Codes for Identification via Channels: Tutorial for Communications Generalists

Caspar von Lengerke, Juan A. Cabrera, Martin Reisslein *Fellow, IEEE*, and Frank H.P. Fitzek *Fellow, IEEE*

***Abstract*—Identification via channels (ID) is a communication problem in which a receiver tries to discern whether a received message matches a previously selected message. With identification, the receiver tries to answer the yes-no question "Is this my message?"; whereas, with message transmission, the receiver tries to answer the question "What message is this?", which has many possible answers. Information and coding theory show that for identification, specialized codes can achieve efficiency gains of an exponential order compared to full transmission of messages. This tutorial gives an introduction to identification via channels and to codes for identification via channels, for communication generalists with an interest in this traction-gaining topic. Specifically, a receiver can identify a message reliably from a received noisy-ID codeword. A noisy-ID codeword can be constructed by concatenating a linear block code that mitigates channel distortion with a noiseless-ID codeword that encodes a message efficiently for reliable identification. Noiseless-ID codes can be implemented as tagging codes or constant-weight codes that both guarantee low collision probabilities due to the distance properties of their underlying linear codes, such as Reed-Solomon codes, Reed-Muller codes, or random linear codes. We revisit and explain the close relationship between noiseless-ID codes and universal hash functions. ID is a ubiquitous communication problem and is relevant to all scenarios where two parties aim to determine whether two pieces of data are exactly identical. Specific non-cryptographic use cases include determining data integrity and state consistency in digital twins.**

***Index Terms*—Coding theory, Goal-oriented communication, Identification via channels, Linear codes, Post-Shannon, Semantic communication, Universal hashing.**

I. Introduction

This work was supported in part by the Federal Ministry of Education and Research of Germany in the programme of "Souverän. Digital. Vernetzt.". Joint project 6G-life, project identification number: 16KISK001K, in part by the German Research Foundation [Deutsche Forschungsgemeinschaft (DFG)] under Project 450566247, and in part by the German Research Foundation as part of Germany's Excellence Strategy – EXC 2050/1 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden under Project 390696704.

C. von Lengerke is with the Deutsche Telekom Chair of Communication Networks, Technische Universität Dresden, 01062 Dresden, Germany, with the Centre for Tactile Internet with Human-in-the-Loop (CeTI), Technische Universität Dresden, Germany, and also with Ecologic-Computing GmbH, Dresden, Germany. (e-mail:caspar.lengerke@tu-dresden.de)

J.A. Cabrera is with the Deutsche Telekom Chair of Communication Networks, Technische Universität Dresden, 01062 Dresden, Germany and also with Ecologic-Computing GmbH, Dresden, Germany. (e-mail:juan.cabrera@tu-dresden.de)

M. Reisslein is with the School of Electrical, Computer, and Energy Eng., Arizona State University, Tempe, AZ 85287-5706, USA, (e-mail: reisslein@asu.edu)

F.H.P. Fitzek is with the Deutsche Telekom Chair of Communication Networks, Technische Universität Dresden, 01062 Dresden, Germany and also with the Centre for Tactile Internet with Human-in-the-Loop (CeTI), Technische Universität Dresden, Germany, (e-mail: frank.fitzek@tu-dresden.de)

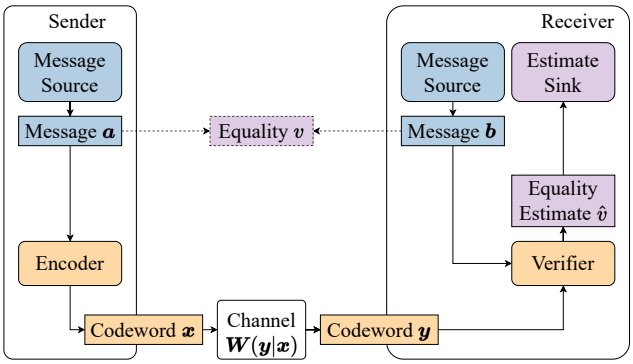


Fig. 1. Schematic overview of identification via channels. The receiver estimates whether the messages of the sender and the receiver are equal.

