Securing Broadcast Against Dishonest Receivers

László Czap
EPFL, Switzerland
Email: laszlo.czap@epfl.ch

Vinod M. Prabhakaran
TIFR, India
Email: vinodmp@tifr.res.in

Suhas Diggavi
UCLA, USA
Email: suhas@ee.ucla.edu

Christina Fragouli
EPFL, Switzerland
Email: christina.fragouli@epfl.ch

***Abstract*—Consider a sender, Alice, who wants to transmit private messages to two receivers, Bob and Calvin, using unreliable wireless broadcast transmissions and short public feedback from Bob and Calvin. In [1], we assumed that Bob and Calvin provide honest feedback, and characterized the secure capacity region of the private messages under the requirement that Bob and Calvin do not learn each other's message. In this paper, we assume that Bob (or Calvin) may provide dishonest feedback; or even control the input message distributions, as is commonly assumed in cryptography literature. We characterize the capacity region in the case of dishonest adversaries, as well as an achievable region for the case when the adversary has complete control on the distribution of the messages. We also design polynomial time protocols for both cases, that rely on the use of coding techniques to mix and secure the private messages. As a side result, we define an extended notion of semantic security for this problem and using a similar approach to [2], we show the equivalence of different security notions.**

I. Introduction

A promising application of network coding in wireless is when a wireless access point, Alice, wants to send private messages to two receivers, Bob and Calvin, who can send back to Alice packet acknowledgments. It is well known in the network coding literature that to achieve the optimal communication rates, Bob and Calvin should try to overhear the packets intended for the other user, while Alice should code across the private packets she has for Calvin and Bob [3], [4]. However, this rate-optimal scheme seems to come with a security compromise, since Bob and Calvin learn parts of each other's message.

In our work, we are interested in security guarantees we can provide in this setting. As a first step, in [1] we assumed that Calvin and Bob send honest acknowledgments, that correctly report what are the packets they have received, and we characterized what is the capacity region of secure communication in such a broadcast setting. That is, we assumed that Bob (or Calvin) is curious, but still honest – which is an optimistic assumption. In this paper, we look at the more realistic setting where Bob (or Calvin) are no longer honest, and try to deceive Alice on what are the packets he correctly received. Moreover, following security definitions in the cryptography literature, we also look at the case where Bob (or Calvin) may even control the input message distributions that Alice sends.

For the case of a dishonest user, we provide a complete capacity characterization, as well as a polynomial time achievability algorithm that leverages coding techniques. Our achievability protocol uses coding and feedback to exploit three aspects of wireless: (i) Alice can broadcast; (ii) Bob and Calvin will not receive exactly the same broadcast packets due to channel errors and (iii) each of them passively collects information about the other's (encrypted) message. Clearly, if a user dishonestly reports what are the messages he has correctly received, we cannot offer guarantee on the message rates he will experience; our guarantees are for the honest receiver. Interestingly, we find that the achievable rate of secure communication to the honest receiver is *not affected* by the dishonest acknowledging of the other receiver.

We also define a stronger notion of security and design a scheme that is secure independently of the joint distribution of the messages, *i.e.,* the adversary might even choose this distribution arbitrarily. Building on the approach of [2] we show equivalence of our security definitions with other notions of security that are used more commonly in the realm of cryptography.

Related work: Secure transmission of messages using noisy channel properties was pioneered by Wyner [5], 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 [6], [7]. 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 [8] and further developed by Ahlswede-Csiszár [9] – but secure key agreement is not the same as secure transmission of *specific* messages. The wiretap channel with secure feedback and its variants for message security have been studied in [10], [11]. The use of feedback and broadcast for private message transmission, *without* security requirements has been studied in [3], [4].

Closest to the current work are [12] and [1]. In [12] the secret message capacity of a single receiver broadcast channel against a passive eavesdropper was established. As mentioned earlier, [1] investigates a similar setting but with a much more restrictive adversary model.

II. Problem formulation and system model

We consider a three party communication setting with one sender (Alice) and two receivers (Bob and Calvin). The goal

This work was supported by the ERC Starting Grant Project NOWIRE ERC-2009-StG-240317. Vinod M. Prabhakaran's research was supported in part by a Ramanujan Fellowship from the Department of Science and Technology, Government of India. The work of S. Diggavi was supported in part by NSF award 1136174 and MURI award AFOSR FA9550-09-064.

of Alice is to securely send private messages W_1 and W_2 to Bob and Calvin, such that the receivers may not learn each other's messages.

