science from the Budapest University of Technology and Economics, Budapest, Hungary, in 2008, and the Ph.D. degree from École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2014.

He was a Researcher at the Laboratory of Cryptography and System Security (CrySys), Budapest University of Technology and Economics, from 2008 to 2010. His interests include communication and system security, information theory, coding and cryptography.

Suhas N. Diggavi (Fellow, IEEE) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Delhi, India, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1998.

After completing his Ph.D., he was a Principal Member Technical Staff in the Information Sciences Center, AT&T Shannon Laboratories, Florham Park, NJ, USA. After that, he was on the Faculty of the School of Computer and Communication Sciences, EPFL, where he directed the Laboratory for Information and Communication Systems (LICOS). He is currently a Professor in the Department of Electrical Engineering, at the University of California, Los Angeles, where he directs the Information Theory and Systems laboratory. His research interests include wireless network information theory, wireless networking systems, network data compression and network algorithms.

Dr. Diggavi is a Corecipient of the 2013 IEEE Information Theory Society & Communications Society Joint Paper Award, the 2013 ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) Best Paper Award, the 2006 IEEE Donald Fink prize paper award, 2005 IEEE Vehicular Technology Conference Best Paper Award and the Okawa Foundation Research. He has served on the editorial board for the IEEE Transactions on Information Theory, the ACM/IEEE Transactions on Networking, the IEEE Communication Letters, and was a Guest Editor for the IEEE Journal of Selected Topics in Signal Processing. He served as the Technical Program Cochair for the 2012 IEEE Information Theory Workshop (ITW) and the Technical Program Cochair for the 2015 IEEE International Symposium on Information Theory (ISIT).