

# Threshold-Secure Coding with Shared Key

Nasser Aldaghri, *Student Member, IEEE*, and Hessam Mahdavifar, *Member, IEEE*

**Abstract**—Cryptographic protocols are often implemented at upper layers of communication networks, while error-correcting codes are employed at the physical layer. In this paper, we consider utilizing readily-available physical layer functions, such as encoders and decoders, together with shared keys to provide a *threshold-type* security scheme. To this end, we first consider a scenario where the effect of the physical layer is omitted and all the channels between the involved parties are assumed to be noiseless. We introduce a model for threshold-secure coding, where the legitimate parties communicate using a shared key such that an eavesdropper does not get any information, in an information-theoretic sense, about the key as well as about any subset of the input symbols of size up to a certain threshold. Then, a framework is provided for constructing threshold-secure codes from linear block codes while characterizing the requirements to satisfy the reliability and security conditions. Moreover, we propose a threshold-secure coding scheme, based on Reed-Muller (RM) codes, that meets security and reliability conditions. Furthermore, it is shown that the encoder and the decoder of the scheme can be implemented efficiently with quasi-linear time complexity. In particular, a successive cancellation decoder is shown for the RM-based coding scheme. Then we extend the setup to the scenario where the channel between the legitimate parties is no longer noiseless. The reliability condition for noisy channels is then modified accordingly, and a method is described to construct codes attaining threshold security as well as desired reliability, i.e., robustness against the channel noise. Moreover, we propose a coding scheme based on RM codes for threshold security and robustness designed for binary erasure channels along with a unified successive cancellation decoder. The proposed threshold-secure coding schemes are flexible and can be adapted for different key lengths.

## I. INTRODUCTION

Conventional cryptosystems are often designed to be computationally secure by relying on unproven assumptions of hardness of mathematical problems. Information-theoretic security methods provide an alternative approach by constructing codes for keyless secure communication, as in wiretap channels introduced in a seminal work by Wyner [2]. Since then, various types of wiretap channels have been considered in the literature [3], [4], and with employing different coding schemes as in [5], [6].

Several approaches to provide security in the physical layer assuming shared secret keys have been considered in

The material in this paper was presented in part at the 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton) in September 2019 [1]. This work was supported in part by the National Science Foundation under Grant CCF-1763348 and Grant CCF-1909771.

N. Aldaghri and H. Mahdavifar are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 (email: aldaghi@umich.edu and hessam@umich.edu).

the literature. Such shared keys can be either fixed prior to communication as in classical cryptographic protocols or can be extracted from a source of common randomness [7] such as characteristics of the physical layer channel, see, e.g., [8]–[10]. For instance, a variation of the wiretap channel model, where a shared secret key is assumed to be constantly generated by the legitimate parties, namely Alice and Bob, is studied in [11]. Another approach is to design an encryption scheme that utilizes properties of certain modulation schemes such as orthogonal frequency-division multiplexing (OFDM) to ensure security, see, e.g., [12]–[14]. Other related works include using channel reciprocity properties [15], classical stream ciphers at the physical layer [16], introducing artificial noise [17], multiple-input and multiple-output (MIMO) systems [18], public-key based McEliece cryptosystem [19], and using error-correcting codes for encryption [20], [21]. These prior works either consider noisy channels as in the wiretap channel model or utilize cryptographic primitives being evaluated using cryptographic measures rather than information-theoretical measures to establish security.

Another related line of research is secure network coding, where a wiretapper has access to a certain number of edges in a network over which a source wishes to communicate messages securely. Several works have considered information-theoretic security measures while designing network codes, see, e.g., [22], [23]. A similar line of work has appeared in the context of index coding, where multiple users have partial information about a set of messages and want to receive certain other messages from a central node. The eavesdropper in this scenario is then assumed to have access to a certain number of messages and a certain number of transmissions while the security of the entire message block is considered, see, e.g., [24], [25]. Also, in the context of distributed storage, security guarantees are studied while having trusted storage nodes in untrusted networks. More specifically, scenarios are considered where an eavesdropper/untrusted node has access to a certain number of coded symbols and the goal is to ensure that it is not feasible to reconstruct any individual symbol of the message, e.g., the message intended for another node, see, e.g., [26]–[28]. These prior works differ from the setting considered in this paper in two major aspects. Firstly, they are concerned with keyless techniques with information-theoretic guarantees, e.g., secret sharing, and secondly, the eavesdropper is often assumed to have access to partial information about the message/set of messages rather than the entire information block.

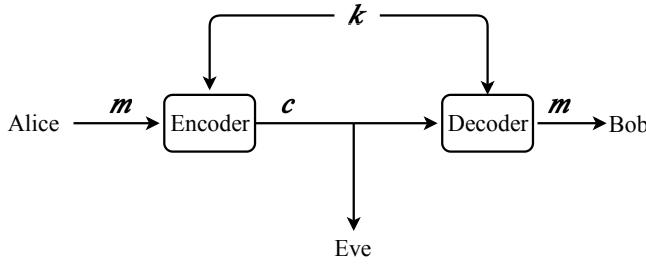


Figure 1. System setup for the proposed coding scheme.

Utilizing error-correcting codes to provide security in the physical layer enables sharing hardware resources between reliability and security schemes in low-cost devices. Consequently, this leads to a promising approach for low-complexity applications, such as Internet-of-Things (IoT) networks. In this paper, we consider using block codes to provide a *threshold-type* security scheme. A fixed key is assumed to be securely shared between the legitimate parties Alice and Bob *a priori*. First, we consider a scenario where the effect of the physical layer is abstracted out and all the channels between the involved parties are assumed to be noiseless. In other words, Alice communicates to Bob over a noiseless channel and her transmissions reach an eavesdropper, namely Eve, also through a noiseless channel, as shown in Figure 1. The security condition in this model is described as follows. Alice encodes her message using the shared key while ensuring that Eve does not obtain any information about the key as well as about any subset of the input message symbols of size up to a certain threshold  $t$ . This condition is referred to as the  $t$ -threshold security condition. Then we consider the case where Alice and Bob share a noisy channel, while the eavesdropper Eve acquires Alice's transmission noise-free. The considered threshold-type security becomes relevant in applications where the knowledge of most, if not all, of the individual data symbols is needed in order to deduce meaningful knowledge about the content of the message. Examples of this type of data include measurement numbers, network commands, the index of elements in a dataset, randomly assigned identification numbers, as well as barcodes or data in any application where the data symbols are already scrambled, hashed, or masked prior to being encoded. A more detailed explanation of such applications is discussed in Section II-B. Furthermore, ensuring the security of the key in the model guarantees that it can be, theoretically, used infinitely many times without leaking any information about it or the messages to Eve.

In the setups considered in this paper, we deviate from conventional physical-layer security settings by removing any condition on the channel from Alice to Eve; in fact, we assume this channel is noiseless. However, we still describe the schemes in a communication setting with the aim of integrating such schemes with channel coding in the physical

layer. To this end, a general scheme for noiseless channels using linear block codes for the  $t$ -threshold-secure coding scheme is shown. Furthermore, we describe a specific construction based on RM codes [29] that meets the threshold security condition, and show an encoder and a decoder, with quasi-linear complexity, to reliably retrieve the message using the shared key. Moreover, we discuss a general method for constructing codes, closely related to concatenated codes [30], for noisy channels that satisfy the threshold security requirements with respect to Eve and provide robust communication for Bob in the presence of channel noise. Also, we propose an explicit RM-based construction that is both  $t$ -threshold-secure and capable of correcting erasures, together with a unified successive cancellation decoder that corrects erasures and retrieves the message simultaneously given the shared key.

The rest of this paper is organized as follows. In Section II we describe the setup and formulate the reliability and the security conditions for noiseless channels based on information-theoretic measures. The proposed coding scheme based on linear block codes is described in detail and its security and reliability are evaluated in Section III. Then, we describe an explicit coding scheme based on RM codes together with an encoder and a successive cancellation decoder in Section IV. A general construction of threshold-secure codes for noisy channels together with an explicit low-complexity RM-based coding scheme for binary erasure channels (BEC) are discussed in Section V. Finally, we conclude the paper in Section VI, and discuss several directions for future work.

## II. SYSTEM MODEL AND APPLICATIONS

In this section, we discuss the system model considered in this paper followed by extending certain applications of this model, as discussed in Section I.

### A. System Model

Consider a system model where Alice wishes to securely communicate with Bob, both are legitimate parties, through a noiseless channel. The eavesdropper, namely Eve, is tapping into that channel and observes all the transmitted symbols, as shown in Figure 1. Alice and Bob share a common key sequence  $k$  of length  $k$ , that can be used for encoding and decoding of message  $m$  of length  $m$ . Both the key and the message symbols are from an alphabet of size  $q$ , where  $q$  is a prime power. A certain known permutation  $\pi(\cdot)$  of Alice's message sequence  $m$  together with the key sequence  $k$  is fed as the input to the encoder, denoted by  $u$ , i.e.,  $u = \pi(k, m)$ . The length of  $u$  is  $n = m + k$  and is encoded to a codeword  $c$  of length  $m$ . The entries in  $k$  as well as in  $m$  are assumed to be independent and uniformly distributed. Alice then transmits the codeword  $c$  to Bob over the noiseless channel. Bob receives the codeword and decodes it using the key  $k$  to retrieve the message  $m$ . Eve observes  $c$  and aims at extracting

information about the message  $\mathbf{m}$  as well as the key  $\mathbf{k}$ . In this setup, Alice and Bob agree on the encoder and the decoder a priori, which are also publicly known to Eve.