IDENTIFICATION via channels (ID via channels) is a communication problem involving two parties that communicate via a channel to estimate whether their respective messages equal each other [1]. Consider a sender called Alice and a receiver called Bob. Alice and Bob both select a message from their respective source, individually, i.e., Alice selects a message $\boldsymbol{a}$ and Bob selects a message $\boldsymbol{b}$, as visualized in Fig. 1 Alice can communicate a codeword to Bob over the channel. Based on the codeword that Alice sent to Bob via the channel, Bob determines whether the two messages $\boldsymbol{a}$ and $\boldsymbol{b}$ are identical or not. In other words, Bob determines an estimate of the *equality* of the two messages, i.e., whether $\boldsymbol{a} = \boldsymbol{b}$. Finally, Bob forwards the equality estimate into the sink, thus achieving the communication goal and completing the communication.

Message ID is a ubiquitous task. For example, checksums, cyclic redundancy checks (CRCs), and hash functions are widely used in heterogeneous use cases to verify whether two messages are identical. Checksums and CRCs typically determine whether a message was correctly copied (reproduced), for example, for packets that are transported via a network or to verify that a human entered a credit card number without typos into an input field. Cryptographic hash functions can be used to verify that two messages are identical and that no one maliciously manipulated the result of such a verification. Codes for identification via channels are closely related to these functions that "hash" a message into a small codeword.

Identification via channels considers the verification whether two messages are identical for *finite message sizes* via a *distorting channel*. Whereby, for this tutorial, we consider a Discrete Memoryless Channel (DMC) to "distort" symbols of

the sender's codeword so that the received symbols can differ from the sent symbols. A code that addresses the ID problem provides *information-theoretic error probability guarantees* including limits on the collision probability for *accidental collisions* of the "hashes" (codewords) of message pairs (in contrast to cryptographic hash functions that are concerned with adversarial attacks via purposeful collisions).

A. Goal-Oriented Communication

ID is a communication goal, whereby the goal of a communication refers to the result that the communicating parties aim to determine (e.g., the equality estimate in ID) and the likelihood of a correct result, i.e., reliability. The goal can also entail how much data the parties send via the channel, i.e., efficiency. In present communication systems, the goal of a communication is not always included in the code design. Rather, a general-purpose (goal-agnostic) channel code provides a reasonably efficient, well-understood method to transmit a message reliably via a noisy channel. In other words, it is possible to address arbitrary communication goals by using channel codes to reliably reproduce the sender's message at the receiver as Claude Shannon demonstrated in [2]. This abstraction of arbitrary communication goals to the "technical problem" of message transmission [3] facilitated the development of modern telecommunication technology because it has focused on channel codes (that enable reliable message transmission) as a general-purpose method for reliable communication. However, further improvement in the efficiency of general-purpose channel codes faces diminishing returns since the wide adoption of Turbo, LDPC, and Polar codes [4]–[14] that are capacity-approaching channel codes. The capacity of a channel is a limit on how efficient a code can be while maintaining high reliability. At capacity, it is impossible to improve the efficiency of a code, i.e., the code rate, without deteriorating the reliability of the communication. Thereby, reliability and efficiency compete with each other, i.e., there is a trade-off between the two.

To improve communication efficiency beyond the limits of capacity-approaching channel codes, finding tailored solutions for specific communication goals, such as ID, has gained interest, e.g., [15]–[17]. Taking into account the specific communication goal, specialized goal-oriented codes offer the potential to increase efficiency (via better achievable code rates) beyond the (Shannon) limit that channel codes are subject to. Goal-oriented codes are not meant to replace channel codes. Rather, goal-oriented codes address specific goals and are typically not applicable to other communication goals. Finding more efficient goal-oriented codes can (for specific goals) significantly reduce the amount of data that is transmitted over channels and networks. ID is one specific communication goal that has received growing attention [18]–[20], partly because goal-oriented codes can address the identification goal up to an exponential order more efficiently than general-purpose channel codes [1], [21]. Research on ID has evolved independently from goal-oriented communication, but aims in the same direction and can be considered a goal-oriented concept.