Alice employs a memoryless erasure broadcast channel defined as follows. The inputs of the channel are length L vectors over $\mathbb{F}_q$, which we call sometimes packets. The ith input is denoted by X_i. The ith output of the channel seen by Bob is $Y_{1,i}$, while the output seen by Calvin is $Y_{2,i}$. The broadcast channel consists of two independent erasure channels towards Bob and Calvin. We note that our assumption on independence eases presentation, but it is not crucial in deriving our results. We denote δ_1 the erasure probability of Bob's channel and δ_2 that of Calvin's channel. More precisely,

$$\Pr\{Y_{1,i}, Y_{2,i}|X_i\} = \Pr\{Y_{1,i}|X_i\}\Pr\{Y_{2,i}|X_i\},$$

$$\Pr\{Y_{k,i}|X_i\} = \begin{cases} 1-\delta_k, & Y_{k,i} = X_i \\ \delta_k, & Y_i = \perp, \end{cases}, \; k \in \{1,2\}$$

where $\perp$ is the symbol of an erasure.

Assumptions: We assume that the receivers send public acknowledgments after each transmission stating whether or not they received the transmission correctly. By *public* we mean that the acknowledgments are available not only for Alice but for the other receiver as well.

We assume that some authentication method prevents the receivers from forging each other's acknowledgments. Also, we assume that *both* Bob and Calvin only know each other's acknowledgments causally, after they have revealed their own.

Let S_i denote the state of the channel in the ith transmission, $S_i \in \{B, C, BC, \emptyset\}$ corresponding to the receptions "Bob only", "Calvin only", "Both" and "None", respectively. Further, S_i^* denotes the state based on the acknowledgments sent by Bob and Calvin. If both users report honestly, then $S_i = S_i^*$. We denote as S^i the vector that collects all the states up to the ith, *i.e.,* $S^i = [S_1 \dots S_i]$, and similarly for S^{*i}.

Beside the communication capability as described above, all users can securely generate private randomness. We denote by Θ_A, Θ_B and Θ_C the private random strings Alice, Bob, and Calvin, respectively have access to. All parties have perfect knowledge of the communication model.

A. Security and reliability requirements

An (n, ϵ, N_1, N_2) scheme sends N_1 packets of length L to Bob and N_2 to Calvin using n transmissions from Alice with error probability smaller than ϵ. Formally:

Definition 1. *An* (n, ϵ, N_1, N_2) scheme *for the two user message transmission problem consists of the following components: (a) message alphabets $\mathcal{W}_1 = \mathbb{F}_q^{LN_1}$ and $\mathcal{W}_2 = \mathbb{F}_q^{LN_2}$, (b) encoding maps $f_i(.)$, $i = 1, 2, \dots, n$, and (c) decoding maps $\phi_1(.)$ and $\phi_2(.)$, such that if the inputs to the channel are*

$$X_i = f_i(W_1, W_2, \Theta_A, S^{*i-1}), \quad i = 1, 2, \dots, n, \tag{1}$$

where $W_1 \in \mathcal{W}_1$ and $W_2 \in \mathcal{W}_2$ are arbitrary messages in their respective alphabets and Θ_A is the private randomness Alice has access to, then, provided the receivers acknowledge honestly, their estimates after decoding $\hat{W}_1 = \phi_1(Y_1^n)$ and $\hat{W}_2 = \phi_2(Y_2^n)$ satisfy

$$\Pr\{\hat{W}_1 \neq W_1\} < \epsilon, \; and \tag{2}$$

$$\Pr\{\hat{W}_2 \neq W_2\} < \epsilon. \tag{3}$$

Dishonest user: We will say that a user is *dishonest* if the user can (*a*) select the marginal distribution of the other user's message arbitrarily; his own message is assumed to be independent of the other user's message and uniformly distributed over its alphabet and the dishonest user does not have (*a priori*) access to his own message, and (*b*) produce dishonest acknowledgments as a (potentially randomized) function of all the information he has access to when producing each acknowledgment (this includes all the packets and the pattern of erasures he received up to and including the current packet he is acknowledging and the acknowledgments sent by the other user over the public channel up to the previous packet). In the following σ denotes the dishonest user's acknowledging strategy.

Note that at most one of the receivers can be dishonest. Indeed, if a user is dishonest, we cannot guarantee that his private messages will remain secure from the other user; thus if both users are dishonest the problem is not meaningful.

It is common to define the *advantage* of the adversary (in our case a dishonest user), which measures the gain that the adversary obtains by observing a protocol. We express the adversarial advantage $\mathbf{Adv}^{\text{mis}}$ in terms of mutual information (mis = mutual information security). We discuss the relation between different security definitions in Section II-B.