In this model, the security condition is the following. Although parts of input  $\mathbf{u}$  are disclosed to Eve, no knowledge, in an information-theoretic sense, about any subset of size up to a certain threshold parameter  $t$  of the input symbols will be leaked to Eve. Note that this is different from the traditional measure of information-theoretic security where the mutual information between the entire message block and Eve's observation needs to be zero/almost zero. In a sense, we consider a *sub-block-wise* measure of information-theoretic security. We aim at designing an encoder and a decoder for a noiseless channel that utilizes a shared key  $\mathbf{k}$  to encode a message  $\mathbf{m}$  such that the following conditions are met:

- 1) *Reliability*: Bob is able to decode the message, knowing the key, with probability one, i.e.,

$$H(\mathbf{m}|\mathbf{c}, \mathbf{k}) = 0. \quad (1)$$

- 2) *Key security*: the codeword  $\mathbf{c}$  does not reveal any information about the key  $\mathbf{k}$ , i.e.,

$$I(\mathbf{k}; \mathbf{c}) = 0. \quad (2)$$

- 3) *t-threshold security*: for any  $\mathbf{v} \subseteq \{u_1, u_2, \dots, u_n\}$  with  $|\mathbf{v}| \leq t$ , we have

$$H(\mathbf{v}|\mathbf{c}) = H(\mathbf{v}), \quad (3)$$

where  $t$  is a design parameter specified later.

**Remark 1.** Note that the secrecy capacity of the communication system in Figure 1, even with a relaxed security condition of  $\lim_{m \rightarrow \infty} \frac{1}{m} I(\mathbf{m}, \mathbf{c}) = 0$ , since  $\mathbf{c}$  is of length  $m$ , i.e., *weak* security, is zero [2]. In a related work [11], a source of common randomness is required to generate a key with a certain rate  $R_k$  to ensure non-zero secrecy capacity. However, here, a key of a fixed length is used repeatedly. In a sense, this implies that the key rate is zero as the message length grows large.

A formal definition of a  $t$ -threshold secure code is defined next.

*Definition 1:* We say a code is  $t$ -threshold secure if it meets the reliability and security conditions, where  $t$  is the maximum cardinality of any  $\mathbf{v} \subseteq \{u_1, u_2, \dots, u_n\}$  that satisfies (3).

It is worth noting that the model considered in this paper subsumes a range of previously studied models, e.g., the perfectly-secure one-time-pad (OTP) encryption which is a code with threshold  $t = m$  used once and hence, we have  $H(\mathbf{m}|\mathbf{c}) = H(\mathbf{m})$ . Another related line of work is on certain types of *keyless* security schemes known as unconditionally-secure all-or-nothing transforms (AONT) [31]. More specifically, cases are studied where the eavesdropper observes a vector  $\mathbf{z}$  whose elements are a subset of size  $m - t$  of the set of elements of  $\mathbf{c}$ , where  $\mathbf{c}$  is of length  $m$  [32]. The security condition is then translated to  $H(\mathbf{v}|\mathbf{z}) = H(\mathbf{v})$  for all  $\mathbf{v}$  of size  $t$  as in [32].

## B. Applications

As briefly discussed in Section I, the considered threshold-type security becomes relevant in applications where the entire message or significant portion of it is needed in order for an eavesdropper to obtain meaningful knowledge about the content of the message. In this section, we briefly expand on one of the applications for the described threshold security setup in Section II-A.

Consider an authentication system based on users' biometric information, such as fingerprints, e.g., as described in [33], where the data is assumed to be hashed prior to encoding. Let us denote the fingerprint measurement vector as  $\tilde{\mathbf{x}}$ . Also, let us have the following two functions: a feature extraction function  $f(\cdot)$  and a secure hash function  $g(\cdot)$ . The function  $f(\cdot)$  is an arbitrary function that maps the input vector  $\tilde{\mathbf{x}}$  to another vector  $\mathbf{x}$ . The hash function  $g(\cdot)$  is a mapping from an input space of size  $a$  to a hash table of size  $b$  with the following property:

$$\Pr(g(\mathbf{x}_1) = g(\mathbf{x}_2) | \mathbf{x}_1 \neq \mathbf{x}_2) = \frac{1}{b}, \quad (4)$$

where  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are any input vectors, and the resulting load factor of this hash function is  $\beta = \frac{a}{b}$  [34]. In this example, when a user scans their fingerprint, the measurement vector  $\tilde{\mathbf{x}}$  is processed using  $f(\cdot)$  to produce the vector  $\mathbf{x}$  that is hashed using the hash function  $g(\cdot)$  to produce the hashed vector denoted as  $\mathbf{m}$ , i.e.,

$$\mathbf{m} = g(\mathbf{x}) = g(f(\tilde{\mathbf{x}})). \quad (5)$$

Then the hashed vector  $\mathbf{m}$  is the input to the threshold-secure encoder together with the key. This hashed vector is uniformly distributed by the assumption on the hash function  $g(\cdot)$  in (4). The hashed vector is to be sent to a database that contains the hashed vectors of all authorized users for authentication. For an eavesdropper that aims to learn the vector  $\mathbf{x}$ , knowledge of the entire  $\mathbf{m}$  is needed. Let us assume that the eavesdropper has access to the hash function  $g(\cdot)$ . If  $\mathbf{m}$  is sent as is, the probability of successfully acquiring  $\mathbf{x}$  by the eavesdropper is  $\frac{1}{\beta}$  since the eavesdropper can discard any vector that does not hash to the observed  $\mathbf{m}$ . However, when using threshold-secure coding with threshold  $t$ , and assuming an alphabet of size  $q$ , this probability becomes at most  $\frac{1}{\beta q^t}$  which is exponentially decaying with  $t$ . This is because the eavesdropper needs to retrieve the hashed vector  $\mathbf{m}$  first. Choosing an appropriate parameter  $t$ , e.g., in the order of a few tens, combined with the uniformity of the hash functions, is sufficient to cripple the eavesdropper in a practical setting.

## III. CODING SCHEMES

With a slight abuse of terminology, we refer to a scheme meeting the reliability and security conditions, as described in Section II, simply as a coding scheme. The coding scheme is revealed to all parties, i.e., Alice, Bob, and Eve. When constructing the coding scheme, we aim at designing an encoder

and a decoder as well as specifying the code. For an input  $\mathbf{u} = \pi(\mathbf{k}, \mathbf{m})$  the encoder produces a codeword  $\mathbf{c}$  as follows

$$\mathbf{c} = \mathbf{u}\mathbf{W} = \pi(\mathbf{k}, \mathbf{m})\mathbf{W}, \quad (6)$$

where  $\mathbf{W}$  is an  $n \times m$  matrix with  $n = m + k$ . In this proposed scheme, we consider this matrix as the transpose of a generator matrix  $\mathbf{G}$  of a linear block code.

Consider a  $[n, m, d_{\min}]_q$  linear block code with generator matrix  $\mathbf{G}$ , i.e., a linear block code whose elements are from an alphabet of size  $q$ , and has rate  $R = m/n$  and minimum distance  $d_{\min}$ . Note that in this setup no redundancy in the codeword is required since the channel is noiseless. We aim at utilizing the generator matrix  $\mathbf{G}$  of certain linear block codes to construct a matrix  $\mathbf{W}$  for our coding scheme such that the reliability and security conditions are met.

One can assume that the length of the key is less than the length of the message; otherwise, if  $k \geq m$ , then the straightforward perfectly-secure one-time pad meets the conditions for  $t = m$ . To encode a message  $\mathbf{m}$ , let us denote the set of indices of the rows of  $\mathbf{W}$  that correspond to the message symbols as  $\mathcal{A} \subseteq [m+k] \stackrel{\text{def}}{=} \{1, 2, \dots, m+k\}$ . Then the set of indices of the rows corresponding to the key symbols is  $\mathcal{A}^c = [m+k] \setminus \mathcal{A}$ . The matrix  $\mathbf{W}_{\mathcal{A}}$  denotes the submatrix of  $\mathbf{W}$  with rows indexed by  $\mathcal{A}$ , and the matrix  $\mathbf{W}_{\mathcal{A}^c}$  denotes the submatrix of  $\mathbf{W}$  with rows indexed by  $\mathcal{A}^c$ . The codeword  $\mathbf{c}$  is then expressed as follows:

$$\mathbf{c} = \mathbf{m}\mathbf{W}_{\mathcal{A}} + \mathbf{k}\mathbf{W}_{\mathcal{A}^c}. \quad (7)$$

The choice of  $\pi(\cdot)$ , which corresponds to the choice of  $\mathcal{A}$  and  $\mathcal{A}^c$ , is critical in ensuring security and reliability conditions. Hence, we have the following definition.

*Definition 2:* A code, as described above, is called *proper* if its matrix satisfies the following requirements:

- 1) The resulting submatrix  $\mathbf{W}_{\mathcal{A}}$  is full row rank, i.e.,  $\text{rank}(\mathbf{W}_{\mathcal{A}}) = m$ .
- 2) The resulting submatrix  $\mathbf{W}_{\mathcal{A}^c}$  is also full row rank, i.e.,  $\text{rank}(\mathbf{W}_{\mathcal{A}^c}) = k$ .

One example of codes that are not *proper* is the turbo code [35] whose generator matrix can be written in the form  $\mathbf{G} = [\mathbf{I}_m \mathbf{A}_1 \mathbf{A}_2]$  where  $\mathbf{I}_m$  is the identity matrix whose columns are dedicated to the message while the rest are dedicated to the key. Note that  $\mathbf{A}_2$  is some row-permuted version of  $\mathbf{A}_1$ , and such a permutation may not necessarily result in  $[\mathbf{A}_1 \mathbf{A}_2]^T$  being a full row-rank matrix. Hence, this code is not necessarily *proper*. A code that is not *proper* will result in a lower equivocation rate for Eve about the message, and leads to leakage of information about the key to Eve, as will be clarified throughout this section.

Next, we show that if a code is *proper*, then it meets the reliability condition, as specified in (1), and the security conditions, as specified in (2) and (3). The following lemma shows that the reliability condition is satisfied.

