Secure Capacity Region for Erasure Broadcast Channels with Feedback

László Czap  Vinod M. Prabhakaran  Suhas Diggavi  Christina Fragouli
EPFL, TIFR, UCLA and EPFL

Abstract. We formulate and study a cryptographic problem relevant to wireless: a sender, Alice, wants to transmit private messages to two receivers, Bob and Calvin, using unreliable wireless broadcast transmissions and short public feedback from Bob and Calvin. We ask, at what rates can we broadcast the private messages if we also provide (information-theoretic) unconditional security guarantees that Bob and Calvin do not learn each other’s message? We characterize the largest transmission rates to the two receivers, for any protocol that provides unconditional security guarantees. We design a protocol that operates at any rate-pair within the above region, uses very simple interactions and operations, and is robust to misbehaving users.

1 Introduction

Wireless bandwidth is scarce – to efficiently use it, we need to mix private messages intended for different users – and this makes securing such channels hard. Consider the situation where a wireless access point, Alice, wants to send private messages to two receivers, Bob and Calvin. To do so, Alice can only use the wireless channel, where each packet transmission is broadcast and subject to errors. A simple strategy is for Alice to keep retransmitting each packet until it is acknowledged by the intended receiver. But as Alice repeatedly broadcasts a packet intended for Bob, Calvin may overhear it. In fact, recent work has established that Calvin should try to overhear the packets intended for Bob, while Alice should code across the private packets she has for Calvin and Bob, as this can significantly increase the communication rates both receivers experience [1,2,3,4,5]. Fig. 1 illustrates such an example. In the wireless community, the need for bandwidth efficiency is acutely perceived, and there is significant effort in developing and deploying such schemes that rely on opportunistic overhearing and mixing of private messages [6,7,8]. However, the gain in efficiency seems to come with a security compromise, since Bob and Calvin learn parts of each other’s message. This leads us to ask a new question.

Question: In a wireless broadcasting setting, can we characterize the optimal unconditional (information-theoretic) secure transmission rates for conveying private messages to Bob and Calvin?

In this paper we answer this question when Alice can use a broadcast erasure channel, while Bob and Calvin can send (public, reliable, and authenticated) packet acknowledgments. That is, each transmission of Alice is either perfectly received or completely lost by Bob and Calvin independently from each other, and they can causally acknowledge this fact. Channels that perfectly erase packets do not exist in nature; even if a packet is corrupted by noise it would still be possible to extract some information from it. However, recent experimental results on wireless testbeds show that one can create almost perfect erasures through careful insertion of interference and appropriate coding [9,10]. This mechanism also enables erasure channels with known erasure probabilities. Furthermore, the results for erasure channels can serve as building blocks for noisy wireless channels [11,12,13].

If we do not insist on information theoretic security guarantees, there exist today methods to answer this question; our work as far as we know is the first to examine whether it is possible to provide unconditional guarantees and at what rates.

Our contributions: We propose a low-complexity two-phase protocol, which first efficiently generates secure keys and then judiciously uses them for encryption exploiting the wireless broadcast properties. We prove our protocol is optimal by showing a matching impossibility result: we show that no other scheme can achieve

1 To formalize this question we need to specify (i) what is a good model for the noisy broadcast channel? (ii) what kind of feedback is feasible? (iii) what is the notion of security that we seek? (iv) what is the power of the adversary? (v) what is the measure of transmission efficiency? We formally discuss these aspects in Section 2.
Fig. 1. An example where mixing private messages increases the transmission efficiency. Alice wants to send packet B to Bob and C to Calvin; symbol X indicates that a transmitted packet is not successfully received due to corruption by the channel. Alice first broadcasts message B; Calvin successfully receives it, while Bob fails to do so. Next, Alice transmits message C; Bob successfully receives it, while Calvin fails. Alice can take advantage of this side information Bob and Calvin have, and transmit the message B+C. This transmission is maximally useful for both receivers: assume both receive it, Bob, since he knows C, he can retrieve B, and Calvin, since he knows B, he can retrieve C. Thus the protocol concludes in three transmissions. In contrast, if we did not use coding across the private messages, we would need at least four transmissions, yielding 33% reduction in efficiency.

better secure private transmission rates to Bob and Calvin. To the best of our knowledge, this is the first result on information-theoretic security where the optimal strategy has a natural need for a two-phase protocol for secure message transmission. Our impossibility result also introduces new information-theoretic techniques that utilize a balance between generated and consumed keys, although the bounds are true for any valid security protocol which does not constrain it to be a two-phase scheme.

Our protocol is based on the following ideas. First, Alice-Bob and Alice-Calvin may create unconditionally secure pairwise secret keys $K_B$ and $K_C$ respectively, using a fundamental observation of Maurer [14]: different receivers have different looks on the transmitted signals, and we can build on these differences with the help of feedback to create secret keys [9,10]. For example, if Alice transmits random packets through independent erasure channels with erasure probability 0.5, there would be a good fraction of them (approximately 25%) that only Bob receives, and we can transform this common randomness between Alice and Bob to a key $K_B$ using privacy amplification [14,9,10,15]. A novel aspect of our protocol is that the secure keys for both Bob and Calvin are generated simultaneously using the same sequence of transmissions by Alice, thus optimally utilizing wireless broadcasting.

A naive approach is to generate the secret keys $K_B, K_C$ with the same size as the respective private messages and use them as one-time pads. This is too pessimistic in our case: Calvin is only going to receive a fraction of the packets intended for Bob and thus we only need to create an amount of key that allows us to protect against this fraction. To build on this observation, feedback is useful; knowing which packets Bob has successfully received (or not), allows us to decide what to transmit next, so that we preserve as much secrecy from Calvin as possible; and symmetrically for Calvin. In the second phase of our protocol, we combine these ideas with a network coding strategy [16,17,11] that makes transmissions maximally useful to both Bob and Calvin. Fig. 2 shows the benefits of our approach (which achieves the secret message capacity) compared to the naive scheme.

Our protocol does not rely on both users operating honestly: even if we assume that Calvin misbehaves, for example by sending fake acknowledgments (see Section 4), we can still provide the same security guarantees and operational rate to the correctly behaving Bob.

Related work: Secure transmission of messages using noisy channel properties was pioneered by Wyner [18], 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 [19,20]. Notably, when the eavesdropper and legitimate channel are statistically identical, then the wiretap framework yields no security.

In fact this can be done using linear combinations of the received packets, thereby allowing for a complexity that is polynomial in the number of transmitted packets [9,15].
The fact that feedback can give security even in this case was first observed for secret key agreement by Maurer [24] and further developed by Ahlswede-Csiszár [21] – 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 [22,23]; 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 [24], where some conclusive results have been established. As mentioned earlier, the use of feedback and broadcast for private message transmission, without security requirements has been studied in [23]. We believe that ours are the first conclusive results that use insecure (and very limited) feedback for information-theoretic security of multiple private messages. We use linear code constructions that superficially seem similar to those in secret sharing [25], but our problem is different since we obtain secrecy for all erasure probabilities, by leveraging feedback. Another problem that is different but bears some similarities, is that of broadcast encryption [26,27], where a group of users receive a common secret message, and the issue is to deal with key management when users unsubscribe.