Definition 2. *An (n, ϵ, N_1, N_2) scheme is said to be* secure against a dishonest user*, if it guarantees decodability and security for an honest user even if the other user is dishonest (as defined above). That is, if Bob is honest, then (2) and*

$$\mathbf{Adv}^{\text{mis}} = \max_{P_{W_1}, \sigma} I(W_1; Y_2^n S^n \Theta_C) < \epsilon \tag{4}$$

are satisfied, and if Calvin is honest, then (3) and

$$\mathbf{Adv}^{\text{mis}} = \max_{P_{W_2}, \sigma} I(W_2; Y_1^n S^n \Theta_B) < \epsilon. \tag{5}$$

are satisfied. The maxima are taken over all adversarial acknowledging strategies and all possible distributions P_{W_1} or P_{W_2} of the corresponding message.

The secret message capacity region $\mathcal{R} \subset \mathbb{R}_+^2$ *is the set of all rate pairs (R_1, R_2), such that for every $\epsilon, \epsilon' > 0$ there are N_1 and N_2 and a large enough n for which there exists an (n, ϵ, N_1, N_2) scheme that is secure against a dishonest user and*

$$R_1 - \epsilon' < \frac{1}{n} N_1 L \log q, \quad R_2 - \epsilon' < \frac{1}{n} N_2 L \log q. \tag{6}$$

Clearly a scheme which is secure against a dishonest user is also secure against honest (but curious) users since the dishonest user may choose to acknowledge truthfully.

When defining security against a dishonest user (Definition 2), we assumed that the dishonest user cannot control his own message distribution. Relaxing this assumption leads to a stronger notion of security. We define the adversarial advantage $\mathbf{Adv}^{\text{mis}}_{\text{dis}}$ (dis = distribution independent security) for this case.

Definition 3. *An* (n, ϵ, N_1, N_2) *scheme is said to provide* distribution independent security*, if it guarantees decodability and security for the honestly acknowledging user (or users) independently of the joint distribution* P_{W_1,W_2} *of* (W_1, W_2)*. That is, if Bob is honest, (2) and*

$$\mathbf{Adv}^{\text{mis}}_{\text{dis}} = \max_{P_{W_1,W_2},\sigma} I(W_1; Y_2^n S^n \Theta_C | W_2) < \epsilon \tag{7}$$

are satisfied, and if Calvin is honest, then (3) and

$$\mathbf{Adv}^{\text{mis}}_{\text{dis}} = \max_{P_{W_1,W_2},\sigma} I(W_2; Y_1^n S^n \Theta_B | W_1) < \epsilon. \tag{8}$$

are satisfied.

A rate pair (R_1, R_2) *belongs to the rate region* $\mathcal{R}^{\text{dis}}$ *if for every* $\epsilon, \epsilon' > 0$ *there are* N_1 *and* N_2 *and a large enough* n *for which there exists an* (n, ϵ, N_1, N_2) *scheme that provides distribution independent security and*

$$R_1 - \epsilon' < \frac{1}{n} N_1 L \log q, \quad R_2 - \epsilon' < \frac{1}{n} N_2 L \log q. \tag{9}$$

B. Security notions

We formulate our results in information theoretic terms, defining secrecy as a mutual information term being negligibly small. In the realm of cryptography it is more common to prove security of an encryption scheme by showing distinguishing security or semantic security. To facilitate the interpretation of our results and to allow a fair comparison with other schemes, we cite a recent result from [2], which shows equivalence between the two approaches. By this, our definition of security against a dishonest user is equivalent to semantic security. We also extend the notion of semantic security such that it handles joint message distributions, which results in a definition matching distribution independent security. We will give the definitions for Bob's security, the security for Calvin is completely symmetric.

The notion of semantic security captures the intuition that an adversary should not learn anything useful about the message. In other words, the probability that the adversary can compute a function f of the message should not increase significantly after observing the protocol compared to the *a priori* probability of a correct guess. The semantic security advantage is defined as

$$\mathbf{Adv}^{\text{ss}} = \max_{f,P_{W_1},\sigma} \Big\{ \max_{\mathcal{A}} \Pr\{\mathcal{A}(Y_2^n, S^n, \Theta_C, \sigma) = f(W_1)\} - \max_{\mathcal{S}} \Pr\{\mathcal{S}(P_{W_1}, f) = f(W_1)\} \Big\},$$

where f is any function of W_1, $\mathcal{A}$ is any function the adversary may compute after observing the protocol and $\mathcal{S}$ is a simulator trying to compute f without accessing the protocol output. Here also, W_2 is uniformly distributed and independent of W_1. The term simulator to denote guessing functions comes from the intuition that ideally there exists an algorithm (simulator) that simulates the run of a protocol without having access to the message and whose output is indistinguishable from the output of a real protocol. Theorems 1, 5 and 8 from [2] prove the following inequalities:

$$\mathbf{Adv}^{\text{ss}} \leq \sqrt{2 \cdot \mathbf{Adv}^{\text{mis}}}; \ \mathbf{Adv}^{\text{mis}} \leq 4 \cdot \mathbf{Adv}^{\text{ss}} \log\left(\frac{2^n}{\mathbf{Adv}^{\text{ss}}}\right)$$

This result shows that requirement (4) is naturally equivalent to semantic security. *i.e.*, a small ϵ in (4) implies that $\mathbf{Adv}^{\text{ss}}$ is also small.