*Lemma 1:* Suppose that the code used in the coding scheme is *proper*, as defined in Definition 2. Then Bob can recover

the message with probability one under maximum a posteriori (MAP) decoding. In other words,

$$H(\mathbf{m}|\mathbf{c}, \mathbf{k}) = 0. \quad (8)$$

*Proof:* By using (7), it can be observed that since Bob has  $\mathbf{c}$  and  $\mathbf{k}$  and since  $\mathbf{W}_{\mathcal{A}}$  is full rank, then Bob can subtract  $\mathbf{k}\mathbf{W}_{\mathcal{A}^c}$  from  $\mathbf{c}$  and then find  $\mathbf{m}$  from  $\mathbf{W}_{\mathcal{A}}$ , which has a unique solution. ■

In the next theorem, we show that a *proper* code meets the key security condition, as specified in (2). Note that satisfying this condition is very critical as even a very small leakage of the key  $\mathbf{k}$  can lead to the entire key being revealed to Eve after using the scheme several times, thereby compromising the security of the message.

*Theorem 2:* Suppose that the code used in the coding scheme is *proper*, as defined in Definition 2. Then the codeword  $\mathbf{c}$  leaks no information about the key  $\mathbf{k}$ , i.e.,

$$I(\mathbf{k}; \mathbf{c}) = 0. \quad (9)$$

*Proof:* The proof is by observing the following set of equalities:

$$I(\mathbf{k}; \mathbf{c}) = H(\mathbf{c}) - H(\mathbf{c}|\mathbf{k}), \quad (10)$$

$$= m \log_2(q) - H(\mathbf{m}\mathbf{W}_{\mathcal{A}} + \mathbf{k}\mathbf{W}_{\mathcal{A}^c}|\mathbf{k}), \quad (11)$$

$$= m \log_2(q) - H(\mathbf{m}\mathbf{W}_{\mathcal{A}}), \quad (12)$$

$$= \log_2(q)(m - \text{rank}(\mathbf{W}_{\mathcal{A}})), \quad (13)$$

$$= 0, \quad (14)$$

where (11) holds by (7) and the uniformity of the key and message symbols, hence the codewords are uniform, (12) holds because  $\mathbf{m}$  and  $\mathbf{k}$  are independent, (13) is by noting that elements of  $\mathbf{m}$  are uniformly distributed and independent, and (14) holds because  $\text{rank}(\mathbf{W}_{\mathcal{A}}) = m$  as the code is *proper* according to Definition 2. ■

Additionally, to fully justify the reuse of  $\mathbf{k}$  for multiple encodings, we include the following corollary.

*Corollary 3:* Suppose that the code used in the coding scheme is *proper*, as defined in Definition 2. Then the codewords  $(\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_v)$  of the independent and uniform messages  $(\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_v)$  leak no information about the key  $\mathbf{k}$ , i.e.,

$$I(\mathbf{k}; \mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_v) = 0. \quad (15)$$

*Proof:* The proof is by observing the following set of equalities:

$$I(\mathbf{k}; \mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_v) = H(\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_v) - H(\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_v|\mathbf{k}), \quad (16)$$

$$= v m \log_2(q) - H(\mathbf{m}_1 \mathbf{W}_{\mathcal{A}}, \mathbf{m}_2 \mathbf{W}_{\mathcal{A}}, \dots, \mathbf{m}_v \mathbf{W}_{\mathcal{A}}), \quad (17)$$

$$= v m \log_2(q) - v H(\mathbf{m}_i \mathbf{W}_{\mathcal{A}}), \quad (18)$$

$$= v \log_2(q)(m - \text{rank}(\mathbf{W}_{\mathcal{A}})), \quad (19)$$

$$= 0, \quad (20)$$

where (17) holds by (7), the uniformity of codewords, and the independence of the key and messages, (18) holds by independence and uniformity of messages  $(\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_v)$ , where  $\mathbf{m}_i$  is uniformly distributed, (19) is by noting that elements of message  $\mathbf{m}_i$  are uniformly distributed and independent, and (20) holds because  $\text{rank}(\mathbf{W}_{\mathcal{A}}) = m$  as the code is *proper* as in Definition 2. ■

The following lemma is well-known. However, it is included here as it is instrumental in characterizing the threshold security of coding schemes based on linear block codes.

*Lemma 4:* [36] For an  $[n, m, d_{\min}]_q$  linear block code with generator matrix  $\mathbf{G}$ , any submatrix of  $\mathbf{G}$  of size  $m \times (n - |\mathcal{D}|)$  obtained by deleting columns indexed by elements of  $\mathcal{D}$ , where  $\mathcal{D} \subseteq [n]$  with  $|\mathcal{D}| = d_{\min} - 1$ , has full row rank, i.e.,

$$\text{rank}(\mathbf{G}_{\mathcal{D}^c}) = m. \quad (21)$$

In the next theorem, we characterize the threshold security of coding schemes based on linear block codes.

*Theorem 5:* A coding scheme constructed by a matrix  $\mathbf{W} = \mathbf{G}^T$ , where  $\mathbf{G}$  is the generator matrix of an  $[n, m, d_{\min}]_q$  linear block code, is  $t$ -threshold secure, where  $t = d_{\min} - 1$ , i.e., we have

$$H(\mathbf{v}|\mathbf{c}) = H(\mathbf{v}), \quad (22)$$

for any  $\mathbf{v} \subseteq \{u_1, u_2, \dots, u_n\}$  with  $|\mathbf{v}| = t$ , and  $t$  is the maximum value for which this condition holds.

*Proof:* Let  $\mathbf{u}$  denote the input to the encoder for the coding scheme, as specified in (6). Suppose that  $\mathbf{v}$  consists of elements of  $\mathbf{u}$  indexed by  $\mathcal{B} = \{i_1, i_2, \dots, i_t\} \subseteq [n]$ , and  $\tilde{\mathbf{u}}$  consists of elements of  $\mathbf{u}$  indexed by  $\mathcal{B}^c = [n] \setminus \mathcal{B}$ . Then we have the following:

$$I(\mathbf{v}; \mathbf{c}) = H(\mathbf{c}) - H(\mathbf{c}|\mathbf{v}), \quad (23)$$

$$= m \log_2(q) - H(\tilde{\mathbf{u}}\mathbf{W}_{\mathcal{B}^c} + \mathbf{v}\mathbf{W}_{\mathcal{B}}|\mathbf{v}), \quad (24)$$

$$= m \log_2(q) - H(\tilde{\mathbf{u}}\mathbf{W}_{\mathcal{B}^c}), \quad (25)$$

$$= \log_2(q)(m - \text{rank}(\mathbf{W}_{\mathcal{B}^c})), \quad (26)$$

$$= 0, \quad (27)$$

where (24) follows due to codewords being uniformly distributed and expansion of random variables, (25) holds by the independence of  $\mathbf{v}$  and  $\tilde{\mathbf{u}}$ , (26) holds due to the uniformity of  $\tilde{\mathbf{u}}$ , and (27) holds by Lemma 4 with  $t = d_{\min} - 1$ . Since the mutual information  $I(\mathbf{v}; \mathbf{c})$  is zero, it implies that the  $t$ -threshold security criteria is met for the parameter  $t = d_{\min} - 1$ , i.e.,

$$H(\mathbf{v}|\mathbf{c}) = H(\mathbf{v}), \quad (28)$$

for any  $\mathbf{v}$  with  $|\mathbf{v}| = t$ , where  $t = d_{\min} - 1$ .

Next, we need to show that  $t = d_{\min} - 1$  is the maximum value for which the threshold security condition holds. Consider a codeword in the codebook generated by  $\mathbf{G}$  that has the

Hamming weight equal to  $t + 1 = d_{\min}$  with non-zero elements at indices denoted by  $\mathcal{F} = \{i_1, i_2, \dots, i_{t+1}\}$ . Then we have the following:

$$H(u_{i_1}, \dots, u_{i_{t+1}}|\mathbf{c}) = H(u_{i_1}, \dots, u_{i_t}|\mathbf{c}) + H(u_{i_{t+1}}|\mathbf{c}, u_{i_1}, \dots, u_{i_t}), \quad (29)$$

$$= H(u_{i_1}, \dots, u_{i_t}|\mathbf{c}), \quad (30)$$

$$\neq H(u_{i_1}, \dots, u_{i_{t+1}}), \quad (31)$$

where (29) follows from the chain rule of entropy, and (30) holds because there exists a linear combination of the entries of  $\mathbf{c} = (c_1, c_2, \dots, c_m)$  such that  $\sum_{i=1}^m \lambda_i c_i = \sum_{j \in \mathcal{F}} \gamma_j u_j$ . Hence, the second term becomes zero, since  $u_{i_{t+1}}$  is uniquely determined given  $\mathbf{c}$  and  $\{u_{i_1}, \dots, u_{i_t}\}$ . Therefore, due to (31), the threshold security condition does not hold for  $t+1 = d_{\min}$ . ■

*Corollary 6:* For any  $t$ -threshold secure coding scheme, constructed from a linear block code, with message length  $m$ , key length  $k$ , and code length  $n = m + k$ , we have  $t \leq k$ .

*Proof:* The proof follows by Theorem 5 together with Singleton bound on the minimum distance of a code. ■

Next, we characterize Eve's equivocation about the entire message  $\mathbf{m}$  after observing the codeword.

*Corollary 7:* If the code is *proper*, then Eve's equivocation about the entire encoded message  $\mathbf{m}$  after observing the codeword is equal to the entropy of the key, i.e.,

$$H(\mathbf{m}|\mathbf{c}) = k \log_2(q). \quad (32)$$

*Proof:* We have the following

$$H(\mathbf{m}|\mathbf{c}) = H(\mathbf{m}) - H(\mathbf{c}) + H(\mathbf{c}|\mathbf{m}), \quad (33)$$

$$= H(\mathbf{k}\mathbf{W}_{\mathcal{A}^c} + \mathbf{m}\mathbf{W}_{\mathcal{A}}|\mathbf{m}), \quad (34)$$

$$= H(\mathbf{k}\mathbf{W}_{\mathcal{A}^c}), \quad (35)$$

$$= k \log_2(q), \quad (36)$$

where (34) follows due to the uniformity of messages and codewords, and expansion of random vectors, (35) holds because of the independence of  $\mathbf{m}$  and  $\mathbf{k}$ , and (36) holds by noting that the matrix  $\mathbf{W}_{\mathcal{A}^c}$  is full row rank since the code is *proper*. ■

The statement of Corollary 7 can be also rephrased by stating that the probability of successfully retrieving the entire message block by Eve is equal to  $q^{-k}$ .