B. Codes for Identification via Channels

While this tutorial focuses on goal-oriented codes for ID via channels, general-purpose channel codes act as a baseline to emphasize the efficiency gains of the goal-oriented codes. Any code that addresses ID via channels should create codewords that are able to overcome the distortion of the noisy channel and enable reliable estimation of the equality of the messages $\boldsymbol{a}$ and $\boldsymbol{b}$ of the sender and the receiver. A channel code achieves this by enabling reliable *message transmission* via the noisy channel. A channel code, such as a linear block code [22], [23], enables reliable message transmission over a noisy channel by mapping a message $\boldsymbol{a}$ from a set $\mathcal{U}$ of messages to a channel codeword $\boldsymbol{x}$ from a larger set $\mathcal{X}$ of channel codewords, thereby adding redundancy to the message $\boldsymbol{a}$, as Fig. 2 visualizes. In other words, the transmitted channel codeword $\boldsymbol{x}$ is larger than the message $\boldsymbol{a}$ that is encoded into the channel codeword $\boldsymbol{x}$. Decoding the linear block (channel) codeword $\boldsymbol{y} \in \mathcal{Y}$ yields an estimate $\hat{\boldsymbol{a}}$ of the message $\boldsymbol{a}$ that the sink accepts. In the framework of goal-oriented communication, Bob has thereby achieved the (general-purpose) goal of message transmission because the sink obtains an estimate $\hat{\boldsymbol{a}}$ of the message $\boldsymbol{a}$. By enabling reliable message transmission, a channel code creates a quasi-noiseless (reliable) channel, cf. Fig. 2. The quasi-noiseless channel makes it possible to address communication goals without regard for the error-prone nature of the underlying channel. By *transmitting* the message $\boldsymbol{a}$ from Alice to Bob, all properties of the message $\boldsymbol{a}$ are known to Bob. Thereby, Bob can use the reproduced message $\boldsymbol{a}$ to achieve also any other communication goal (aside from the achieved transmission goal).

However, in ID via channels, Bob must learn only one property: the equality of Alice's message $\boldsymbol{a}$ to Bob's message $\boldsymbol{b}$. The equality v is a binary property that can either be true v_p (when the messages are identical) or false v_n (when the messages are different). Therefore, the equality v can be represented using a single bit, i.e., the equality v is a one-bit property. Figure 2 visualizes ID via channels in comparison with the message transmission problem. As Ja Ja established in [27], the ID goal (finding the equality estimate $\hat{v}$) is an easier communication goal than the message transmission goal because the receiver has fewer results to select from. In other words, the output of message transmission (i.e., the message estimate $\hat{\boldsymbol{a}} \in \mathcal{U}$) is significantly larger than the binary output of the ID goal (i.e., the equality estimate $\hat{v} \in \{\hat{v}_\mathrm{p}, \hat{v}_\mathrm{n}\}$). Because the equality v and its estimate $\hat{v}$ are one-bit properties, the identification goal can be achieved much more efficiently than by using general-purpose channel codes [1], [28], [29]. Channel codes address the more difficult goal of transmission instead of directly addressing the simpler goal of ID.

Goal-oriented codes that are tailored for ID via channels can be more efficient than channel codes. We refer to these goal-oriented codes as *noisy-ID codes*. In comparison to a linear block (channel) code, a noisy-ID code enables reliable identification for a much larger number of messages as Fig. 2 visualizes. Alternatively, for a given number of messages, a noisy-ID code can enable reliable identification using much