Outline: The rest of the paper is organized as follows. Section 2 describes the communication and security model, Section 3 gives our main result and a simple example, Section 4 formally describes our protocol, Section 5 contains the security analysis, Section 6 establishes optimality by proving an impossibility theorem and Section 7 concludes by discussing several possible extensions. Detailed proofs are provided in the Appendices.

2 Problem formulation and system model

We consider a three party communication setting with one sender (Alice) and two receivers (Bob and Calvin). The goal 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 $i$th input is denoted by $X_i$. The $i$th 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 denote $\delta_1$ the erasure probability of Bob’s channel and $\delta_2$ that of Calvin’s channel. More precisely,

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

$$
\Pr\{Y_{1,i} | X_i\} = \begin{cases} 1 - \delta_1, & Y_{1,i} = X_i \\ \delta_1, & Y_{1,i} = \perp \end{cases}
$$

and

$$
\Pr\{Y_{2,i} | X_i\} = \begin{cases} 1 - \delta_2, & Y_{2,i} = X_i \\ \delta_2, & Y_{2,i} = \perp \end{cases},
$$

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 acknowledgment causally, after they have revealed their own (we justify this in Section 7 when we discuss Denial-of-Service attacks).

Let $S_i$ denote the state of the channel in the $i$th 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 $i$th, i.e., $S^i = [S_1 \ldots S_i]$, and similarly for $S^{i*}$.

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

\footnote{Secret sharing would require that the number of erasures of the adversary is smaller than that of the legitimate receiver.}

\footnote{In a practical setting, we do not need to actually have a public error-free channel to send the acknowledgments: since these are very short packets, we can utilize a sufficiently strong error correcting code and send them through the noisy wireless channels.}
2.1 Security and reliability requirements

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

**Definition 1.** An \((n, \epsilon, N_1, N_2)\) scheme for the two user message transmission problem consists of the following components: (a) message alphabets \(W_1 = \mathbb{F}_q^{L,N_1}\) and \(W_2 = \mathbb{F}_q^{L,N_2}\), (b) encoding maps \(f_i(\cdot), i = 1, 2, \ldots, n\), and (c) decoding maps \(\phi_1(\cdot)\) and \(\phi_2(\cdot)\), such that if the inputs to the channel are

\[
X_i = f_i(W_1, W_2, \Theta_A, S^{i-1}), \quad i = 1, 2, \ldots, n, \tag{1}
\]

where \(W_1 \in W_1\) and \(W_2 \in W_2\) are arbitrary messages in their respective alphabets and \(\Theta_A\) is the private randomness Alice has access to, then, provided the receivers acknowledge honestly, their estimates after decoding \(\hat{W}_1 = \phi_1(Y^n_1)\) and \(\hat{W}_2 = \phi_2(Y^n_2)\) satisfy

\[
\Pr\{\hat{W}_1 \neq W_1\} < \epsilon, \quad \text{and} \quad \Pr\{\hat{W}_2 \neq W_2\} < \epsilon. \tag{2}
\]

**Definition 2.** An \((n, \epsilon, N_1, N_2)\) scheme is said to be secure against honest-but-curious users if in case both receivers are honest and the input messages \(W_1\) and \(W_2\) are independent random variables distributed uniformly over their respective alphabets, in addition to conditions (2)–(3) the following two conditions also hold:

\[
I(W_1; Y^n_1 S^n \Theta_C) < \epsilon \quad \tag{4}
\]

\[
I(W_2; Y^n_2 S^n \Theta_B) < \epsilon. \tag{5}
\]

**Malicious user:** We will say that a user is malicious 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 his alphabet and the malicious user does not have 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). We allow at most one user to be malicious.

**Definition 3.** An \((n, \epsilon, N_1, N_2)\) scheme is said to be secure against a malicious user, if in case one of the receivers is malicious (as defined above), the scheme guarantees for the other (honest) receiver decodability and security as in definitions 1 and 2. That is, if Calvin is the malicious user, (2) and (4) are satisfied for Bob, while if Bob is the malicious user, (3) and (5) are satisfied for Calvin.

Clearly a scheme which is secure against a malicious user is also secure against honest-but-curious users since the malicious user may choose the uniform distribution for the other user’s message and choose to acknowledge truthfully.

**Secret message capacity region:** The communication rate \(R_i\) towards receiver \(i\) expresses the number of message \(W_i\) bits successfully and securely delivered to receiver \(i\) per channel use\(^5\). We are interested in characterizing all rate pairs \((R_1, R_2)\) that our channel can support.

**Definition 4.** The rate pair \((R_1, R_2) \in \mathbb{R}_+^2\) is said to be achievable, if for every \(\epsilon, \epsilon' > 0\) there are \(N_1\) and \(N_2\) and a large enough \(n\) such that there exists an \((n, \epsilon, N_1, N_2)\) scheme that is secure against a malicious user and\(^6\)

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

The secret message capacity region \(\mathcal{R} \subset \mathbb{R}_+^2\) is defined as the set of all achievable rate pairs.

\(^5\) Channel use refers to Alice using the channel once to send one packet.

\(^6\) All logarithms in this paper are to the base 2 unless otherwise specified.
Fig. 2. Achieved rate region (in bits/packet) by the naive scheme in contrast to the optimal scheme and to the capacity region for the message sending problem with no security requirements. For this example $\delta_1 = 0.7$, $\delta_2 = 0.6$, $L = 1$, $q = 2$. Translating this region to bits/sec depends on how fast the source can send packets: for example, if we operate at the point $(R_1 = 0.1, R_2 = 0.2)$ and the transmission rate of 1Mbits/sec that IEEE802.11b supports, we would securely send $\approx 100$Kbits/sec to Bob and $\approx 200$Kbits/sec to Calvin.

3 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:

$$\frac{R_1(1 - \delta_2)}{\delta_2(1 - \delta_1)(1 - \delta_1 \delta_2)} + \frac{R_1}{1 - \delta_1} + \frac{R_2}{1 - \delta_1 \delta_2} \leq L \log q, \quad (7)$$

$$\frac{R_2(1 - \delta_1)}{\delta_1(1 - \delta_2)(1 - \delta_1 \delta_2)} + \frac{R_1}{1 - \delta_1} + \frac{R_2}{1 - \delta_2} \leq L \log q. \quad (8)$$

The first term of these inequalities 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 [2]. The difference between these two capacity regions (with and without secrecy requirements) is illustrated in Fig. 2 for some specific values of the parameters $\delta_1$, $\delta_2$, $L$ and $q$.

We prove Theorem 1 in two steps. First, we provide a protocol in Section 4 and prove in Section 5 that this protocol achieves all the rate pairs in the capacity region. The complexity of the scheme is discussed in Appendix A. Then we provide in Section 5 a proof to show that (7) and (8) are also bounds that are impossible to exceed by any protocol (a converse in information-theory parlance). Interestingly, our converse holds even for the capacity region defined using the weaker honest-but-curious security definition, i.e., a malicious user cannot deteriorate the performance experienced by an honest user. The following simple example illustrates the main ideas in our protocol.