The above discussion assumed that Calvin cannot choose the distribution of his own message W_2. We now extend the above definition of semantic security such that it does not rely on the distribution of W_2, which results a stronger notion of security. We define the adversarial advantage for this case as

$$\mathbf{Adv}^{\text{ss}}_{\text{dis}} = \max_{f,P_{W_1,W_2},\sigma} \Big\{ \max_{\mathcal{A}} \Pr\{\mathcal{A}(Y_2^n, S^n, \Theta_C, \sigma, W_2) = f(W_1, W_2)\} - \max_{\mathcal{S}} \Pr\{\mathcal{S}(P_{W_1,W_2}, f, W_2) = f(W_1, W_2)\} \Big\}. \tag{10}$$

Note that here we allow the simulator to have access to the message W_2 which an honest Calvin will learn. We show the following lemma which implies that requirement (7) is equivalent to this extended notion of semantic security. Due to space constraints we give the proof in the extended version of this paper [13].

Lemma 1.

$$\mathbf{Adv}^{\text{ss}}_{\text{dis}} \leq \sqrt{2 \cdot \mathbf{Adv}^{\text{mis}}_{\text{diss}}}$$
$$\mathbf{Adv}^{\text{mis}}_{\text{dis}} \leq 4 \cdot \mathbf{Adv}^{\text{ss}}_{\text{diss}} \log\left(\frac{2^n}{\mathbf{Adv}^{\text{ss}}_{\text{diss}}}\right).$$

This lemma suggests that our results on mutual information security (see Theorems 1-2 in the next section) also a characterize the rate region for semantic security.

III. Main result

Theorem 1. *The secret message capacity region as defined in Definition 2 is the set of all rate pairs* $(R_1, R_2) \in \mathbb{R}^2_+$ *which satisfy the following two inequalities:*

$$\frac{R_1(1-\delta_2)}{\delta_2(1-\delta_1)(1-\delta_1\delta_2)} + \frac{R_1}{1-\delta_1} + \frac{R_2}{1-\delta_1\delta_2} \leq L \log q, \tag{11}$$

$$\frac{R_2(1-\delta_1)}{\delta_1(1-\delta_2)(1-\delta_1\delta_2)} + \frac{R_1}{1-\delta_1\delta_2} + \frac{R_2}{1-\delta_2} \leq L \log q. \tag{12}$$

We prove Theorem 1 in two steps. First, we provide a protocol in Section IV and show that this protocol achieves all the rate pairs in the capacity region. We then apply the converse proof developed for the weaker honest-but-curious security definition to get an upper bound. The two regions match, *i.e.,* a dishonest user cannot deteriorate the performance experienced by an honest user. The first term of (11) and (12) can be interpreted as the overhead for security, because – as we will see soon – it corresponds to the duration of a secret key generation phase. Omitting these terms gives us the capacity region for the message transmission problem with two users without any secrecy requirements [4].

In the case of distribution independent security we do not have a complete characterization: we construct a scheme that satisfies this stronger security definition, however its optimality is not clear. The next theorem gives the rate region achieved by our scheme.

Theorem 2. *If a rate pair* (R_1, R_2) *satisfies*

$$\frac{R_1(1-\delta_2)}{\delta_2(1-\delta_1)(1-\delta_1\delta_2)} + \frac{R_2(1-\delta_1)}{\delta_1(1-\delta_2)(1-\delta_1\delta_2)} + \frac{R_1}{1-\delta_1} + \frac{R_2}{1-\delta_1\delta_2} \leq L\log q, \quad (13)$$

$$\frac{R_1(1-\delta_2)}{\delta_2(1-\delta_1)(1-\delta_1\delta_2)} + \frac{R_2(1-\delta_1)}{\delta_1(1-\delta_2)(1-\delta_1\delta_2)} + \frac{R_1}{1-\delta_1\delta_2} + \frac{R_2}{1-\delta_2} \leq L\log q. \quad (14)$$

then $(R_1, R_2) \in \mathcal{R}^{\text{dis}}$.