Now that we have established the properties that the coding schemes based on linear block codes satisfy, we need to show how to maximize the threshold  $t$  as stated in Corollary 6, provided that  $q$  is large enough. To this end, we utilize maximum distance separable (MDS) codes to arrive at the following theorem.

*Theorem 8:* For any message length  $m$  and key length  $k$ , there exists a *proper* code with threshold  $t = k$ , provided that the alphabet size  $q \geq m + k + 1$ .

*Proof:* To prove the theorem, we give an example of a code that is shown to be *proper* with  $t = k$ . We utilize

Reed-Solomon (RS) codes, which are a well-known family of codes that are maximum distance separable (MDS) codes, i.e.,  $d_{\min} = n - m + 1 = k + 1$  [36]. For any  $[n, m, d_{\min}]_q$  RS code, all we need to show is that the matrix  $\mathbf{W}$  which is the transpose of the generator matrix  $\mathbf{G}$  of the RS code can be used to construct a *proper* code. One of the properties of MDS codes is that every set of  $m$  columns of the matrix  $\mathbf{G}$  are linearly independent [36, Proposition 11.4]. Note that rows of  $\mathbf{W}$  correspond to columns of  $\mathbf{G}$ . Hence, any choice of  $m$  columns of  $\mathbf{G}$  will have rank  $m$ , and the remaining  $k$  columns of  $\mathbf{G}$  will also have rank  $k$  as it is assumed that  $k < m$ . Therefore, the code generated by  $\mathbf{W}$  is *proper*, with threshold  $t = k$ . ■

Note that the straightforward Gaussian elimination method, with complexity  $\mathcal{O}(m^3)$ , can be always used for decoding of coding schemes based on linear block codes. However, when the underlying linear block code belongs to well-known families of linear block codes, e.g., Reed-Solomon codes, it is desirable to study low-complexity decoders for the resulting coding schemes using the off-the-shelf encoding/decoding methods. For instance, low-complexity decoding of RS codes is based on a low-complexity computation of the inverse of a Vandermonde matrix. Now, for the coding schemes based on RS codes, the evaluation points for the RS encoder are chosen as consecutive powers of  $\alpha$ , where  $\alpha$  is a primitive element of  $\mathbb{F}_q$ . The specific choice of the message and key indices is as follows: the first  $m$  rows of  $\mathbf{W}$  are dedicated for the message  $\mathbf{m}$ , and the last  $k$  rows of  $\mathbf{W}$  are dedicated for the key  $\mathbf{k}$ . Since  $\mathbf{W}$  is a Vandermonde matrix, this choice of message indices together with the specific choice of evaluation points result in a scenario where the submatrix  $\mathbf{W}_{\mathcal{A}}$  is also a Vandermonde matrix. To decode a codeword using the key, the decoder computes  $\mathbf{m} = (\mathbf{c} - \mathbf{k}\mathbf{W}_{\mathcal{A}^c})\mathbf{W}_{\mathcal{A}}^{-1}$ . Note that the inverse of a square Vandermonde matrix of order  $m$  can be computed with complexity  $\mathcal{O}(m^2)$  [37]. This results in  $\mathcal{O}(m^2)$  complexity for the decoding in coding schemes based on RS codes.

#### IV. LOW-COMPLEXITY CONSTRUCTION

In this section, we focus on designing binary codes to meet the reliability and security conditions while providing encoding and decoding algorithms with linear/quasi-linear complexity. To this end, we consider Reed-Muller codes due to their recursive construction and low-complexity decoder. In addition, since they are designed with the objective of maximizing the minimum distance, given their particular recursive structure, we can achieve a reasonably high threshold  $t$  for the  $t$ -threshold security.

It is worth noting that various types of decoders for Reed-Muller codes are proposed in the literature, see, e.g., [29], [38], [39]. However, the proposed decoder here differs from these works as it has different constraints and objectives. The goal of the decoder here is not to correct errors, but rather to successfully recover the message from an error-free

codeword encoded by having the message as well as the key as the input. Also, the message cannot be retrieved completely without complete knowledge of the key itself. This shows the need to adapt or modify encoders/decoders in such a way that they can be utilized for threshold-security decoding accordingly.

##### A. Encoder

First, a brief description of Reed-Muller codes is provided. An  $\text{RM}(s, r)$  code is a  $[2^s, \sum_{i=0}^r \binom{s}{i}, 2^{s-r}]_2$  linear block code. The generator matrix of the  $\text{RM}(s, r)$  code, denoted by  $\mathbf{G}(s, r)$ , is obtained by keeping the rows with the Hamming weight of at least  $2^{s-r}$  from the matrix  $\mathbf{F}^T = (\mathbf{F}_2^{\otimes s})^T$  and removing the remaining rows, where  $\otimes$  denotes the Kronecker product,  $T$  is the transpose operator, and  $\mathbf{F}_2$  is the following kernel matrix

$$\mathbf{F}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \quad (37)$$

Although there are different ways of describing the encoding and the generator matrix of RM codes, the above description helps us to choose the message and key indices, which is the next step towards designing a code that is *proper*. Due to the recursive structure of  $\mathbf{F}$ , it can be observed that indices of the rows with the lowest weight, the second lowest weight, etc, from  $\mathbf{F}$  correspond to indices of columns with the highest column weight, the second highest weight, etc, from  $\mathbf{F}$ , respectively. When specifying the matrix  $\mathbf{G}(s, r)$  as a sub-matrix of  $\mathbf{F}^T$  we choose the set of indices of the removed rows from  $\mathbf{F}^T$  as  $\mathcal{A}^c$  to assign the rows of  $\mathbf{W}$  dedicated for the key, while the indices of the remaining rows are used as the message indices  $\mathcal{A}$ . Then we have the following proposition.

*Proposition 9:* The choice of the sets  $\mathcal{A}$ , and  $\mathcal{A}^c$  as mentioned above results in a *proper* code.

*Proof:* To prove this proposition, it suffices to show that  $\mathbf{W}_{\mathcal{A}}$  and  $\mathbf{W}_{\mathcal{A}^c}$  are both full row rank.

First, it is shown that  $\mathbf{W}_{\mathcal{A}}$  is full row rank. Note that for a full rank lower-triangular matrix, a submatrix obtained by removing a subset of columns and rows with the same indices results also in a full rank lower-triangular matrix. Also, note that  $\mathcal{A}^c$  is the subset of indices of deleted columns as well as that of the rows dedicated for the key from  $\mathbf{F}$ . Hence, the matrix  $\mathbf{W}_{\mathcal{A}}$  is full row rank.

Next, we show that  $\mathbf{W}_{\mathcal{A}^c}$  is full row rank. This is done by induction. Note that  $k < m$  is assumed, as mentioned before. Also, to simplify the proof, let us have  $r' = s - r$ , and also re-express  $k$  and  $m$  in the remainder of the proof as follows

$$k = \sum_{i=0}^{r'} \binom{s}{i},$$

and

$$m = \sum_{i=r'+1}^s \binom{s}{i},$$

where we have  $r' \leq \lfloor \frac{s-1}{2} \rfloor$ . Note that  $\mathbf{W}_{\mathcal{A}^c}$  contains the  $\sum_{i=0}^{r'} \binom{s}{i}$  rows dedicated for the key from  $\mathbf{F}$  with the same number of lowest-weight columns removed. Let this matrix be also denoted by  $\mathbf{F}(s, r')$ . Let also  $\mathbf{F}'(s, r')$  denote the matrix that contains the  $\sum_{i=0}^{r'} \binom{s}{i}$  rows dedicated for the key from  $\mathbf{F}$  with only  $\sum_{i=0}^{r'-1} \binom{s}{i}$  lowest weight columns removed. Due to the recursive structure of the matrix  $\mathbf{F}$ ,  $\mathbf{F}(s, r')$  can be expressed as follows:

$$\mathbf{F}(s, r') = \begin{bmatrix} \mathbf{F}(s-1, r'-1) & \mathbf{0} \\ \mathbf{F}'(s-1, r') & \mathbf{F}(s-1, r') \end{bmatrix}. \quad (38)$$

Next, we show that the matrix  $\mathbf{F}(s, r')$  is full row rank for the maximum value  $r' = \lfloor \frac{s-1}{2} \rfloor$  and for  $s \geq 2$  by induction on  $s$ . Then it will be discussed why this also holds for  $r' < \lfloor \frac{s-1}{2} \rfloor$ .

**Step 1:** The induction basis is for  $s = 2$  and  $r' = 0$ , and for  $s = 3$  and  $r' = 1$ , which can be easily verified, i.e., for  $s = 2$  and  $r' = 0$ , the rank of  $\mathbf{F}(2, 0)$  is 1. Also, for  $s = 3$  and  $r' = 1$ , the rank of  $\mathbf{F}(3, 1)$  is 4.