### 3.1 A simplified example when both receivers are honest-but-curious

Alice wants to securely send the $N_1 = 3$ message packets $W_1 = [W_{1,1}, W_{1,2}, W_{1,3}]$ to Bob and the $N_2 = 3$ message packets $W_2 = [W_{2,1}, W_{2,2}, W_{3,2}]$ to Calvin. Both Bob and Calvin are honest and report back truthfully. The protocol proceeds as depicted in Table 1.
### Table 1. An example of a simplified protocol when both receivers are honest but curious.

| Alice sends | Bob’s Calvin’s | Bob’s key | Calvin’s key | Bob decoded | Calvin decoded |
|-------------|----------------|-----------|--------------|-------------|---------------|
| $X_1$ random | ✓              | $K_{B,1} = X_1$ |              |             |               |
| $X_2$ random | ✓              | $K_{B,1}$  |              |             |               |
| $X_3$ random | ×              |           | $K_{B,1}$  |             |               |
| $X_4$ random | ×              |           |              | $K_{C,1} = X_3$ |               |
| $X_5$ random | ✓              |           |              | $K_{C,1}, K_{C,2} = X_4$ |               |

**Key generation:** Alice transmits five random packets $X_1, \ldots, X_5$. At the end of this phase, Alice and Bob share the two secret key packets $K_{B,1} = X_1$ and $K_{B,2} = X_5$ that Bob received and Calvin did not. Similarly, Alice and Calvin share the secret key packets $K_{C,1} = X_3$ and $K_{C,2} = X_4$. The packet $X_2$ which was received by both Bob and Calvin is discarded.

**Message transmission for Bob:** We secure Bob’s messages with one-time pads and transmit them until either Bob or Calvin receive them; we start by sending $X_6 = W_{1,1} \oplus K_{B,1}$, where $\oplus$ denotes addition in $\mathbb{F}_q^L$.
- Since only Calvin receives $X_6$, we consider the key $K_{B,1}$ as consumed (Calvin observed a linear combination of it), and the message $W_{1,1}$ as undelivered (Bob did not receive anything). We will deal with undelivered messages at the last stage; at this point we proceed to send the new packet $X_7 = W_{1,2} \oplus K_{B,2}$.
- Since only Bob receives $X_7$, we consider the message $W_{1,2}$ as delivered (Bob can retrieve it from $X_7$) and the key $K_{B,2}$ as unconsumed (it remains secret from Calvin); we can reuse it to send $X_8 = W_{1,3} \oplus K_{B,2}$.
- Since both Calvin and Bob receive $X_8$, the message $W_{1,3}$ is delivered to Bob and the key $K_{B,2}$ is consumed. At the end of this phase Bob has received packets $W_{1,2}$ and $W_{1,3}$ and is missing packet $W_{1,1}$.

**Message transmission for Calvin:** We similarly make a first attempt to deliver Calvin’s message. We assume we are less successful than before, and although we consume both keys $K_{C,1}$ and $K_{C,2}$, we only deliver to Calvin $W_{2,1}$. Note that $X_{10}$ which is not received by any user is simply retransmitted.

**Message transmission for both:** To deliver the remaining messages $W_{1,1}, W_{2,2}$ and $W_{2,3}$, we take advantage of the fact that Bob already has $X_{11}$ and Calvin has $X_6$: we send $X_{13} = X_6 \oplus X_{11}$ that is maximally useful for both. Bob can recover $X_6$ and from this $W_{1,1}$, while Calvin can recover $X_{11}$ and from this $W_{2,2}$. Note that $X_{13}$ brings no information to Bob or Calvin for each other’s message.

**Important properties:** This simple scheme has the following properties.
- The number of key packets we set up and consume is smaller than the number of message packets we convey per user, because we can reuse certain keys that the adversary did not receive.
- At the last message transmission phase, we exploit side information users have for each other’s message to make a single transmission useful to both, without consuming any new key.

**Towards the general protocol:** If a node is dishonest he can send fake acknowledgments. Interestingly, we can rely on the expected behavior of the channel (and coding techniques) to have no performance loss for the honest user. For example, in the key generation phase, if we expect that Calvin will only receive one of the three
We now describe a $(n,ε,N_1,N_2)$ scheme that is secure against a malicious user as in Definition 3.

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

\[ k_B = N_1 \frac{1 - δ_2}{1 - δ_1 δ_2} + \left( N_1 \frac{1 - δ_2}{1 - δ_1 δ_2} \right)^{3/4}, \quad k_C = N_2 \frac{1 - δ_1}{1 - δ_1 δ_2} + \left( N_2 \frac{1 - δ_1}{1 - δ_1 δ_2} \right)^{3/4}. \]

\[ k_1 = \frac{k_B}{δ_2} + \frac{1}{δ_2} \left( \frac{2k_B}{δ_2} \right)^{3/4}, \quad k_2 = \frac{k_C}{δ_1} + \frac{1}{δ_1} \left( \frac{2k_C}{δ_1} \right)^{3/4}. \]

\[ n_1 = \max \left( \frac{k_1}{1 - δ_1} + \left( \frac{k_1}{1 - δ_1} \right)^{3/4}, \quad \frac{k_2}{1 - δ_2} + \left( \frac{k_2}{1 - δ_2} \right)^{3/4} \right). \]

\[ n_2 = \frac{N_1}{1 - δ_1 δ_2} + \left( \frac{N_1}{1 - δ_1 δ_2} \right)^{3/4}, \quad n_3 = \frac{N_2}{1 - δ_1 δ_2} + \left( \frac{N_2}{1 - δ_1 δ_2} \right)^{3/4}. \]

\[ n_4 = \max \left( \frac{N_1}{1 - δ_1} + \left( \frac{N_1}{1 - δ_1} \right)^{3/4} - n_2, \quad \frac{N_2}{1 - δ_2} + \left( \frac{N_2}{1 - δ_2} \right)^{3/4} - n_3 \right). \]

\[ n = n_1 + n_2 + n_3 + n_4. \]

**Main idea:** Using our protocol, 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. She either succeeds in sending $W_1$ to Bob or declares an error for Bob. Similarly, she either succeeds in sending $W_2$ to Calvin or declares an error for Calvin. We will argue in Section 5 that the failure probability can be made arbitrarily small. She proceeds in the following steps.

I. **Key generation.** Generation of $k_B$ shared secret key packets between Alice-Bob and $k_C$ secret packets between Alice-Calvin using $n_1$ transmissions. This step fails if we do not succeed to generate the required number of secret key packets.

II. **Message encryption and transmission.** She encrypts the messages using the produced keys and reliably transmits them to the two receivers. This step fails if we do not manage to deliver all the $N_1$ and $N_2$ message packets within the prescribed number of transmissions.

**Protocol Description**

**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.

packets Bob successfully receives, we can produce for Bob the keys $K_{B,1} = X_1 \oplus X_2$ and $K_{B,2} = X_2 \oplus X_3$ that are secure from Calvin no matter which packet he received. Similarly, when we send packets to Bob, assume we expect Calvin to receive two of these transmissions – but we do not know which two. We then create three linear combinations of Bob’s keys, say $K_{1,1} = K_{B,1}$, $K_{1,2} = K_{B,2}$, $K_{1,3} = K_{B,1} \oplus K_{B,2}$, and transmit $X_6 = W_{1,1} \oplus K_{1,1}$, $X_7 = W_{1,2} \oplus K_{1,2}$, and $X_8 = W_{1,3} \oplus K_{1,3}$ - no matter which two of these Calvin receives we are secure. Our protocol builds on these ideas.
3. Let $X^B_i$ 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 $K_B = X^B_1 G_{KB}$, where $G_{KB}$ is a $(k_1 \times k_B)$ matrix and is a parity check matrix of an $(k_1, k_1 - k_B)$ Maximum Distance Separable (MDS) code [28]. $K_B$ is a shared key set up between Alice and Bob, and consists of $k_B$ length $L$ packets. Similarly, using the first $k_2$ packets that Calvin acknowledges, Alice and Calvin create $k_C$ secret key packets. The MDS codes 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 keys as $K_B' = K_B G_{KB}'$, where $G_{KB}'$ 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 $k_C$ keys.