IV. Protocol for dishonest receivers

We describe an (n, ϵ, N_1, N_2) scheme that is secure against a dishonest user as defined in Definition 2. In our new scheme we bring together ideas that secure message transmission in the presence of an adversary and ideas that allow efficient transmissions for multiple parties.

Main steps: We apply a two-phase approach introduced first in [12]. Alice attempts to send N_1 message packets $W_1 = (W_{1,1}, W_{1,2}, \ldots, W_{1,N_1})$ to Bob and $W_2 = (W_{2,1}, W_{2,2}, \ldots, W_{2,N_2})$ to Calvin using at most n packet transmissions.

I. *Key generation.* Alice sends uniform i.i.d. random packets over the channel. From the acknowedged packets, secret key packets between Alice-Bob and between Alice-Calvin are set up such that Bob's key is secret from Calvin and Calvin's key is secret from Bob. Privacy amplification [14], [15] is used to ensure security of the keys.

II. *Message encryption and transmission.* Alice encrypts the messages using the key packets and reliably transmits them to the two receivers. The encryption operation is a simple XOR with the encryption key packets, however the encryption key packets are not independent. Instead, they are produced from the secret key packets using a maximum distance separable (MDS) code. This allows efficient usage of the keys [12].

In both phases we rely on channel properties and exploit that neither receiver receives all transmitted packets. This allows both efficient key generation and efficient (in terms of key size) encryption. Previous work [12] has shown that it is sufficient to know the expected behavior of the channel, there is no need to know *exactly* which packets are received by an adversary.

To illustrate, consider the key generation phase. Assume that Alice transmits three random (independent and uniformly distributed) packets X_1, X_2, X_3, and assume Bob receives X_1, X_2, while Calvin receives X_2, X_3. If we could rely on Bob and Calvin's honesty, we could then assign $K_B = X_1$ as a secret key between Alice and Bob, while $K_C = X_3$ as the key between Alice and Calvin. If we cannot rely on Bob and Calvin's honesty, but we do know that Bob and Calvin have received at most one packet in common, we could allocate $K_B = X_1 \oplus X_2$ as the key between Alice-Bob, and $K_C = X_2 \oplus X_3$ as the Alice-Calvin key.

Note that although similar techniques are used in [1], [12], none of these previous schemes can handle an adversary who – being a legitimate receiver – has some control over the protocol run and who can also *actively* deviate from the protocol. To summarize, our new scheme has the following distinguishing features:

1) In the key generation phase the set of packets we use to compute the keys for Bob and for Calvin are not disjunct, still keys are secure.

2) Although Calvin can influence the run of the protocol, we ensure that independently of his acknowledging strategy, he cannot control how many times a given encrypted packet with Bob's message appears on the channel. From this property it follows that we can estimate accurately the number of packets Calvin overhears which makes it possible to use an encryption similar to [12].

3) In the second phase we need coding to make transmissions maximally useful for both users as seen in [4]. Alice can send an XOR-ed packet only if both receivers have a side information packet. However, a dishonest user might deny having a side information packet and hinder these coded transmissions. In our scheme we apply a round robin type scheduling to ensure that dishonest feedback cannot diminish the rate experienced by the other user.

A. Protocol description

Parameters: The operation of the protocol utilizes a set of parameters which we can directly calculate before the protocol starts, and whose use will be described in the following.

$$k_B = N_1\frac{1-\delta_2}{1-\delta_1\delta_2} + \left(N_1\frac{1-\delta_2}{1-\delta_1\delta_2}\right)^{3/4}, \quad (15)$$

$$k_C = N_2\frac{1-\delta_1}{1-\delta_1\delta_2} + \left(N_2\frac{1-\delta_1}{1-\delta_1\delta_2}\right)^{3/4}. \quad (16)$$

$$k_1 = \frac{k_B}{\delta_2} + \frac{1}{\delta_2}\left(\frac{2k_B}{\delta_2}\right)^{3/4}, \quad k_2 = \frac{k_C}{\delta_1} + \frac{1}{\delta_1}\left(\frac{2k_C}{\delta_1}\right)^{3/4}.$$

$$n_1 = \max\left(\frac{k_1}{1-\delta_1} + \left(\frac{k_1}{1-\delta_1}\right)^{3/4}, \frac{k_2}{1-\delta_2} + \left(\frac{k_2}{1-\delta_2}\right)^{3/4}\right) \quad (17)$$

$$n_2' = \frac{N_1}{1-\delta_1} + \frac{N_2}{1-\delta_1\delta_2} + \left(\frac{N_1}{1-\delta_1} + \frac{N_1}{1-\delta_1\delta_2}\right)^{3/4} \quad (18)$$