**Step 2:** Suppose that the induction hypothesis holds for  $s$  and  $s$  is odd. Then we have the following matrix:

$$\mathbf{F}(s+1, r') = \begin{bmatrix} \mathbf{F}(s, r'-1) & \mathbf{0} \\ \mathbf{F}'(s, r') & \mathbf{F}(s, r') \end{bmatrix}. \quad (39)$$

We need to show that  $\text{rank}(\mathbf{F}(s+1, r')) = \sum_{i=0}^{r'} \binom{s+1}{i}$ . Note that  $\mathbf{F}(s, r')$  is full row rank by induction hypothesis, i.e.,  $\text{rank}(\mathbf{F}(s, r')) = \sum_{i=0}^{r'} \binom{s}{i}$ . Then  $\mathbf{F}(s, r'-1)$ , which contains a subset of the rows in  $\mathbf{F}(s, r')$ , is also full row rank. Hence, we have  $\text{rank}(\mathbf{F}(s, r'-1)) = \sum_{i=0}^{r'-1} \binom{s}{i}$ . Therefore,

$$\begin{aligned} \text{rank}(\mathbf{F}(s+1, r')) &= \text{rank}(\mathbf{F}(s, r'-1)) \\ &\quad + \text{rank}(\mathbf{F}(s, r')), \end{aligned} \quad (40)$$

$$= \sum_{i=0}^{r'-1} \binom{s}{i} + \sum_{i=0}^{r'} \binom{s}{i}, \quad (41)$$

$$= \sum_{i=0}^{r'} \binom{s+1}{i}, \quad (42)$$

which is equal to the number of rows in  $\mathbf{F}(s+1, r')$ . Hence, it is full row rank.

For even  $s$  with corresponding parameter  $r'$ , we need to show the following matrix is full row rank

$$\mathbf{F}(s+1, r'+1) = \begin{bmatrix} \mathbf{F}(s, r') & \mathbf{0} \\ \mathbf{F}'(s, r'+1) & \mathbf{F}(s, r'+1) \end{bmatrix}. \quad (43)$$

First, we have  $\text{rank}(\mathbf{F}(s, r')) = \sum_{i=0}^{r'} \binom{s}{i}$  by induction hypothesis. Regarding  $\text{rank}(\mathbf{F}'(s, r'+1))$ , we can see that  $\mathbf{F}'(s, r'+1)$  has  $\sum_{i=0}^{r'} \binom{s}{i}$  rows that are also included in  $\mathbf{F}(s, r')$ . However, when considering the indices of such rows in  $[\mathbf{F}'(s, r'+1) \ \mathbf{F}(s, r'+1)]$ , the corresponding rows are independent from all other rows in  $[\mathbf{F}(s, r'), \ \mathbf{0}]$ . Furthermore, there are  $\binom{s}{r'+1}$  additional rows in  $\mathbf{F}'(s, r'+1)$  that

are linearly independent from the remaining rows due to the structure of the zero blocks in this matrix, similar to (38). We can then find the rank of  $\mathbf{F}(s+1, r'+1)$  as follows

$$\begin{aligned} \text{rank}(\mathbf{F}(s+1, r'+1)) &= \text{rank}(\mathbf{F}(s, r')) \\ &\quad + \text{rank}(\mathbf{F}'(s, r'+1)), \end{aligned} \quad (44)$$

$$\begin{aligned} &= \sum_{i=0}^{r'} \binom{s}{i} + \sum_{i=0}^{r'} \binom{s}{i} \\ &\quad + \binom{s}{r'+1}, \end{aligned} \quad (45)$$

$$= \sum_{i=0}^{r'+1} \binom{s+1}{i}. \quad (46)$$

Hence,  $\mathbf{F}(s+1, r'+1)$  is full row rank, and the induction hypothesis holds for  $s+1$  with the maximum value of  $r'$ . For keys of shorter lengths, it is straightforward to see that for any  $r'' < r'$ , the matrix  $\mathbf{F}(s, r'')$  whose rows are a subset of  $\mathbf{F}(s, r')$  with additional columns inserted at different locations is also full row rank. This completes the proof. ■

**Remark 2.** In the proposed scheme based on RM codes, we have  $n = 2^s$ ,  $m = \sum_{i=0}^r \binom{s}{i}$ , for some  $r \geq s/2$ , and  $k = n - m < m$ . Note that the underlying RM code has rate  $R > \frac{1}{2}$ . By using Theorem 5 and noting that the minimum distance of the underlying code is  $2^{s-r}$ , the achievable threshold security parameter  $t$  for the RM-based scheme with parameters  $(s, r)$  is  $t = 2^{s-r} - 1$ . Note that, in general, for an RM code of constant rate, i.e.,  $R = O(1)$ , we have  $r = s/2 + O(\sqrt{s})$ . Hence, the threshold security parameter of the corresponding scheme is  $t = \sqrt{n} \exp(O(\sqrt{\log n}))$ .

### B. Decoder

In this part, we discuss a low-complexity successive cancellation (SC) decoder to decode the message in the RM-based coding scheme while utilizing the shared key. As Reed-Muller codes are closely related to polar codes [40], a decoder closely related to that of polar codes described in [40] is natural. However, there are fundamental differences that will be clarified throughout this section.

The decoder is described in Algorithm 1. We first embed erasures within the entries of the codeword  $\mathbf{c}$  in order to get a vector of length  $n$ , denoted by  $\mathbf{z}$ , by inserting the erasures at locations indexed by  $\mathcal{A}^c$ . More specifically,  $\mathbf{z} = \pi_1(\mathbf{e}_k, \mathbf{c})$  where  $\mathbf{c}$  is the codeword and  $\mathbf{e}_k$  is an erasure vector of length  $k$  such that the permutation places the erasures at locations denoted by  $\mathcal{A}^c$ . Note that, as mentioned before,  $\mathcal{A}^c$  corresponds to the location of the key bits at the encoder.

The decoder takes the key bits  $\mathbf{k}$ , the codeword embedded with erasures  $\mathbf{z} = \pi_1(\mathbf{e}_k, \mathbf{c})$ , indices of the key bits  $\mathcal{A}^c$  and a recursion index  $i$  as inputs, and outputs the vector  $\mathbf{u} = [u_1, u_2, \dots, u_n] = \pi(\mathbf{k}, \mathbf{m})$  from which the message can be retrieved  $\mathbf{m} = \mathbf{u}_{\mathcal{A}}$ . The high-level idea of the decoder is as follows. The vector  $\mathbf{z}$  is divided into two parts;  $z_1^{n/2} =$

**Algorithm 1** Successive cancellation decoder (Decoder)

---

```

1: Initialization:  $i = 1$ .
2: Input:  $\mathbf{k}$ ,  $z_1^n = \pi_1(\mathbf{e}_k, \mathbf{c})$ ,  $\mathcal{A}^c$ ,  $i$ .
3: Output:  $h_1^n$ ,  $u_1^n$ .
4: if  $n = 2$  then
5:   if  $z_2 = e$  then
6:      $u_i = k_i$ 
7:   else
8:      $u_i = z_2$ 
9:   end if
10:  if  $z_1 = e$  then
11:     $u_{i-1} = k_{i-1}$ 
12:  else
13:     $u_{i-1} = u_i \oplus z_1$ 
14:  end if
15:   $h_1^n = [u_{i-1} \oplus u_i, u_i]$ 
16: else
17:    $\mathbf{h}' \leftarrow \text{Decoder}(\mathbf{k}_2, z_{n/2+1}^n, \mathcal{A}_2^c, 2i)$ 
18:    $\bar{z}_1^{n/2} = \mathbf{h}' \oplus z_1^{n/2}$ 
19:    $\mathbf{h}'' \leftarrow \text{Decoder}(\mathbf{k}_1, \bar{z}_1^{n/2}, \mathcal{A}_1^c, 2i - 1)$ 
20:    $h_1^n = [\mathbf{h}'' \oplus \mathbf{h}', \mathbf{h}']$ 
21: end if
22: return  $h_1^n$ 

```

---

$[z_1, z_2, \dots, z_{n/2}]$  and  $z_{n/2+1}^n = [z_{n/2+1}, z_{n/2+2}, \dots, z_n]$ , that are decoded successively. As opposed to the SC decoder of polar codes [40], the second sub-block is processed first, cancelled from the first sub-block, and then the first sub-block is processed. Each of these sub-blocks is also decoded recursively by splitting them into two parts and so on.

**Remark 3.** When describing the recursive SC decoding process we often use the binary tree terminology in which the input codeword, i.e.,  $\mathbf{z}$ , is assigned to the root of the tree and then the first and the second sub-blocks are assigned to the *left child* and the *right child*, respectively. The decisions are made at the leaves of the tree and then are re-encoded and propagated back through the tree, see, e.g., [41] for more details.

The following claim verifies that the decoder successfully outputs the message bits with probability 1 for any key length. Note that since the proof follows by induction, we discard the assumption that  $k \leq m$  and simply show the claim for any  $k \leq n$ .

*Claim 10:* The RM-based coding scheme can be successfully decoded using the SC decoder in Algorithm 1 for any key length  $k \leq n$ .

*Proof:* We use induction on  $l$ , where  $n = 2^l$ , to show that the claim holds.

**Step 1:** For the induction basis, consider  $n = 2$ . We need to show decoding is successful for  $k = 0, 1, 2$ . For  $k = 0$ , which corresponds to the case with no erasure, the induction hypothesis holds trivially as  $\mathbf{F}$  is non-singular. For  $k = 1$ , one needs to show the induction hypothesis for both possible

cases for  $\mathcal{A}^c$ . First, let us consider that  $z_1 = e$  and  $z_2 = c_1$ , which corresponds to  $u_1 = k_1$ , and  $u_2 = m_1$ . In this case, the decoder outputs  $u_1 = k_1$  and  $u_2 = z_2$ . For the other case where  $z_1 = c_1$  and  $z_2 = e$ , which corresponds to  $u_1 = m_1$  and  $u_2 = k_1$ , the decoder first corrects the erasure, assigning  $u_2 = k_1$ . It then computes  $u_1 = m_1 = u_2 \oplus z_1 = k_1 \oplus z_1$ . Finally, we show it succeeds for  $k = 2$ , where both  $z_1$  and  $z_2$  are erased. Then  $u_1 = k_1$  and  $u_2 = k_2$  and the decoder is successful.