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 $F^2_q$ 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

$$U_{C,i} = W_{2,i} \oplus K_{C,i}', \quad i = 1 \ldots N_2$$

**Transmissions to Bob**

6. Alice sequentially takes each of the $U_{B,i}, i = 1 \ldots N_1$, encrypted packets and repeatedly transmits it, until it is acknowledged either by Bob or Calvin. That is, if at time $i$ Alice transmits $X_i = U_{B,j}$ for some $j < N_1$, then

$$X_{i+1} = \begin{cases} X_i, & \text{if } S^*_i = \emptyset \\ U_{B,j+1}, & \text{otherwise.} \end{cases} \quad (15)$$

7. (a) At the end of $n_2$ transmissions of $U_B$ packets, if Bob did not acknowledge $N_1(1 - \delta_1)/(1 - \delta_1 \delta_2)$ of them, a protocol error is declared for Bob and we move to step 8 below. If Calvin did not acknowledge $N_1(1 - \delta_2)/(1 - \delta_1 \delta_2)$ packets, an error is declared for Calvin and continue transmitting $U_B$ packets. (b) If all the $U_B$ packets are not exhausted before $n_2 + n_4$ transmissions, we declare a protocol error for Bob and proceed to the next step.

**Transmissions to Calvin**

8. Similarly, Alice takes each of the $U_{C,i}, i = 1 \ldots N_2$ packets and repeatedly transmits it, until it is acknowledged either by Bob or Calvin. That is, if at time $i$ we had $X_i = U_{C,j}$ for some $j < N_2$, then

$$X_{i+1} = \begin{cases} X_i, & \text{if } S^*_i = \emptyset \\ U_{C,j+1}, & \text{otherwise.} \end{cases} \quad (16)$$

9. (a) At the end of $n_3$ transmissions of $U_C$ packets, if Calvin did not acknowledge $N_3(1 - \delta_2)/(1 - \delta_1 \delta_2)$ of them, a protocol error is declared for Calvin and we move to step 10 below. If Bob did not acknowledge $N_3(1 - \delta_1)/(1 - \delta_1 \delta_2)$ packets, we declare an error for Bob and continue transmitting $U_C$ packets. (b) If all the $U_C$ packets are not exhausted before $n_3 + n_4$ transmissions, we declare a protocol error for Calvin and proceed to the next step.

**Transmissions to both**

10. At this step, Alice knows the following:
– At the end of step 6, there may exist some encrypted packets that are acknowledged only by Calvin and not by Bob. Assume we have $N'_{1}$ such packets, and denote them as $U_{B,i}^{'}$, $i = 1 \ldots N'_{1}$.
– Similarly, at the end of step 8, there may exist some $N'_{2}$ encrypted packets $U_{C,i}^{'}$, $i = 1 \ldots N'_{2}$, that only Bob has acknowledged and not Calvin.

Alice proceeds to transmit $\oplus$ combinations of $U_{B,i}^{'}$ and $U_{C,i}^{'}$ packets.
– She starts by transmitting $U_{B,1}^{'} \oplus U_{C,1}^{'}$.
– If at time $i$ Alice transmits $X_{i} = U_{B,j}^{'} \oplus U_{C,\ell}^{'}$ for some $j < N'_{1}$ and $\ell < N'_{2}$, then

$$X_{i+1} = \begin{cases} X_{i} & \text{if } S_{i}^{*} = \emptyset, \\ U_{B,j+1}^{'} \oplus U_{C,\ell}^{'} & \text{if } S_{i}^{*} = B, \\ X_{i+1} = U_{B,j}^{'} \oplus U_{C,\ell+1}^{'} & \text{if } S_{i}^{*} = C, \\ X_{i+1} = U_{B,j+1}^{'} \oplus U_{C,\ell+1}^{'} & \text{if } S_{i}^{*} = BC. \end{cases}$$

– If Alice first finishes all the $N'_{1}$ packets $U_{B,i}^{'}$, she continues by transmitting the remaining $N'_{2}$ packets until Calvin receives all of them.
– Similarly if she first finishes all the $U_{C,i}^{'}$ packets, she continues with the remaining $N'_{1}$ packets until Bob acknowledges them.

11. If steps 6 and 10 together exceed $n_{2} + n_{4}$ transmissions an error is declared for Bob. Similarly, if steps 8 and 10 take together exceed $n_{3} + n_{4}$ transmissions an error is declared for Calvin. In any case, the protocol is terminated when the channel has been used for $n$ times.

5 Analysis

**Theorem 2.** For any $\epsilon, \epsilon' > 0$ there exists a large enough $n$ for which the scheme described above is secure against a malicious user and achieves (in the sense of (6)) any rate pair in the region defined by (7)-(8).

**Proof.** Below, we prove that the above scheme is secure against a malicious user and runs without error with high probability. The rate assertion of the theorem follows from a simple numerical evaluation with the given parameter values.

5.1 Security

In our argument we focus on the secrecy of $W_{1}$ against a malicious Calvin, but the same reasoning works for $W_{2}$ against a malicious Bob as well. Since we do not intend to give security guarantees to a malicious user and consider at most one user to be malicious, we may assume that Bob is honest. Moreover, under our definition of malicious user, $W_{1}$ and $W_{2}$ are independent and the latter is uniformly distributed over its alphabet, but the distribution of $W_{1}$ is arbitrary and controlled by the malicious Calvin.

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 in the Section 5.2.

We first show that $I(K_{B}^{n}; Y_{2}^{n_{1}} S^{n_{1}})$ can be made small, i.e., the key generation phase is secure.