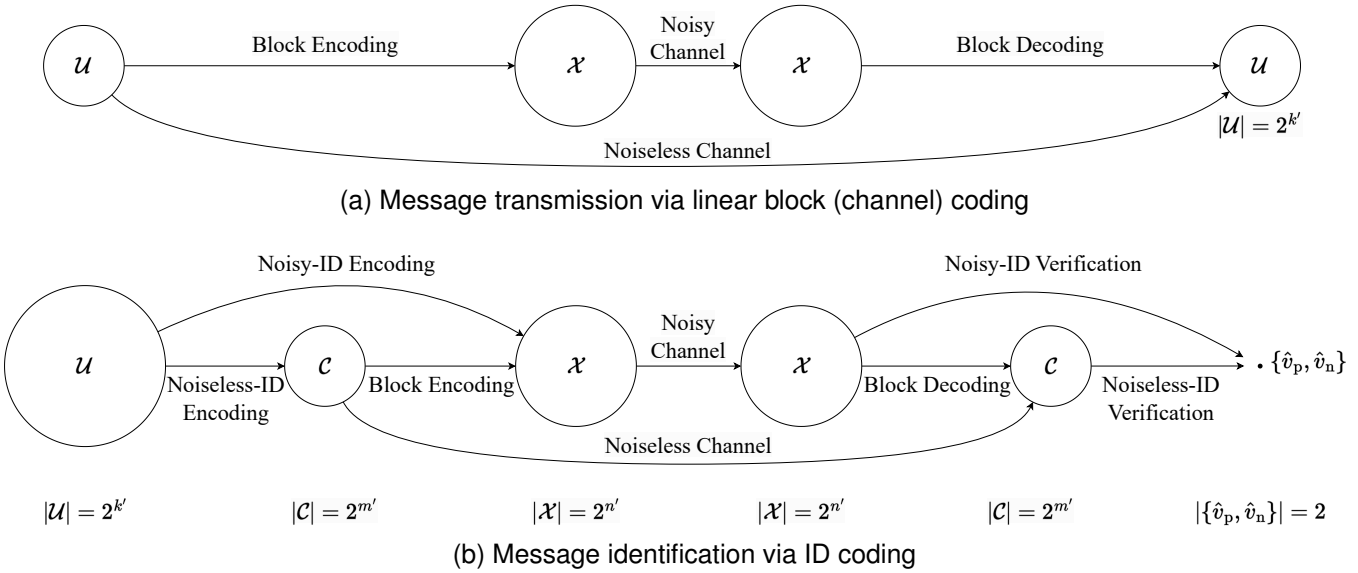


Fig. 2. Comparison of linear block (channel) coding for message transmission (top) with noisy-ID coding for message identification (bottom). The size of the circles represents the size of the represented set and thereby the size in bit of the respective set's elements. The figure visualizes the set $\mathcal{U}$ of messages of size k', the set $\mathcal{X}$ of channel codewords of size n', and the set $\mathcal{C}$ of noiseless-ID codewords of size m'.

TABLE I
OVERVIEW OF CODES THAT THIS TUTORIAL REFERS TO.

Name	Section	Description	Example
Channel Codes	III	Enable reliable transmission via noisy channels, aim to create virtual noiseless channel	Linear block codes, Convolutional codes
Linear Block Codes	III	Can be used as channel codes and to construct tagging codes	Reed-Solomon (RS), Reed-Muller (RM), Polar codes
Noisy-ID Codes	IV	Enable reliable ID via noisy channels, special-purpose codes only for ID problem	Randomized and deterministic noisy-ID codes
Rand. Noisy-ID Codes	IV	Noisy-ID codes with randomized encoding, up to exponential efficiency gain over channel codes in ID problem	Concatenated noisy-ID codes
Det. Noisy-ID Codes	IV-H	Noisy-ID codes with deterministic encoding, efficiency gains over channel codes in ID problem	Single parity check (SPC) concatenated with Reed-Solomon (RS) code [24]
Concat. Noisy-ID Codes	V, IX	Randomized noisy-ID codes constructed by concatenating a channel code with a noiseless-ID code	Polar code concatenated with RS2 ID (tagging) code [25]
Noiseless-ID Codes	VI	Randomized codes that enable reliable ID via noiseless channels, related to universal hash functions	Tagging Codes, Constant Weight Codes
Tagging Codes	VII	Noiseless-ID codes based on linear block codes	"Concatenated" RS (RS2) ID Code [21], [26]
Constant Weight Codes	VIII	Can be used as noiseless-ID codes; then, typically constructed from tagging codes	RS2 Pulse Position Modulation (RS2-PPM) ID Code [21]