$$n_2'' = \frac{N_2}{1-\delta_2} + \frac{N_1}{1-\delta_1\delta_2} + \left(\frac{N_2}{1-\delta_2} + \frac{N_2}{1-\delta_1\delta_2}\right)^{3/4} \quad (19)$$

$$n = n_1 + \max\{n_2', n_2''\}. \quad (20)$$

Key Generation

1) Alice transmits n_1 packets $X_1, \ldots, X_{n_1}$. She generates these packets uniformly at random from $\mathbb{F}_q^L$ using her private randomness, and independently of W_1, W_2.
2) Bob and Calvin acknowledge which packets they have received. If Bob receives less than k_1 packets we declare a protocol error for him. Similarly for Calvin if he receives less than k_2 packets. When an error is declared for both users, the protocol terminates. If not, we continue with the user not in error, as if the user in error did not exist.
3) Let X_1^B be a $L \times k_1$ matrix that has as columns the first k_1 packets that Bob acknowledged. Alice and Bob create k_B secret key packets as $K_B = X_1^B G_{K_B}$, where G_{K_B} is a $(k_1 \times k_B)$ matrix and is a parity check matrix of a $(k_1, k_1 - k_B)$ MDS code. Similarly, using the first k_2 packets that Calvin acknowledges, Alice and Calvin create k_C secret key packets using the matrix G_{K_C}. Matrices G_{K_B}, G_{K_C} are publicly known and fixed in advance.

Message encryption and transmission
Encryption

4) Alice and Bob produce N_1 linear combinations of their k_B secret key packets as $K_B' = K_B G_{K_B'}$, where $G_{K_B'}$ is a $(k_B \times N_1)$ matrix and is a generator matrix of an (N_1, k_B) MDS code which is also publicly known. Similarly, Alice and Calvin create N_2 linear combinations of their k_C key packets.
5) Alice creates N_1 encrypted messages to send to Bob

$$U_{B,i} = W_{1,i} \oplus K'_{B,i}, \quad i = 1 \ldots N_1$$

where $\oplus$ is addition in the $\mathbb{F}_q^L$ vector space. Let U_B denote the set of $U_{B,i}, i = 1, \ldots, N_1$. She similarly produces a set U_C of N_2 encrypted messages to send to Calvin.

Encrypted transmissions

6) Alice sequentially takes the first encrypted packet from $U_{B,i}$, $i = 1 \ldots N_1$, that is not yet acknowledged by Bob and repeatedly transmits it, until there is one that only Calvin acknowledges. That is, if at time i Alice transmits $X_i = U_{B,j}$ for some $j < N_1$, then

$$X_{i+1} = \begin{cases} X_i, & \text{if } S_i^* = \emptyset \\ U_{B,j+1}, & \text{if } S_i^* \in \{B, BC\}. \end{cases} \tag{21}$$

Let Q_B denote the last transmitted packet of this step, *i.e.,* the first such packet that only Calvin acknowledges. If there is no such packet, Q_B is empty.
7) Similarly, Alice sends the first not-yet-acknowledged (by Calvin) encrypted packet from $U_{C,i}$, $i = 1 \ldots N_2$, until there is one that only Bob acknowledges. If at time i Alice transmits $X_i = U_{C,j}$ for some $j < N_2$, then

$$X_{i+1} = \begin{cases} X_i, & \text{if } S_i^* = \emptyset \\ U_{C,j+1}, & \text{if } S_i^* \in \{C, BC\}. \end{cases} \tag{22}$$

Let Q_C denote the first such packet that only Bob acknowledges. If there is no such packet, Q_C is empty.
8) Alice transmits the sum of the two undelivered packets: $Q_B \oplus Q_C$. If Q_B or Q_C is empty, then Alice sends the non-empty packet. If both Q_B and Q_C are empty, then both messages W_1 and W_2 are successfully delivered and we stop. If at time i Alice sends X_i, then

$$\begin{cases} \text{if } S_i^* = \emptyset, & \text{then } X_{i+1} = X_i \\ \text{if } S_i^* = B, & \text{then repeat steps 6 and 8.} \\ \text{if } S_i^* = C, & \text{then repeat steps 7 and 8.} \\ \text{if } S_i^* = BC, & \text{then repeat steps 6, 7 and 8.} \end{cases} \tag{23}$$

If at any point, the overall number of transmissions would exceed n as defined in (20) we stop and declare an error for the party (or parties) who has not acknowledged all his encrypted message packets.