$^{7}$ More precisely, if $E_{1:B}$ is the indicator random variable for an error being declared for Bob in the key generation phase,

$$I(W_{1}; Y_{2}^{n}, S^{n}, \Theta_{C}) \leq I(W_{1}; Y_{2}^{n}, S^{n}, \Theta_{C}, E_{1:B}) = I(W_{1}; Y_{2}^{n}, S^{n}, \Theta_{C}|E_{1:B}) \leq I(W_{1}; Y_{2}^{n}, S^{n}, \Theta_{C}|E_{1:B} = 0).$$

To avoid clutter, we leave out the conditioning event in the rest of this subsection.
Lemma 1. When Bob is honest and no error is declared for Bob in the key generation phase,
\[ I(K_B; Y_2^{n_1} S^{n_1}) \leq k_B e^{-c_1 \sqrt{K_l} L \log q}, \] (17)
if \( k_1 = \frac{k_B}{\delta_2} + \frac{1}{\delta_2} \left( \frac{2k_B}{\delta_2} \right)^{3/2} \) and \( k_B \geq \frac{2}{\delta_2} \), where \( c_1 > 0 \) is some constant. Moreover, \( K_B \) 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 (4) is satisfied by the scheme even if Calvin controls the distribution of \( W_1 \). We have
\[ I(W_1; Y_2^n S^n \Theta C) \leq I(W_1; Y_2^n S^n \Theta C U_C) = I(W_1; Y_2^n | Y_2^{n_1} S^n \Theta C U_C), \] (18)
where the last equality 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 \( 1_{B,i} \) be the indicator random variable for the event that Calvin observes the packet \( U_{B,i} \) either in its pure form or in a form where the \( U_{B,i} \) packet is added with some \( U_{C,j} \) packet. Let \( M_{B,i}^c \) be the random variable which denotes the number of distinct packets of \( U_B \) that Calvin observes, so \( M_{B,i}^c = \sum_{i=1}^{N_1} 1_{B,i}^c \). We have the following two lemmas:

**Lemma 2.** \( H(Y_2^n | Y_2^{n_1} S^n \Theta C U_C) \leq \mathbb{E} \{ M_B^c \} L \log q. \)

**Lemma 3.** \( H(Y_2^n | W_1 Y_2^n S^n \Theta C U_C) \geq \mathbb{E} \{ \min (k_B, M_B^c) \} - I(K_B; Y_2^{n_1} S^{n_1}). \)

Using these in (18), we have
\[ I(W_1; Y_2^n S^n \Theta C) \leq \mathbb{E} \{ \max (0, M_B^c - k_B) \} L \log q + I(K_B; Y_2^{n_1} S^{n_1}). \] (19)

Lemma 1 gives a bound for the second term. We can bound the first term using concentration inequalities. In order to do this, let \( Z_{B,i} \) be the number of repetitions of a packet \( U_{B,i} \) that Alice makes until Bob acknowledges it (where we count both the transmission in pure form and in addition with some packet from \( U_C \)). Note that the random variables \( Z_{B,i} \) are independent of each other and have the same distribution. This follows from the fact that the \( S_i \) packet is i.i.d., and each \( S_i \) is independent of \( (Y_2^{n_1-1}, S^{n_1-1}, \Theta C) \). In other words, Calvin can exert no control over the channel state. Further, for the same reason, with every repetition the chance that Calvin obtains the transmission is \( 1 - \delta_2 \). This implies that the indicator random variables \( 1_{B,i}^c \) are i.i.d. with
\[ \Pr \{ 1_{B,i}^c = 1 \} = (1 - \delta_2) + \delta_1 \delta_2 (1 - \delta_2) + \ldots = \frac{1 - \delta_2}{1 - \delta_1 \delta_2}. \]

Notice that \( M_B^c \) is a sum of \( N_1 \) such independent random variables, and hence \( \mathbb{E} \{ M_B^c \} = N_1 \frac{1 - \delta_2}{1 - \delta_1 \delta_2} \). Since \( k_B = N_1 \frac{1 - \delta_2}{1 - \delta_1 \delta_2} + \left( N_1 \frac{1 - \delta_2}{1 - \delta_1 \delta_2} \right)^{3/2} \), by applying Chernoff-Hoeffding bound we have
\[ \mathbb{E} \{ \max (0, M_B^c - k_B) \} \leq N_1 \Pr \{ M_B^c > k_B \} \leq N_1 e^{-c_2 \sqrt{N_1}}, \] (20)
for a constant \( c_2 > 0 \). Substituting this together with Lemma 1 in (19) we get
\[ I(W_1; Y_2^n S^n \Theta C) \leq N_1 e^{-c_2 \sqrt{N_1}} + k_B e^{-c_2 \sqrt{N_1}}, \]
for constants \( c_1, c_2 > 0 \). By choosing\(^8\) a large enough value of \( N_1 \), we may meet (4).

\(^8\) Recall from \( \text{[4]} \) that by saying that we choose \( N_1 \) large enough we cause \( n \) to be large enough.
5.2 Error probability

We need to bound the probability that an error is declared for Bob\(^9\). An error happens if:

- Bob receives less than \(k_1\) packets in the first phase,
- he does not receive \(N_1(1 - \delta_1)/(1 - \delta_1 \delta_2)\) packets of \(U_B\) in step 7(a), or \(N(1 - \delta_2)/(1 - \delta_1 \delta_2)\) packets of \(U_C\) in step 9(a),
- he does not receive all the \(N_1\) packets of \(U_B\) (either in pure form or added with a packet in \(U_C\)) before step 11 intervenes.

All 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. The probability of these events can be bounded by applying the Chernoff-Hoeffding bound as we did to show the security guarantee (20). The sum of these bounds gives an upper bound on the overall error probability of the scheme, which in turn can be made smaller than \(\epsilon\) by choosing \(N_1\) large enough. A straightforward computation using the parameters in (9)-(14) shows that (6) is also satisfied.

\[\Box\]

6 Impossibility result (converse)

With Theorem 3 we complete the proof of Theorem 1. Throughout this section we will assume that both Bob and Calvin are honest. Obviously, an upper bound for this case is a valid upper bound in the case of a malicious user as well. Interestingly, we get the same bounds for the honest-but-curious users’ and for the malicious users’ case. Our proof relies on a few Lemmas which can be found together with their proofs in Appendix C.

Theorem 3. For the secret message capacity region as defined in Definition 4 it holds that:

\[
\frac{R_1(1 - \delta_2)}{\delta_2(1 - \delta_1)(1 - \delta_1 \delta_2)} + \frac{R_1}{1 - \delta_1} + \frac{R_2}{1 - \delta_1 \delta_2} \leq L \log q,
\]

\[
\frac{R_2(1 - \delta_1)}{\delta_1(1 - \delta_2)(1 - \delta_1 \delta_2)} + \frac{R_1}{1 - \delta_1} + \frac{R_2}{1 - \delta_2} \leq L \log q.
\]

Proof. We will prove the first inequality, the second follows from symmetry. We look at Alice’s transmissions from Bob’s perspective and express them in three terms (21a)-(21c) using elementary properties of entropy:

\[
nL \log q \geq nH(X_i) \geq \sum_{i=1}^{n} H(X_i|Y_{i-1}^{i-1}S^{i-1}) = \sum_{i=1}^{n} [H(X_i|Y_{i-1}^{i-1}Y_{2i-1}^{i-1}S^{i-1}) + I(X_i; Y_{2i-1}^{i-1}|Y_{i-1}^{i-1}S^{i-1})]
\]

\[
= \sum_{i=1}^{n} \left[ H(X_i|Y_{i-1}^{i-1}Y_{2i-1}^{i-1}S^{i-1}W_1) + I(X_i; W_1|Y_{i-1}^{i-1}Y_{2i-1}^{i-1}S^{i-1}) + I(X_i; Y_{2i-1}^{i-1}|Y_{i-1}^{i-1}S^{i-1}) \right].
\]

Lemmas 4-7 give lower bounds on each of the three terms in (21a); putting these together results in the stated inequality.

\[\Box\]

Intuition: We now informally interpret the terms in (21) with our protocol (of Section 4) in mind. Note however that the proof provides a general impossibility bound that holds for any scheme that satisfies Definition 4. The terms (21a)-(21c) classify the information Alice sends during the \(i\)th transmission. These terms can also be interpreted as balancing key generation and consumption for secrecy, as described below.