smaller codewords than a linear block code. In other words, noisy-ID codes can address ID via channels much more efficiently than block codes at comparable reliability. The increase in the number of supported messages comes at the cost of an additional error probability, whereby this additional error probability is limited and thereby asymptotically does not deteriorate reliability. Because ID is a simpler goal than transmission, it is possible for the noisy-ID encoder to encode the message $\boldsymbol{a} \in \mathcal{U}$ to a smaller channel codeword $\boldsymbol{x}$ that suffices to reliably determine the equality estimate $\hat{v}$ at the receiver, whereby "reliably" means "with a small, limited error probability". As we will explain in Section V, it is possible to construct noisy-ID codes by concatenating a linear block code (that addresses the distortion of the noisy channel to provide reliability) with an *ID code* that addresses the verification step of ID efficiently and reliably. The linear block code can be considered to form a noiseless channel such that the noiseless-ID code can communicate over a noiseless channel, cf. Fig. 2. In other words, a *channel code* addresses the channel, and an *ID code* addresses ID via *noiseless channels*. This separation transforms ID via noisy channels (the noisy-ID problem) into two sub-problems: the transmission goal (that channel codes address), and the goal of ID via noiseless channels (the noiseless-ID problem that noiseless-ID codes address). With this separation, it is possible to partially rely on established channel codes (that profit from an established theory, and efficient implementations) to construct noisy-ID codes that can much more efficiently address ID via channels. The literature does not always distinguish clearly between noiseless-ID coding and noisy-ID coding. Both noiseless-ID coding and noisy-ID coding are referred to as "ID coding" in the literature.

When concatenated, a linear block (channel) code and a noiseless-ID code can form a noisy-ID code that addresses ID via noisy channels. Since the concatenation of linear block codes with noiseless-ID codes constitutes noisy-ID codes, both "sub-codes" (linear block codes and noiseless-ID codes) are covered in this tutorial next to the overall noisy-ID code itself that the two "sub-codes" constitute. In a joint coding effort, noisy-ID codes address both the noisy channel and the ID problem, i.e., the overall ID via (noisy) channels problem. Noisy-ID codes are goal-oriented codes tailored to reliable message *identification* and are not suited for reliable message *transmission* (in contrast to general-purpose channel codes) nor any other communication goal that is more general than identification.

C. Reliable, Efficient Identification via ID Codes

ID codes aim to address the ID problem via noiseless channels efficiently. To improve efficiency, the ID encoding reduces the information content of the message, thereby introducing a limited error probability. The receiver verifies the equality of two messages directly based on the noiseless-ID codeword that encodes the message.

ID coding closely relates to functions, such as checksums, cyclic redundancy checks (CRCs), and (non-cryptographic) hash functions. These functions have in common: i) the encoding of a message to a shorter codeword, and ii) a comparison operation (verification step) involving that codeword (e.g., the checksum, the check value, the hash, or the digest). Additionally, these functions remove information from the message such that several messages map to the same codeword. Hence, these functions (including noiseless-ID codes) introduce a small error probability associated with the "collision" of differing messages in the same codeword ("hash"). In contrast to most checksums, CRCs, and hash functions, the noiseless-ID codes limit the collision probability of differing messages not only via the average collision probability but also via a *worst-case collision probability bound*. Specifically, randomized noiseless-ID codes are closely related to universal hash functions [30].

Because the encoding of (noisy-)ID codes removes information from the message it is impossible to *reliably* determine the encoded message from the codeword. However, it is still possible to reliably determine from the codeword an estimate of the equality of two messages, i.e., to reliably address the ID problem. If the receiver tries to decode the received noiseless-ID codeword into a message estimate instead, then the receiver determines a (possibly infinite) list of messages that the sender may have selected. Thereby, the verification step in ID coding can be framed as list decoding with (possibly) infinite list size.

To reproduce a message reliably, Claude Shannon's source coding theorem [2] formulates the minimum amount of information that has to be conveyed to the receiver. Goal-oriented codes can go below this limit if the specific communication goal differs from reproducing the sender's message at the receiver. For such goals, it can suffice to provide the receiver with less information than the sender's message holds. An efficient goal-oriented code thereby removes some information from the message in the encoding step whereby the information that remains in the codeword suffices to reliably address the specific communication goal.

TABLE II
OVERVIEW OF RELATED SURVEY AND TUTORIAL LITERATURE.