B. Protocol analysis

We prove that the presented scheme is secure against a dishonest user as defined in Definition 2 and runs without error with high probability. We use lemmas whose proofs are provided in [13]. A simple calculation with the given parameters shows that it achieves any rate pair in the the region defined by (11)-(12).

1) Security: In our argument we focus on the secrecy of W_1 against a dishonest Calvin, but the same reasoning works for W_2 against a dishonest Bob as well.

To analyze the secrecy of W_1, we may, without loss of generality, assume that no error was declared for Bob during the key generation phase. Recall that an error is declared for Bob only if Bob fails to acknowledge at least k_1 packets. If an error was in fact declared for Bob, no information about Bob's message W_1 is ever transmitted by Alice. However, note that we do account for this error event when we analyze the probability of error for Bob (Section IV-B2).

We first show that $I(K_B; Y_2^{n_1} S^{n_1})$ can be made small, *i.e.,* the key generation phase is secure.

Lemma 2. *When Bob is honest and no error is declared for Bob in the key generation phase,*

$$I(K_B; Y_2^{n_1} S^{n_1}) \leq k_B e^{-c_1 \sqrt{k_1}} L \log q, \tag{24}$$

if $k_1 = \frac{k_B}{\delta_2} + \frac{1}{\delta_2}\left(\frac{2k_B}{\delta_2}\right)^{3/4}$ *and* $k_B \geq \frac{2}{\delta_2}$, *where* $c_1 > 0$ *is a constant. Further,* K_B *is uniformly distributed over its alphabet.*

The key facts we use in proving this lemma are (i) the number of packets seen by Calvin concentrates around its mean and (ii) an MDS parity check matrix can be used to perform privacy amplification in the packet erasure setting.

We still need to show that the secrecy condition is satisfied by the scheme even if Calvin chooses any message distribution P_{W_1} and applies any acknowledging strategy, *i.e.,* (4) holds. In the proof we omit taking the maximum, but the argument holds for any message distribution and any adversarial strategy, so the statement follows. We have

$$I(W_1; Y_2^n S^n \Theta_C) \leq I(W_1; Y_2^n | Y_2^{n_1} S^n \Theta_C U_C), \tag{25}$$

where the inequality used the fact that $\Theta_A, \Theta_C, W_2, S^n$ are independent of W_1 and we may express $Y_2^{n_1}, U_C$ as deterministic functions of $\Theta_A, \Theta_C, W_2, S^n$. Let M_B^C be the random variable which denotes the number of distinct packets of U_B that Calvin observes either in its pure form or in a form where the $U_{B,i}$ packet is added with some $U_{C,j}$ packet. We have the following two lemmas:

Lemma 3. $H(Y_2^n|Y_2^{n_1}S^n\Theta_CU_C) \le \mathbb{E}\left\{M_B^C\right\} L\log q$.

Lemma 4.

$$H(Y_2^n|W_1Y_2^{n_1}S^n\Theta_CU_C) \ge \mathbb{E}\left\{\min\left(k_B, M_B^C\right)\right\} L\log q \\ - I(K_B;Y_2^{n_1}S^{n_1})$$

Using these in (25), we have

$$I(W_1;Y_2^nS^n\Theta_C) \le \mathbb{E}\left\{\max\left(0, M_B^C - k_B\right)\right\} L\log q \quad (26)$$

$$+ I(K_B;Y_2^{n_1}S^{n_1}). \quad (27)$$

Lemma 2 gives a bound for the second term. We can bound the first term using concentration inequalities. Notice that the probability that Calvin overhears a packet $U_{B,i}$ (where we count overhearing in both pure form or as part of a linear combination), is $\frac{1-\delta_2}{1-\delta_1\delta_2}$ independently of Calvin's acknowledging strategy. Thus, M_B^C is a sum of N_1 independent random variables, and hence $\mathbb{E}\left\{M_B^C\right\} = N_1\frac{1-\delta_2}{1-\delta_1\delta_2}$. Since $k_B = N_1\frac{1-\delta_2}{1-\delta_1\delta_2} + \left(N_1\frac{1-\delta_2}{1-\delta_1\delta_2}\right)^{3/4}$, by applying Chernoff-Hoeffding bound we have

$$\mathbb{E}\left\{\max\left(0, M_B^C - k_B\right)\right\} \le N_1 \Pr\left\{M_B^C > k_B\right\} \le N_1e^{-c_2\sqrt{N_1}},$$

for a constant $c_2 > 0$. Substituting this together with Lemma 2 in (27) we get

$$I(W_1;Y_2^nS^n\Theta_C) \le N_1e^{-c_2\sqrt{N_1}} + k_Be^{-c_2\sqrt{k_B}},$$

for constants $c_1, c_2 > 0$. By choosing a large enough value of N_1, we may meet (4).