Term (21a) can be interpreted as anything that is not related to Bob’s message \(W_1\) (and has not been seen by either Bob or Calvin). This is lower bounded through Lemmas 6-7. For example, in our protocol this corresponds to a key generation attempt for either Bob or Calvin, or a new encrypted message for Calvin, \(i.e.,\) steps 1 and 8.

\(^9\) Note that, under our protocol, if no error is declared for Bob, he will be able to decode \(W_1\).
Term (21b) is interpreted as an encrypted packet that Alice tries to send to Bob, which has not already been received by Calvin. This is lower bounded in Lemma 4. In our protocol this occurs in transmissions directed towards Bob only in step 6.

Term (21c) brings information that Calvin has already seen during previous transmissions, but Bob has not seen. This is lower bounded in Lemma 5. In our protocol this would correspond to transmissions in step 10.

7 Extensions and discussion

We showed that it is possible to provide unconditional security guarantees while wirelessly broadcasting two private messages; we characterized all possible transmission rate pairs for the private messages, and showed we can achieve these using a simple protocol that efficiently generates and utilizes an appropriate amount of key. We conclude our paper with several natural extensions and some open questions.

Practical deployment: Our protocol has low complexity (see Appendix A) and does not require changing the physical layer transceivers of the three users; it is thus attractive for a potential system deployment. Although we only claim optimality under the modeling assumptions of Section 2, we believe such a system could enable operation at high secret message rates (for the parameters in Fig. 2, of the order of 100Kbits/sec per user), using channel conditioning techniques of [9,10].

Common message: Assume that besides the private messages, we also have a common message $W_c$ (and corresponding rate $R_c$) that we want to deliver to both Bob and Calvin. Our protocol and the converse proof can be easily extended to cover this case. The capacity region becomes

$$
\max \left\{ \frac{R_1(1 - \delta_2)}{\delta_2(1 - \delta_1)(1 - \delta_1\delta_2)}, \frac{R_2(1 - \delta_1)}{\delta_1(1 - \delta_2)(1 - \delta_1\delta_2)} \right\} + \max \left\{ \frac{R_1 + R_c}{1 - \delta_1}, \frac{R_2}{1 - \delta_1\delta_2}, \frac{R_1}{1 - \delta_1\delta_2}, \frac{R_2 + R_c}{1 - \delta_2} \right\} \leq L \log q. \quad (22)
$$

where the second term is the known bound for (not-secure) message sending [2], while the first term corresponds to the overhead of key generation, same as before.

Partially secret messages: Another natural extension is to keep secret only one part of the private message to each user; that is, we have $W_1 = (W_1', W_1'')$, $W_2 = (W_2', W_2'')$ with a secrecy requirement only for $W_1'$ and $W_2'$. Accordingly, $R_1 = R_1' + R_1''$, $R_2 = R_2' + R_2''$. Assuming the messages are independent and $W_1'$, $W_2'$ are uniformly distributed over their alphabets, our results easily extends to this case, with capacity region

$$
\max \left\{ \frac{R_1'(1 - \delta_2)}{\delta_2(1 - \delta_1)(1 - \delta_1\delta_2)} - \frac{R_1''}{1 - \delta_1\delta_2}, \frac{R_2'(1 - \delta_1)}{\delta_1(1 - \delta_2)(1 - \delta_1\delta_2)} - \frac{R_2''}{1 - \delta_1\delta_2}, 0 \right\} + \max \left\{ \frac{R_1 + R_c}{1 - \delta_1}, \frac{R_2}{1 - \delta_1\delta_2}, \frac{R_1}{1 - \delta_1\delta_2}, \frac{R_2 + R_c}{1 - \delta_2} \right\} \leq L \log q. \quad (23)
$$

Correlated erasures: Our results extend to arbitrary correlation between the erasure patterns, as long as the distribution is known a-priori. Both the protocol and the converse can be modified to characterize the secure transmission rates for this case. The resulting capacity region depends on the joint distribution.

Strengthening the malicious user: Our security guarantees assume that the malicious user may choose the marginal distribution of the other user’s message, but his own message is assumed to be independent and uniformly distributed over its alphabet; moreover, he can only learn his message through the channel outputs he receives. A stronger malicious user could choose the joint distribution of the messages (and may also have access to his own message). Even under this stronger security definition, it is not hard to see we can achieve nonzero rates (e.g., by two instantiations of our protocol, first for Bob with $R_2$ set to 0 and then for Calvin with $R_1$ set to 0), however we conjecture that the capacity region is in general smaller than what we derived here.

Denial-of-Service (DoS) attacks: We leave the question open if the malicious user launches denial-of-service attacks (outside of what our current model allows); however, in general such attacks can be deterred by ensuring they reveal who the attacker is. As an example, in the key generation phase of our protocol, we assumed that
Bob & Calvin cannot learn the other’s feedback before sending their own. This assumption stops a malicious Bob from acknowledging the exact same packets as Calvin which would lead to protocol failure for Calvin and a DoS attack. In practice, for half the ACKs, we can ask Bob to send them first before Calvin, and for the other half Calvin to send them first before Bob, and thus identify users attempting such attacks. In this category are also attacks that attempt to (partially) control the channel, for example through physical layer jamming, where we can resort to physical layer techniques to find the jammer’s real location.
A Complexity considerations

It is clear from the analysis in Section 5 that the length $n$ of the scheme grows as $\max\{O(\log^2(\frac{1}{\epsilon})), O(\frac{1}{\epsilon^4})\}$, where $\epsilon$ is the security and probability of error parameter, and $\epsilon'$ is the gap parameter associated with the rate (see Definition 4). The algorithmic complexity is quadratic in $n$; quadratic from the matrix multiplication to produce the key.

Also, for the proposed scheme, the size (entropy in bits) of $\Theta_A$ is linear in $n$ and no private randomness is needed at Bob and Calvin. For a malicious user, we allow unlimited amount of private randomness.
at least here we assume that an error was not declared for Bob in the key generation phase and hence Bob did receive (not necessarily the same as those that he acknowledges) out of the first Proof.