Semantic and Goal-Oriented Communication:	
General Principles	[31]–[48]
In Wireless Networks	[49]–[64]
Security, Privacy, and Trustworthiness	[65]–[70]
For Specific Network Types	[71]–[74]
Edge Computing	[75], [76]
Signal Processing	[77], [78]
Digital Twins	[79]
Semantic Metaverse	[80]–[83]
Artificial Intelligence	[84]–[90]
Literature with ID-specific Tutorial Content:	
Background on Constant-Weight Codes for Noiseless-ID	[91]
Background on Tagging and Constant-Weight Codes	[26]
Information-Theoretic Background on Rand. Noisy-ID	[29]
Background on Tagging Codes and Noisy-ID	[92]
Information-Theoretic Background on Det. Noisy-ID	[93]
Review of Constant-Weight Codes for Noiseless-ID	[94]
Background on Tagging Codes for Noiseless-ID	[95]

D. Related Survey and Tutorial Literature

The broad field of communication mechanisms at the semantic and goal levels [3], which encompass the sub-field of identification via channels, has been covered in several surveys in recent years, which are all orthogonal to our tutorial article. In order to contrast our tutorial article from the existing survey and tutorial literature, we first give a brief overview of the related surveys on semantic and goal-oriented communication in Section I-D1. We then contrast our tutorial from the existing publications that contain tutorial content on identification via channels in Section I-D2.

1) Semantic and Goal-Oriented Communication: Shannon's framework [2], [3] separated the meaning (semantics) and the goal (effectiveness) of a communication from the technical problem of reliably reproducing a message at a receiver. To enable efficiency gains beyond the traditional Shannon limit, semantic and goal-oriented communication propose to investigate the meaning and the goal of a communication. ID belongs to the field of goal-oriented communication as Section I-A explains. Semantic and goal-oriented communication have also been referred to as Beyond Shannon [49], [58] or Post Shannon [92] communication.

The existing research studies on the general principles of the field of semantic and goal-oriented communication have been surveyed in [31]–[48]. Recently, several surveys have covered the existing research studies in semantic and goal-oriented communication in wireless networks, including fifth- and sixth-generation (5G and 6G) wireless systems [49]–[64]. A different group of surveys has covered the existing research studies on the security, privacy, and trustworthiness aspects of semantic communication [65]–[70]. Also, recent surveys have covered semantic communication for specific network types, such as Industry 4.0 networks [71], integrated space-air-ground-sea networks [72], unmanned aerial vehicle

networks [73], as well as broadly the Internet [74]. A few surveys have covered related specialty topics, such as edge computing [75], [76], signal processing [77], [78], digital twins [79], and the semantic metaverse [80]–[83]. A very recent trend in semantic communication has been the use of artificial intelligence [84], [85], including generative artificial intelligence models [86]–[90].

To the best of our knowledge, only one specific aspect of the topic of identification via channels has so far been covered in a survey. The topical review [94] covered the existing noiseless-ID codes, in particular, the constant-weight codes for ID. We contrast this tutorial in detail from the tutorial content in the topical review [94] in Section I-D2.

In contrast to these survey articles which have focused on covering the existing research studies on a wide range of aspects of semantic communication, our article is a tutorial, i.e., we do not cover existing research studies. Instead, this tutorial article focuses on conveying the fundamental principles of one specific form of goal-oriented communication, namely identification via channels, to communications generalists.

2) Articles with Tutorial Content on Identification via Channels: The existing ID literature includes research studies with significant background sections that aim to educate readers about the emerging topic of ID [26], [91], [93], [95]. There are also books with chapters about ID [29], [92]. In contrast to the existing literature, which is mostly mathematically oriented in its presentation, we espouse a broader presentation style that combines extensive graphical illustrations and text explanations with the mathematical formulas. With our richly illustrated presentation style that includes rigorous mathematical notations in the graphical illustrations, we explain mathematical concepts via extensive concrete examples to make the currently rather abstract research area of ID readily accessible for the wide community of communications generalists. Our tutorial is accessible with knowledge of elementary probability and statistics, and does not require specialized prior knowledge in information theory or coding theory.

Additionally, this tutorial for the first time comprehensively explains the ID problem from first principles in information theory while extending to software implementations of ID codes that use specific processor instruction sets. We explicitly visualize and explain the typically only implied construction of noisy-ID codes by concatenating block codes with noiseless-ID codes (such as tagging codes). We use coherent names and variable symbols throughout, thereby clarifying which quantities correspond to each other in different abstraction levels, representations, or implementations of ID codes. Furthermore, whenever applicable, this tutorial points out similarities between noiseless-ID codes and hash functions to facilitate a more intuitive understanding of noiseless-ID codes (and by extension, of noisy-ID codes) for generalist readers who are familiar with hash functions.