**Step 2:** Now, suppose that the induction hypothesis holds for  $n = 2^l$  and for any  $k \leq 2^l$ , where  $k$  is the length of the key, regardless of the indices of the key bits. However, note that, as specified before, the row indices corresponding to the key bits and the column indices corresponding to the erasures are the same and are both denoted by  $\mathcal{A}^c$ . We now show that the claim is true for  $n = 2^{l+1}$  and any  $k \leq 2^{l+1}$ . Let us split the key indices  $\mathcal{A}^c$  into two sets,  $\mathcal{A}_1^c$  and  $\mathcal{A}_2^c$ , with sizes  $|\mathcal{A}_1^c| = k_1$  and  $|\mathcal{A}_2^c| = k_2$ , where  $k = k_1 + k_2$ , as follows. The set  $\mathcal{A}_1^c$  consists of the indices of erasures in  $z_1^{n/2}$ . Also, let  $\mathbf{k}_1$  denote the corresponding part of the key of size  $k_1$ . Similarly,  $\mathcal{A}_2^c$  consists of the indices of erasures in  $z_{n/2+1}^n$ . Also, let  $\mathbf{k}_2$  denote the corresponding part of the key of size  $k_2$ . First, the right child with input  $z_{n/2+1}^n$ , which has  $k_2$  erasures, is processed. Note that there are also  $k_2$  known key bits indexed by  $\mathcal{A}_2^c$  in the second half sub-block  $u_{n/2+1}^n$ . Note that the decoder for the right child has an input of length  $n' = 2^l$  and  $k' = k_2$  erasures as well as key bits  $\mathbf{k}_2$  indexed by  $\mathcal{A}_2^c$ . The decoder succeeds by the induction hypothesis. The right child then passes

$$u_{n/2+1}^n \mathbf{F}_2^{\otimes l} \oplus z_1^{n/2} = \mathbf{h}' \oplus z_1^{n/2} = \bar{z}_1^{n/2}$$

to the left child. The decoder is then run on  $\bar{z}_1^{n/2}$ , which is of length  $n' = 2^l$  and has  $k' = k_1$  erasures and key bits  $\mathbf{k}_1$  indexed by  $\mathcal{A}_1^c$ . The decoder is successful on this node as well by the induction hypothesis. Hence, the decoder is successful for  $n = 2^{l+1}$  which completes the proof of the claim. ■

## V. ROBUSTNESS

In this section we study a natural scenario for extension of the considered setup and the results. In particular, it is assumed that a noisy channel is present between the legitimate parties and the goal is to study the robustness of the framework and the proposed solution when channel noise is present.

The revised system model, shown in Figure 2, is as follows: the channel between Alice and Bob is no longer noiseless, and it can be a certain type of channel to be studied, e.g., binary symmetric channel (BSC), binary erasure channel (BEC), additive-white Gaussian noise channel (AWGN), etc. However, for the eavesdropper, we still consider a worst-case scenario from the legitimate parties' perspective. In other words, it is assumed that Eve receives the transmitted codeword through a noiseless channel, and hence, she has access to the codeword error-free. Alice aims to utilize a coding scheme such that the threshold security requirement at Eve

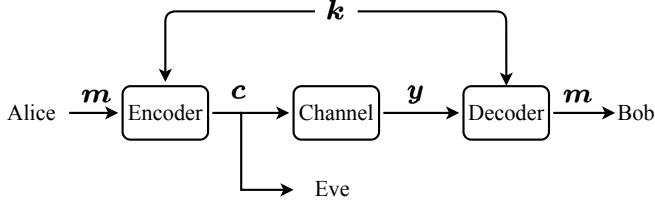


Figure 2. Modified setup for the proposed coding scheme in the presence of a noisy channel.

is satisfied while establishing a reliable communication with Bob that is robust in the presence of channel noise.

Note that the assumption on Eve's observation here makes it reasonable to keep the conditions in (2) and (3) the same in this revised model. On the other hand, the reliability condition in (1) needs to be modified to account for the noisy channel. We do this from a conventional block coding perspective where reliability is measured in terms of a certain number of errors and erasures that can be corrected. More specifically, the reliability condition is still stated as

$$H(m|y, k) = 0, \quad (47)$$

provided that the number of erasures and errors introduced in  $y$  satisfies a certain condition that depends on the underlying coding scheme. For instance, consider coding schemes based on linear block codes. Suppose that the minimum distance of the robustness coding scheme is  $D_{\min}$  when the key is fixed, which is different from the minimum distance of the threshold security coding scheme, i.e.,  $d_{\min}$ . Then the condition on the number of errors and erasures is simply  $2\tau + \rho \leq D_{\min} - 1$ , where  $\tau$  is the number of errors and  $\rho$  is the number of erasures, same as in conventional block codes.

In the remainder of this section, we discuss a general method to construct codes for threshold security and robustness, and describe an explicit low-complexity construction based on Reed-Muller codes for binary erasure channels along with a SC decoder.

#### A. General construction

A straightforward solution to construct coding schemes for the setting described in this section is by utilizing concatenation of two codes. More specifically, a coding scheme, constructed to guarantee the desired threshold security in the error-free case, would be concatenated with an inner code, that can be an off-the-shelf block code, to guarantee the desired reliability for Alice-Bob communication. Although this solution is straightforward, one needs to ensure that the threshold security guarantee is not compromised when more redundancy is added through the inner encoder which will be then revealed to Eve.

In the aforementioned concatenation scheme, the overall encoder and decoder at Alice and Bob, respectively, are referred to as *supercoder* and *superdecoder*, respectively. The

construction of the concatenated scheme is described in more details next. Consider a *proper* coding scheme, that guarantees threshold security requirement, that is obtained from an  $[n, m, d_{\min}]_q$  linear block code with the generator matrix  $\mathbf{W}^T$ . Also, consider an error-correcting code, used as an inner code to guarantee the reliability, that is an  $[N, m, D_{\min}]_q$  linear block code with the generator matrix denoted by  $\mathbf{G}_r$ . It is important to note that both codes have the same dimension  $m$ .

The encoding process is as follows. First,  $\mathbf{u} = \pi(k, m)$  is passed through the outer threshold security encoder that multiplies  $\mathbf{u}$  by  $\mathbf{W}$ . The result is then passed to the inner encoder, which multiplies its input by  $\mathbf{G}_r$ . Then the resulting codeword  $\mathbf{c} = \mathbf{u}\mathbf{W}\mathbf{G}_r$  is transmitted to Bob through the noisy channel. Bob receives a corrupted version of the codeword  $\mathbf{c}$ , denoted as  $\mathbf{y}$ , and passes it through the decoder consisting of an inner decoder and an outer decoder. The inner decoder retrieves  $\tilde{\mathbf{c}} = \mathbf{u}\mathbf{W}$ . Note that we have  $\tilde{\mathbf{c}}$  error-free provided that the number of errors and/or erasures satisfies the given condition on the reliability guarantee of the inner code. Then  $\tilde{\mathbf{c}}$  together with the key  $k$  are passed through the outer decoder, designed for the threshold security coding scheme; hence, retrieving  $\mathbf{m}$ . The following lemma states that this construction does not compromise the key and threshold security conditions.

*Lemma 11:* The aforementioned concatenation scheme results in a  $t$ -threshold secure code.

*Proof:* To show that the lemma holds, we need to have  $\text{rank}(\mathbf{WG}_r) = m$ ,  $\text{rank}(\mathbf{W}_A\mathbf{G}_r) = m$ ,  $\text{rank}(\mathbf{W}_{A^c}\mathbf{G}_r) = k$ , and  $\text{rank}(\mathbf{W}_{B^c}\mathbf{G}_r) = m$ , where  $\mathcal{A}$  and  $\mathcal{A}^c$  are chosen such that the code is *proper*, as stated in Definition 2, and  $\mathcal{B}^c$  is as defined in Theorem 5. It can be observed that all these equations hold simply because  $\mathbf{G}_r$  is full row rank. ■

#### B. Low-complexity construction

In this section, we aim at presenting a unified coding scheme, for threshold security and robustness, that can be decoded using one unified SC decoder. This would potentially result in more efficient hardware implementation and improved latency compared to the general concatenated scheme.

In particular, a scenario with binary symbol erasures is considered, where at most  $\rho = D_{\min} - 1$  erasures are assumed to occur with  $D_{\min}$  being the minimum distance of the underlying code. For the proposed coding scheme, an encoder is presented together with a superdecoder that simultaneously corrects erasures and decodes the message using the key. To this end, the coding scheme presented for noiseless channels in Section IV is extended to be utilized along with an RM-based code to handle binary erasures.

1) *Encoder:* In the considered scheme, the same RM code is used for threshold security and robustness. More specifically, an  $\text{RM}(s, r)$  is used, which is, as previously described, a  $[2^s, \sum_{i=0}^r \binom{s}{i}, 2^{s-r}]_2$  with the generator matrix denoted by

$$\tilde{\mathbf{G}}(s, r) = \mathbf{G}^T(s, r)\mathbf{G}(s, r), \quad (49)$$

$$= \begin{bmatrix} \mathbf{G}^T(s-1, r-1) & \mathbf{G}^T(s-1, r) \\ \mathbf{0} & \mathbf{G}^T(s-1, r) \end{bmatrix} \begin{bmatrix} \mathbf{G}(s-1, r-1) & \mathbf{0} \\ \mathbf{G}(s-1, r) & \mathbf{G}(s-1, r) \end{bmatrix}, \quad (50)$$

$$= \begin{bmatrix} \mathbf{G}^T(s-1, r-1)\mathbf{G}(s-1, r-1) + \mathbf{G}^T(s-1, r)\mathbf{G}(s-1, r) & \mathbf{G}^T(s-1, r)\mathbf{G}(s-1, r) \\ \mathbf{G}^T(s-1, r)\mathbf{G}(s-1, r) & \mathbf{G}^T(s-1, r)\mathbf{G}(s-1, r) \end{bmatrix}, \quad (51)$$

$$= \begin{bmatrix} \tilde{\mathbf{G}}(s-1, r-1) + \tilde{\mathbf{G}}(s-1, r) & \tilde{\mathbf{G}}(s-1, r) \\ \tilde{\mathbf{G}}(s-1, r) & \tilde{\mathbf{G}}(s-1, r) \end{bmatrix}. \quad (52)$$

$\mathbf{G}(s, r)$ . The encoder with input  $\mathbf{u}$ , consisting of both the message and the key, outputs the codeword  $\mathbf{c}$  specified as follows:

$$\mathbf{c} = \mathbf{u}\mathbf{G}^T(s, r)\mathbf{G}(s, r) = \mathbf{u}\tilde{\mathbf{G}}(s, r), \quad (48)$$

where  $\tilde{\mathbf{G}}(s, r)$  is a notation introduced here to denote  $\mathbf{G}^T(s, r)\mathbf{G}(s, r)$ . Note that the encoder can be implemented recursively, since  $\tilde{\mathbf{G}}(s, r)$  can be expressed recursively as shown in (52).