Lemma 1. When Bob is honest and no error is declared for Bob in the key generation phase, 
\[ I(K_B; Y_2^{n_1} S^{n_1}) \leq k_B e^{-c_1 \sqrt{KL \log q}}, \]  
(24)

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

**Proof.** With a slight abuse of notation, in the following \( X_1^{BC} \) will denote the actual packets Calvin received (not necessarily the same as those that he acknowledges) out of the first \( k_1 \) packets Bob received. Note that here we assume that an error was not declared for Bob in the key generation phase and hence Bob did receive at least \( k_1 \) packets in the key generation phase. Also let \( X_1^{B0} \) be the packets seen only by Bob among the first \( k_1 \) he receives. Let \( I_{B0} \) and \( I_{BC} \) be the index sets corresponding to \( X_1^{B0} \) and \( X_1^{BC} \). Recall that \( X_1^B \) denotes the first \( k_1 \) packets received by Bob. The notation \( M^I \) will denote a matrix \( M \) restricted to the columns defined by index set \( I \). Given this,

\[
I(K_B; Y_2^{n_1} S^{n_1}) = I(X_1^B G_{K_B}; X_1^{BC} S^n) \\
= H(X_1^B G_{K_B}) - H(X_1^B G_{K_B} | X_1^{BC} S^n) \\
= k_B L \log q - H(X_1^B G_{K_B} | X_1^{BC} S^n) \\
= k_B L \log q - H([X_1^{B0} G_{K_B}^{I_{B0}} X_1^{BC} G_{K_B}^{I_{BC}}] | X_1^{BC} S^n) \\
= k_B L \log q - H([X_1^{B0} G_{K_B}^{I_{B0}} X_1^{BC} S^n]),
\]

where the third equality follows from the MDS property of the matrix \( G_{K_B} \). Using the same property, we have

\[
H(X_1^{B0} G_{K_B}^{I_{B0}} | S^n) = \sum_{i=0}^{k_1} \min\{i, k_B\} L \log q \Pr\left\{|X_1^{B0}| = i\right\} \\
\geq k_B L \log q \sum_{i=k_B}^{k_1} \Pr\left\{|X_1^{B0}| = i\right\} = k_B L \log q \Pr\left\{|X_1^{B0}| \geq k_B\right\} \\
= k_B L \log q \left(1 - \Pr\left\{|X_1^{B0}| < k_B\right\}\right)
\]
\[ I(K_B; Y^{n_1} S_i) \leq k_B L \log q e^{-c_1 \sqrt{k_1}}. \quad (25) \]

where the third equality follows from the fact that the indicator random variable \( 1_{B,i}^{C_i} \) is a deterministic function of the conditioning random variables.

\[ \square \]
Lemma 3.

\[ H(X^n_2 | W_1 Y^n_2 S^n \Theta_C U_C) \geq \mathbb{E} \left\{ \min \left( k_B, M_B^{(C)} \right) \right\} L \log q - I(K_B; Y^n_1 S^n). \]

Proof. We adopt the notation for \( U_B^C \) and \( 1^C_{B,i} \) introduced in the proof of Lemma 2. In addition, let \( K'^C_B \) be defined in a similar manner as \( U_B^C \) such that \( K'^C_B, i = \bot \) if \( U_B^C = \bot \) and \( K'^C_B, i = K_B, i \) otherwise. Also, let \( 1^C_{B,i} \) be the vector of indicator random variables \( 1^C_{B,i} \), \( j = 1, \ldots, N_1 \).

Proceeding as in the proof of Lemma 2 we have

\[
H(Y^n_2 | W_1 Y^n_2 S^n \Theta_C U_C) = H(U_B^C | W_1 Y^n_2 S^n \Theta_C U_C)
\]
\[
\geq H(K'^C_B | W_1 Y^n_2 S^n \Theta_C U_C)
\]
\[
= H(K'^C_B | W_1 Y^n_2 S^n \Theta_C U_C) - I(K'^C_B; W_1 Y^n_2 S^n \Theta_C U_C | 1^C_B)
\]

But, from the MDS property of \( G_{K'B} \), and the fact that \( K_B \) is uniformly distributed over its alphabet, we have

\[
H(K'^C_B | 1^C_B) = \sum_{i=1}^{N_1} \min(i, k_B) \Pr \left\{ \sum_{j=1}^{N_1} 1^C_{B,j} = i \right\} L \log q
\]
\[
= \mathbb{E} \left\{ \min \left( k_B, \sum_{i=1}^{N_1} 1^C_{B,i} \right) \right\} L \log q.
\]

Also,

\[
I(K'^C_B; W_1 Y^n_2 S^n \Theta_C U_C | 1^C_B) \overset{(a)}{=} I(K'^C_B; Y^n_1 S^n | 1^C_B)
\]
\[
\leq I(K'^C_B 1^C_B; Y^n_2 S^n)
\]
\[
\leq I(K_B; Y^n_2 S^n).
\]

where (a) follows from the fact that the distribution of \( W_2 \) (uniform and independent of \( S^n, \Theta_A, \Theta_C \)) implies that \( U_C \) is independent of \( \Theta_A, S^n \) and using this we can argue that the following is Markov chain

\[
K'^C_B - (1^C_B, Y^n_2, S^n) - (W_1, \Theta_C, U_C).
\]

Substituting back we have the lemma. \( \square \)

C Proof of Lemmas 4-7

First we give a bound on (21b). The first lemma expresses that Alice has to send sufficient information of message \( W_1 \) such that the Bob and Calvin together (in fact Bob himself also) can reconstruct it despite of erasures.

Lemma 4. From conditions (7)-(3) it follows that

\[
\sum_{i=1}^{n} I(X_i; W_1 | Y^{i-1}_1 Y^{i-1}_2 S^{i-1}) \geq \frac{nR_1}{1 - \delta_1 \delta_2} - \mathcal{E}_1
\]

where \( \mathcal{E}_1 = \frac{h_2(c') + L \log q}{1 - \delta_1 \delta_2} \).
Proof.

\[
nR_1 - \mathcal{E}_1(1 - \delta_1 \delta_2) \leq I(Y_1^n Y_2^n S^n; W_1) = \sum_{i=1}^{n} I(Y_i; Y_2; W_1 | Y_i^{i-1} Y_2^{i-1} S_i^{i-1}),
\]

\[
= \sum_{i=1}^{n} I(Y_i; Y_2; W_1 | Y_i^{i-1} Y_2^{i-1} S_i^{i-1}) = \sum_{i=1}^{n} I(Y_i; Y_2; W_1 | Y_i^{i-1} Y_2^{i-1} S_i^{i-1}, S_i \neq \emptyset) \Pr\{S_i \neq \emptyset\}
\]

\[
= \sum_{i=1}^{n} I(X_i; W_1 | Y_i^{i-1} Y_2^{i-1} S_i^{i-1})(1 - \delta_1 \delta_2)
\]

Here, the first inequality is Fano’s inequality \footnote{Fano’s inequality is a fundamental result in information theory that relates the mutual information between two random variables to the probability of error in communication.} (Chapter 2). Besides, we exploited the independence property of \(S_i\).

\[\square\]

\textbf{Lemma 5.} From conditions \footnote{These conditions are not explicitly stated in the document, but they are likely related to the independence or the distribution of the random variables.} it follows that

\[
\sum_{i=1}^{n} I(X_i; Y_2^{i-1} | Y_1^{i-1} S_i^{i-1}) \geq \frac{n R_1 \delta_1 (1 - \delta_2)}{(1 - \delta_1)(1 - \delta_1 \delta_2)} - \mathcal{E}_2,
\]

where \(\mathcal{E}_2 = \frac{h_2(\epsilon') + \epsilon' L \log q}{1 - \delta_1} \).