2) Error probability: An error happens if (a) Bob receives less than k_1 packets in the first phase, or (b) he does not receive N_1 encrypted message packets in steps 6 and 8 before the protocol terminates. Both these error events have the same nature. An error happens if Bob collects significantly fewer packets than he is expected to receive in a particular step. We apply Chernoff-Hoeffding bound as we did to show the security guarantee proving that the probability of these events can be made arbitrarily small. We omit details to avoid parallel arguments.

3) Optimality: We can assume that both Bob and Calvin are honest and apply the converse proof developed in [1] (Theorem 4 in [1]). Obviously, this is a valid upper bound in the case of a dishonest user as well. With this we prove optimality and complete the proof of Theorem 1.

Theorem 3. *Any achievable rate pair* $(R_1, R_2) \in \mathcal{R}$ *as defined in Definition 2 satisfies inequalities* (11) *and* (12).

V. Distribution independent scheme

In the following we describe a scheme which satisfies the stronger security notion as defined in Definition 3. The protocol of Section IV cannot satisfy distribution independent security, because if Calvin knows his message *a priori*, then U_C carries information about the packets used in the key generation phase, hence potentially giving him extra information about Bob's key. We can overcome this issue if we modify the key generation phase and make sure that no packet used in generating Calvin's key contributes to Bob's key, thus U_C is conditionally independent of Bob's key given Calvin's observation of the protocol and W_2. This results in two separate key generation phases, one for Bob and one for Calvin.

Instead of sending n_1 key generation packets as defined in (17), we have a key generation of length $n_1^* + n_2^*$, where

$$n_1^* = \frac{k_1}{1-\delta_1} + \left(\frac{k_1}{1-\delta_1}\right)^{3/4}; \ n_2^* = \frac{k_2}{1-\delta_2} + \left(\frac{k_2}{1-\delta_2}\right)^{3/4}.$$

Bob's key is then computed from the first n_1^* packets, while Calvin's key is computed from the next n_2^* packets. All other parameters remain the same as in Section IV and the second phase remains unchanged too.

This scheme provides distribution independent security, which property is proved formally in [13]. A straightforward rate calculation completes the proof of Theorem 2.

References

[1] L. Czap, V. Prabhakaran, S. Diggavi, and C. Fragouli, "Broadcasting private messages securely," in *International Symposium on Information Theory (ISIT)*. IEEE, 2012, pp. 428–432.

[2] M. Bellare, S. Tessaro, and A. Vardy, "Semantic Security for the Wiretap Channel." in *International Cryptology Conference (CRYPTO)*. Springer, 2012, pp. 294–311.

[3] Y. Wu, P. Chou, and S. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," in *Conference on Information Sciences and Systems (CISS)*, 2005.

[4] L. Georgiadis and L. Tassiulas, "Broadcast erasure channel with feedback-capacity and algorithms," in *Workshop on Network Coding, Theory, and Applications, (NetCod)*. IEEE, 2009, pp. 54–61.

[5] A. D. Wyner, "The wire-tap channel," *The Bell system Technical Journal*, vol. 54, no. 8, pp. 1355–1387, 1975.

[6] I. Csiszár and J. Körner, "Broadcast channels with confidential messages," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 339–348, 1978.

[7] Y. Liang, H. V. Poor, and S. Shamai, "Information Theoretic Security." *Foundations and Trends in Communications and Information Theory*, vol. 5, no. 4-5, pp. 355–580, 2009.

[8] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, May 1993.

[9] R. Ahlswede and I. Csiszár, "Common randomness in information theory and cryptography - I: Secret sharing," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1121–1132, 1993.

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

[11] E. Ardestanizadeh, M. Franceschetti, T. Javidi, and Y. Kim, "Wiretap channel with secure rate-limited feedback," *IEEE Transactions on Information Theory*, vol. 55, no. 12, pp. 5353–5361, 2009.

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

[13] L. Czap, V. M. Prabhakaran, S. Diggavi, and C. Fragouli, "Securing Broadcast Against Dishonest Receivers," Tech. Rep. [Online]. Available: http://arni.epfl.ch/~czap/netcod13.pdf

[14] U. Maurer and S. Wolf, "Privacy amplification secure against active adversaries," *Advances in Cryptology (CRYPTO)*, pp. 307–321, 1997.

[15] M. Jafari Siavoshani, S. Diggavi, C. Fragouli, U. K. Pulleti, and K. Argyraki, "Group Secret Key Generation over Broadcast Erasure Channels," in *Asilomar Conference on Signals, Systems, and Computers*, 2010, pp. 719–723.