Note that the encoder described by (48) utilizes the construction presented in Section IV-A, which achieves threshold security parameter  $t = 2^{s-r} - 1$ , and we use the same choice of indices dedicated for the key and the message that results in a *proper* code.

2) *Decoder*: We present a unified SC superdecoder for the coding scheme described above that corrects  $\rho \leq D_{\min} - 1$  erasures, where  $D_{\min} = 2^{s-r}$ , and recovers the message given the shared key. The recursive decoder takes the received bit sequence  $y_1^n$ , the shared key  $\mathbf{k}$ , key indices  $\mathcal{A}^c$ , code parameters  $s, r$ , and a recursion parameter  $j$  as inputs. Initially,  $j = 1$ . It outputs  $h_1^n$ , i.e., which is equal to the codeword  $\mathbf{c}$  provided that  $\rho \leq D_{\min} - 1$ , as well as  $u_1^n = \pi(\mathbf{k}, \mathbf{m})$ , which is used to retrieve the message  $\mathbf{m}$ , and a recursion index  $j'$  used to track the index of the last decoded bit. A pseudocode for the decoder is shown in Algorithm 2. The following claim shows the success of the described decoder.

*Claim 12*: The proposed unified RM-based coding scheme together with the unified SC superdecoder in Algorithm 2 successfully retrieves the message as long as  $\rho \leq D_{\min} - 1$ .

*Proof*: Let the received sequence be denoted by  $y_1^n$  which has at most  $\rho$  erasures. Let also the key bits be denoted by  $\mathbf{k}$  which are assigned to entries of  $\mathbf{u}$  indexed by elements of  $\mathcal{A}^c$ . We use induction on the parameter  $s$  of the underlying RM code of length  $2^s$  to prove the claim. The induction hypothesis is that the decoder is successful for any RM-based coding scheme of length  $2^s$  with some parameter  $r \leq s$ , and a key with size  $\sum_{i=r+1}^s \binom{s}{i}$ , assuming there are at most  $\rho = 2^{s-r} - 1$  erasures. The induction base is  $s = 0$ , for which the induction hypothesis is trivial. Now, suppose that the induction hypothesis holds for  $s$  and we want to show it for  $s + 1$ .

**Case 1:**  $r = 0$ , i.e., we have an RM( $s+1, 0$ ) which becomes a repetition code of length  $n = 2^{s+1}$ . In this case,  $\tilde{\mathbf{G}}(s+1, 0)$

---

**Algorithm 2** Unified SC decoder for binary erasures (DecBE)

---

```

1: Input:  $\mathbf{k}, y_1^n, \mathcal{A}^c, s, r, j$ .
2: Output:  $h_1^n, u_1^n, j'$ .
3: if  $r = 0$  then
4:    $\mathcal{I} = [j, j+1, \dots, j+2^s-1]$ 
5:    $i_1 \leftarrow$  index of any non-erasure bit in  $y_1^n$ .
6:   for  $i \in \mathcal{A}^c$  do
7:      $u_i = k_i$ 
8:   end for
9:    $i' \in \mathcal{I} \setminus \mathcal{A}^c$ 
10:   $u_{i'} = y_{i_1} \oplus_{i \in \mathcal{A}^c} u_i$ 
11:   $h_1^n = [y_{i_1}, y_{i_1}, \dots, y_{i_1}]$ 
12:   $j' = j + 2^s - 1$ 
13: else
14:    $\bar{\mathbf{y}} = y_1^{n/2} \oplus y_{n/2+1}^n$ 
15:    $\mathbf{h}'_1, u_1^{n/2}, j'_1 \leftarrow \text{DecBE}(\mathbf{k}_1, \bar{\mathbf{y}}, \mathcal{A}_1^c, s-1, r-1, j)$ 
16:    $\mathbf{h}'_2 = u_1^{n/2}\tilde{\mathbf{G}}(s-1, r)$ 
17:    $\mathbf{h}' = [\mathbf{h}'_1 \oplus \mathbf{h}'_2, \mathbf{h}'_2]$ 
18:    $\tilde{y}_1^n = y_1^n \oplus \mathbf{h}' = [\tilde{y}_1, \tilde{y}_2]$ 
19:    $l = \underset{j \in 1, 2}{\text{argmin}} (\text{number of erasures in } \tilde{y}_j)$ 
20:    $\mathbf{h}''_1, u_{n/2+1}^n, j' \leftarrow \text{DecBE}(\mathbf{k}_2, \tilde{y}_l, \mathcal{A}_2^c, s-1, r, j'_1 + 1)$ 
21:    $\mathbf{h}'' = [\mathbf{h}''_1, \mathbf{h}''_1]$ 
22:    $h_1^n = \mathbf{h}' \oplus \mathbf{h}''$ 
23: end if
24: return  $u_1^n, h_1^n, j'$ 

```

---

is the all-ones matrix and the entries of codeword are all equal to the sum of entries in  $\mathbf{u}$ . Note that the number of message bits is  $m = \sum_{i=0}^r \binom{s+1}{i} = 1$  and we have  $2^{s+1} - 1$  key bits. Also, the maximum number of erasures the code can correct is  $2^{s+1} - 1$ . Hence, the decoder successfully retrieves the message bit using the non-erasure symbols, which there is at least one, in  $y_1^n$ . Suppose that the non-erasure bit is indexed by  $i_1$ . Since the locations of the key bits are known, we can place them at their respective locations retrieving  $u_i$ 's for all  $i \in \mathcal{A}^c$ . Next, the message bit located at  $i'$  is retrieved as  $u_{i'} = y_{i_1} \oplus_{i \in \mathcal{A}^c} u_i$ , and the corresponding codeword is also retrieved correctly. Hence, the decoder is successful. Note that this case corresponds to lines 4-12 of Algorithm 2.

**Case 2:**  $r > 0$ . The code length is  $n = 2^{s+1}$  and the

key length is  $\sum_{i=r+1}^{s+1} \binom{s+1}{i}$ . We split the key indices into two parts, namely  $\mathcal{A}_1^c$  and  $\mathcal{A}_2^c$ , representing the key bits  $k_1$  and  $k_2$  in the first and the second half sub-blocks of  $\mathbf{u}$ , respectively. The lengths of  $k_1$  and  $k_2$  are  $|\mathcal{A}_1^c| = \sum_{i=r}^s \binom{s}{i}$  and  $|\mathcal{A}_2^c| = \sum_{i=r+1}^s \binom{s}{i}$ , respectively, due to the aforementioned choice of indices. The decoder first computes  $\tilde{\mathbf{y}} = \mathbf{y}_1^{n/2} \oplus \mathbf{y}_{n/2+1}^n$  which will have at most  $2^{s+1-r} - 1$  erasures. It then passes this to the left child, in the binary tree representation terminology discussed earlier, along with  $k_1$  and the set of its corresponding indices  $\mathcal{A}_1^c$ . The left child decodes a codeword of length  $n' = 2^s$  using a code with parameter  $r' = r - 1 \geq 0$ , which can correct up to  $2^{s-r'} - 1 = 2^{s+1-r} - 1$  erasures, and retrieves the message bits in  $u_1^{n/2}$  given the key  $k_1$  of length  $\sum_{i=r'+1}^s \binom{s}{i} = \sum_{i=r}^s \binom{s}{i}$ . The decoder on the left child is successful by induction hypothesis. It outputs  $u_1^{n/2}$  and  $\mathbf{h}'_1$ . After that, the decoder computes  $\mathbf{h}'_2 = u_1^{n/2} \tilde{\mathbf{G}}(s, r)$  followed by  $\mathbf{h}' = [\mathbf{h}'_1 \oplus \mathbf{h}'_2, \mathbf{h}'_2]$ . Then, the decoder computes  $\tilde{\mathbf{y}}_1^n = \mathbf{y}_1^n \oplus \mathbf{h}'$  and chooses either  $\tilde{\mathbf{y}}_1^{n/2}$  or  $\tilde{\mathbf{y}}_{n/2+1}^n$ , whichever has a smaller number of erasures, and passes it to the right child together with  $k_2$  and the corresponding set of indices  $\mathcal{A}_2^c$ . The number of erasures in what is passed to this child is at most  $2^{s+1-r}/2 - 1 = 2^{s-r} - 1$ , and the length of the key is  $\sum_{i=r+1}^s \binom{s}{i}$ . The decoder on this child decodes a codeword of length  $n' = 2^s$  using a code with parameter  $r' = r > 0$ , which can correct up to  $2^{s-r'} - 1 = 2^{s-r} - 1$  erasures and retrieves the message bits in  $u_{n/2+1}^n$  using the key  $k_2$  of length  $\sum_{i=r'+1}^s \binom{s}{i} = \sum_{i=r+1}^s \binom{s}{i}$ . Decoding here is also successful by induction hypothesis. It outputs  $u_{n/2+1}^n$  and  $\mathbf{h}''_1$ . The overall decoder then computes  $\mathbf{h}'' = [\mathbf{h}''_1, \mathbf{h}''_1]$  and outputs  $\mathbf{h}_1^n = \mathbf{h}' \oplus \mathbf{h}''$  and  $u_1^n$ . Hence,  $u_1^n$  is retrieved and the proof is complete.  $\blacksquare$

## VI. CONCLUSION

In this work, we propose a model for threshold-secure coding with a shared key such that specific conditions for reliability and security based on information-theoretic measures are met. The specification of such model includes a threshold parameter which is to be designed based on the application for such coding schemes. Also, methods for utilizing error-correcting linear block codes in constructing threshold-secure coding schemes are discussed, where the parameter  $t$  of the threshold-secure scheme is shown to be directly related to the minimum distance of the underlying linear block code. Furthermore, a coding scheme based on Reed-Muller codes is described. Its encoding is done recursively and is shown to satisfy the conditions for a *proper* code. Moreover, a setup taking into account the noise in the communication channel between legitimate parties is considered. Then, a robust and threshold-secure coding scheme, based on code concatenation, is suggested for general channels. Also, a unified coding scheme built upon Reed-Muller codes for both threshold security and robustness in the presence of erasures is described.