\textbf{Proof.} From Lemma \footnote{The details of the proof are not provided in the document.} \[
\frac{n R_1}{1 - \delta_1} - \mathcal{E}_2 \leq \sum_{i=1}^{n} I(X_i; Y_1^{i-1} S_i^{i-1})
\]

\[
= \sum_{i=1}^{n} I(X_i; W_1 | Y_1^{i-1} Y_2^{i-1} S_i^{i-1}) + I(X_i; Y_2^{i-1} | Y_1^{i-1} S_i^{i-1}) - I(X_i; Y_2^{i-1} | Y_1^{i-1} S_i^{i-1} W_1)
\]

\[
\leq \sum_{i=1}^{n} I(X_i; W_1 | Y_1^{i-1} Y_2^{i-1} S_i^{i-1}) + I(X_i; Y_2^{i-1} | Y_1^{i-1} S_i^{i-1})
\]

\[
\leq \frac{n R_1}{1 - \delta_1 \delta_2} + \sum_{i=1}^{n} I(X_i; Y_2^{i-1} | Y_1^{i-1} S_i^{i-1})
\]

To get \footnote{The details of the proof are not provided in the document.} we used Lemma \footnote{The details of the proof are not provided in the document.} \[\square\]

Lemma \footnote{The details of the proof are not provided in the document.} can be interpreted as the connection between the generation and consumption of the randomness Bob knows but Calvin doesn’t.

\textbf{Lemma 6.} From conditions \footnote{These conditions are not explicitly stated in the document, but they are likely related to the independence or the distribution of the random variables.} it follows that

\[
\sum_{i=1}^{n} H(X_i | Y_1^{i-1} Y_2^{i-1} S_i^{i-1} W_1) \geq \frac{(1 - \delta_2)}{(1 - \delta_1) \delta_2} \sum_{i=1}^{n} I(X_i; Y_1^{i-1} | Y_2^{i-1} S_i^{i-1} W_1)
\]

\textbf{Proof.}

\[
0 \leq H(Y_1^n S^n | Y_2^n W_1) = H(Y_1^n S^n | Y_2^n W_1) + H(Y_1^n S^n W_1 | Y_2^n W_1)
\]

\[
= H(Y_1^n S^n | Y_2^n W_1) - I(Y_1^n S^n | Y_2^n W_1) + H(Y_1^n S^n W_1 | Y_2^n W_1)
\]

\[
= H(Y_1^n S^n | Y_2^n W_1) - I(Y_1^n S^n | Y_2^n W_1) + H(Y_1^n S^n W_1 | Y_2^n W_1)
\]

\[
= H(Y_1^n S^n W_1 | Y_2^n W_1) - I(Y_1^n S^n W_1 | Y_2^n W_1) + H(Y_1^n S^n W_1 | Y_2^n W_1)
\]

\[
+ H(Y_1^n | Y_2^n W_1, S_n = B) \Pr\{S_n = B\}
\]

\[
+ H(Y_1^n | Y_2^n S^n W_1, S_n = BC) \Pr\{S_n = BC\}
\]
Lemma 7. From the conditions (1)-(3) it also follows that

\[ \sum_{i=1}^{n} I(X_i; Y_i^{i-1} | Y_2^{i-1} S^{i-1} W_1) + \mathcal{E}_3 > \frac{nR_1}{1 - \delta_1 \delta_2} + \frac{nR_2\delta_2(1 - \delta_1)}{(1 - \delta_2)(1 - \delta_1\delta_2)} \]

Proof. From Lemma 10:

\[ \mathcal{E}_3 > \sum_{i=1}^{n} I(X_i; W_1 | Y_2^{i-1} S^{i-1}) \]

\[ = \sum_{i=1}^{n} H(X_i | Y_2^{i-1} S^{i-1}) - H(X_i | Y_2^{i-1} S^{i-1} W_1) \]

\[ = \sum_{i=1}^{n} H(X_i | Y_2^{i-1} S^{i-1}) - H(X_i | Y_2^{i-1} S^{i-1} W_1) + I(X_i; Y_1^{i-1} | Y_2^{i-1} S^{i-1}) - H(X_i | Y_2^{i-1} S^{i-1} W_1) \]

\[ = \sum_{i=1}^{n} -I(X_i; Y_1^{i-1} | Y_2^{i-1} S^{i-1} W_1) + I(X_i; W_1 | Y_1^{i-1} Y_2^{i-1} S^{i-1}) + I(X_i; Y_1^{i-1} | Y_2^{i-1} S^{i-1}) \]

From Lemma 1,

\[ \sum_{i=1}^{n} I(X_i; W_1 | Y_1^{i-1} Y_2^{i-1} S^{i-1}) \geq \frac{nR_1}{1 - \delta_1 \delta_2} - \mathcal{E}_1. \]

Further, a symmetric result to Lemma 5 shows:

\[ \sum_{i=1}^{n} I(X_i; Y_1^{i-1} | Y_2^{i-1} S^{i-1}) \geq \frac{nR_2\delta_2(1 - \delta_1)}{(1 - \delta_2)(1 - \delta_1\delta_2)} - \mathcal{E}_4, \]

where \( \mathcal{E}_4 = \frac{h_2(e^c) + e^c L \log q}{1 - \delta_2} \). Applying these bounds results the statement of the lemma, with \( \mathcal{E}_5 = \mathcal{E}_3 + \mathcal{E}_1 + \mathcal{E}_4. \)

Lemma 8. From conditions (1)-(3) it follows that

\[ \frac{nR_1}{1 - \delta_1 \delta_2} \geq \sum_{i=1}^{n} I(X_i; W_1 | Y_1^{i-1} Y_2^{i-1} S^{i-1}) \]
Proof.

\[ nR_1 \geq H(W_1) \geq I(Y_1^n Y_2^n S^n; W_1) = \sum_{i=1}^{n} I(Y_i; Y_2^n; W_1|Y_1^{i-1}Y_2^{i-1}S^{i-1}) \]

\[ = \sum_{i=1}^{n} I(Y_i; Y_2^n; W_1|Y_1^{i-1}Y_2^{i-1}S^{i-1}, S_i \neq \emptyset) \Pr \{S_i \neq \emptyset\} \]

\[ = \sum_{i=1}^{n} I(X_i; W_1|Y_1^{i-1}Y_2^{i-1}S^{i-1})(1 - \delta_1\delta_2) \]

We used the same properties as before.

With the same type of argument that we used to prove Lemma 4, we can show also the following:

**Lemma 9.** From conditions (1)-(3) it follows that

\[ \sum_{i=1}^{n} I(X_i; W_1|Y_1^{i-1}S^{i-1}) \geq \frac{nR_1}{1 - \delta_1} - \mathcal{E}_2. \]

**Lemma 10.** From the security condition (4) it follows that

\[ \mathcal{E}_3 > \sum_{i=1}^{n} I(X_i; W_1|Y_2^{i-1}S^{i-1}), \]

where \( \mathcal{E}_3 = \frac{\epsilon}{1 - \delta_2} \).

*Proof.* From (4), we have that

\[ \epsilon > I(Y_2^n S^n \cap C; W_1) \geq I(Y_2^n S^n; W_1) = \sum_{i=1}^{n} I(W_1; Y_2^n; W_2|Y_2^{i-1}S^{i-1}) = \sum_{i=1}^{n} I(Y_2^{i-1}S^{i-1}C) \Pr \{C \subset S_i\} \]

\[ = \sum_{i=1}^{n} I(X_i; W_1|Y_2^{i-1}S^{i-1}, C \subset S_i)(1 - \delta_2) = \sum_{i=1}^{n} I(X_i; W_1|Y_2^{i-1}S^{i-1})(1 - \delta_2) \]

\[ \square \]