A possible direction for future work is to design coding schemes based on punctured Reed-Muller codes to allow for more flexible rates. To this end, ideas from punctured schemes for closely related polar codes can be useful [42], [43]. Also, it is interesting to explore whether unified coding schemes for threshold security and robustness, similar to the RM-based scheme presented in Section V-B, can be constructed from other well-known families of codes. Another possible direction of future work is to study threshold security in settings with wiretap channels, where the eavesdropper's channel is also noisy. Also, extending the considered setup to multi-user scenarios, as in wiretap multiple access [44] or as in multi-user secret sharing setups [28], is another interesting future direction.

## REFERENCES

- [1] N. Aldaghri and H. Mahdavifar, "Threshold-secure coding with shared key," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2019, pp. 552–559.
- [2] A. D. Wyner, "The wire-tap channel," *Bell system technical journal*, vol. 54, no. 8, pp. 1355–1387, 1975.
- [3] P. K. Gopala, L. Lai, and H. El Gamal, "On the secrecy capacity of fading channels," in *2007 IEEE International Symposium on Information Theory*. IEEE, 2007, pp. 1306–1310.
- [4] F. Oggier and B. Hassibi, "The secrecy capacity of the MIMO wiretap channel," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 4961–4972, 2011.
- [5] A. Thangaraj, S. Dihidar, A. R. Calderbank, S. W. McLaughlin, and J.-M. Merolla, "Applications of LDPC codes to the wiretap channel," *IEEE Transactions on Information Theory*, vol. 53, no. 8, pp. 2933–2945, 2007.
- [6] H. Mahdavifar and A. Vardy, "Achieving the secrecy capacity of wiretap channels using polar codes," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6428–6443, 2011.
- [7] U. M. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, 1993.
- [8] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *Proceedings of the 14th ACM International Conference on Mobile computing and Networking*. ACM, 2008, pp. 128–139.
- [9] N. Aldaghri and H. Mahdavifar, "Physical layer secret key generation in static environments," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2692–2705, 2020.
- [10] N. Ebrahimi, H. Mahdavifar, and E. Afshari, "A novel approach to secure communication in physical layer via coupled dynamical systems," *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, 2018.
- [11] W. Kang and N. Liu, "Wiretap channel with shared key," in *2010 IEEE Information Theory Workshop*. IEEE, 2010, pp. 1–5.
- [12] W. Zhang, C. Zhang, W. Jin, C. Chen, N. Jiang, and K. Qiu, "Chaos coding-based QAM IQ-encryption for improved security in OFDMA-PON," *IEEE Photonics Technology Letters*, vol. 26, no. 19, pp. 1964–1967, 2014.
- [13] F. Huo and G. Gong, "A new efficient physical layer OFDM encryption scheme," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 1024–1032.
- [14] J. Zhang, A. Marshall, R. Woods, and T. Q. Duong, "Design of an OFDM physical layer encryption scheme," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2114–2127, 2017.
- [15] M. Tahir, S. P. Jarot, and M. Siddiqi, "Wireless physical layer security using encryption and channel pre-compensation," in *2010 International Conference on Computer Applications and Industrial Electronics*. IEEE, 2010, pp. 304–309.

- [16] A. Zuquete and J. Barros, "Physical-layer encryption with stream ciphers," in *2008 IEEE International Symposium on Information Theory*. IEEE, 2008, pp. 106–110.
- [17] X. Zhou and M. R. McKay, "Physical layer security with artificial noise: Secrecy capacity and optimal power allocation," in *2009 3rd International Conference on Signal Processing and Communication Systems*. IEEE, 2009, pp. 1–5.
- [18] S. Goel and R. Negi, "Guaranteeing secrecy using artificial noise," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2180–2189, 2008.
- [19] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," *Coding Thv*, vol. 4244, pp. 114–116, 1978.
- [20] C. N. Mathur, K. Narayan, and K. Subbalakshmi, "High diffusion cipher: Encryption and error correction in a single cryptographic primitive," in *International Conference on Applied Cryptography and Network Security*. Springer, 2006, pp. 309–324.
- [21] Y.-S. Kim, J.-H. Kim, and S.-H. Kim, "A secure information transmission scheme with a secret key based on polar coding," *IEEE Communications Letters*, vol. 18, no. 6, pp. 937–940, 2014.
- [22] N. Cai and R. W. Yeung, "Secure network coding," in *Proceedings IEEE International Symposium on Information Theory*. IEEE, 2002, p. 323.
- [23] K. Bhattacharjee, K. R. Narayanan *et al.*, "Weakly secure network coding," in *Proceedings First Workshop on Network Coding, Theory, and Applications (NetCod)*, 2005.
- [24] S. H. Dau, V. Skachek, and Y. M. Chee, "On the security of index coding with side information," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3975–3988, 2012.
- [25] L. Ong, B. N. Vellambi, P. L. Yeoh, J. Kliewer, and J. Yuan, "Secure index coding: Existence and construction," in *2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2016, pp. 2834–2838.
- [26] P. F. Oliveira, L. Lima, T. T. Vinhoza, J. Barros, and M. Médard, "Coding for trusted storage in untrusted networks," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1890–1899, 2012.
- [27] J. Y. Chung, C. Joe-Wong, S. Ha, J. W.-K. Hong, and M. Chiang, "Cyrus: Towards client-defined cloud storage," in *Proceedings of the Tenth European Conference on Computer Systems*, 2015, pp. 1–16.
- [28] M. Soleymani and H. Mahdavifar, "Distributed multi-user secret sharing," *IEEE Transactions on Information Theory*, 2020.
- [29] I. S. Reed, "A class of multiple-error-correcting codes and the decoding scheme," MIT Lexington Lincoln Lab, Tech. Rep., 1953.
- [30] G. D. Forney, "Concatenated codes." 1965.
- [31] D. R. Stinson, "Something about all or nothing (transforms)," *Designs, Codes and Cryptography*, vol. 22, no. 2, pp. 133–138, 2001.
- [32] N. N. Esfahani, I. Goldberg, and D. R. Stinson, "Some results on the existence of  $t$ -all-or-nothing transforms over arbitrary alphabets," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 3136–3143, 2017.
- [33] Y. Sutcu, H. T. Sencar, and N. Memon, "A secure biometric authentication scheme based on robust hashing," in *Proceedings of the 7th workshop on Multimedia and security*, 2005, pp. 111–116.
- [34] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.
- [35] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proceedings of ICC'93-IEEE International Conference on Communications*, vol. 2. IEEE, 1993, pp. 1064–1070.
- [36] R. Roth, *Introduction to coding theory*. Cambridge University Press, 2006.
- [37] Å. Björck and V. Pereyra, "Solution of Vandermonde systems of equations," *Mathematics of computation*, vol. 24, no. 112, pp. 893–903, 1970.
- [38] G. Schnabl and M. Bossert, "Soft-decision decoding of Reed-Muller codes as generalized multiple concatenated codes," *IEEE Transactions on Information Theory*, vol. 41, no. 1, pp. 304–308, 1995.
- [39] I. Dumer, "Recursive decoding and its performance for low-rate Reed-Muller codes," *IEEE Transactions on Information Theory*, vol. 50, no. 5, pp. 811–823, 2004.
- [40] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.
- [41] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "Fast multi-dimensional polar encoding and decoding," in *2014 Information Theory and Applications Workshop (ITA)*. IEEE, 2014, pp. 1–5.
- [42] D.-M. Shin, S.-C. Lim, and K. Yang, "Design of length-compatible polar codes based on the reduction of polarizing matrices," *IEEE Transactions on Communications*, vol. 61, no. 7, pp. 2593–2599, 2013.
- [43] M. El-Khamy, H.-P. Lin, J. Lee, H. Mahdavifar, and I. Kang, "HARQ rate-compatible polar codes for wireless channels," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.
- [44] E. Tekin and A. Yener, "The Gaussian multiple access wire-tap channel," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5747–5755, 2008.



**Nasser Aldaghri** (S'18) received the B.S. degree in electrical engineering from King Saud University, Riyadh, Saudi Arabia, in 2014, and the M.S. degree in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2017, where he is currently pursuing his Ph.D. degree. His research interests include information theoretic security, physical layer security, and coding theory.



**Hessam Mahdavifar** (S'10, M'12) is an Assistant Professor in the Department of Electrical Engineering and Computer Science at the University of Michigan Ann Arbor. He received the B.Sc. degree from the Sharif University of Technology, Tehran, Iran, in 2007, and the M.Sc. and the Ph.D. degrees from the University of California San Diego (UCSD), La Jolla, in 2009, and 2012, respectively, all in electrical engineering. He was with the Samsung US R&D between 2012 and 2016, in San Diego, US, as a staff research engineer.

He received the NSF career award in 2020. He also received Best Paper Award in 2015 IEEE International Conference on RFID, and the 2013 Samsung Best Paper Award. He also received two Silver Medals at International Mathematical Olympiad in 2002 and 2003, and two Gold Medals at Iran National Mathematical Olympiad in 2001 and 2002. His main area of research is coding and information theory with applications to wireless communications, storage systems, security, and privacy.