The research article [26] contains a detailed preliminaries section that explains noiseless-ID codes and, specifically, "concatenated" Reed-Solomon codes in a tutorial fashion. Similarly, the technical report [91] examines the construction of constant-weight ID codes using Reed-Solomon codes and gives a mathematical tutorial-style introduction to the topic area, albeit less detailed than in [26]. The research article [95] includes a tutorial-style introduction to randomized noiseless-ID codes in a mathematical fashion without any visualizations. In contrast to these existing tutorial-style introductions, we provide a comprehensive tutorial that provides a broad general introduction to ID and noisy-ID codes that can partially rely on randomized noiseless-ID codes. The research article [93] includes an extensive list of related work on ID in the introduction and a visualization of a deterministic noisy-ID code. Additionally, we visualize the more powerful randomized noisy-ID codes, while including rigorous mathematical notation in the graphical illustrations to facilitate the mental mapping from the illustrated concepts to the corresponding formal mathematical representation.

The book chapter [92] includes a tutorial-style section on noisy-ID codes and tagging codes. In contrast, in this tutorial, we cover the entire range from the information-theoretic definition of the ID problem to the implementation details involved in using specific CPU instruction sets. Also, this tutorial includes extensive additional explanations and visualizations on error types as well as on codeword sets in noisy-ID and in noiseless-ID. Importantly, in this tutorial, we explicitly explain the concatenation of a block code with a noiseless-ID code to create an efficient, reliable noisy-ID code in Section IX.

The book [29] provides tutorial-style background on traditional information theory before it introduces ID with respect to traditional Shannon information theory. The book [29] targets an information-theoretic audience and mostly lacks extensive explanation and visualization, relying on mathematical expressions instead.

The topical review [94] covers most noiseless-ID codes, specifically, constant-weight codes for ID. This tutorial is much broader in scope as Fig. 3 visualizes: the constant-weight codes for ID are a specific representation of tagging codes that *can* be used but are not required to construct noisy-ID codes by concatenation with a block code. In contrast to [94], in this tutorial we focus on explaining noisy-ID codes and their components from first principles. For the sake of completeness, in Section VIII in this tutorial we briefly cover constant-weight codes for ID to clarify their position within the larger field of noisy-ID codes.

E. Contributions and Structure of this Tutorial

Aided by extensive visualizations, this tutorial gives a comprehensive introduction to identification codes for communications and networking generalists. This tutorial does not require specialized expert knowledge in information theory or coding for message transmission.

The logical structure of Sections II to VII is graphically illustrated in Fig. 3. In Section II, we define the identification problem via noisy channels and explain the terminology that is used throughout this tutorial article. Furthermore, we describe the inputs and outputs of the identification process, namely the messages (that come from message sources) and the equality estimate that is delivered to the sink. In Section III, we describe how the sender and the receiver can use a linear

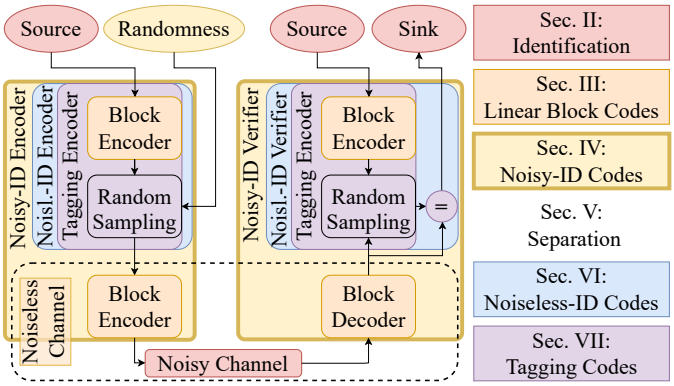


Fig. 3. Structure of Secs. II to VII with respect to their part within randomized noisy-ID codes.

block (channel) code (as widely used in message transmission) to overcome the channel distortion and correctly identify a message. This approach can be considered *ID via channel codes* and acts as a baseline for the main subject of this tutorial that are noisy-ID codes. More specifically, ID via channel codes means that instead of encoding the message into a short codeword, we use a long codeword to transmit the entire message, i.e., we use the "Shannon approach" to ID. In contrast, noiseless ID codes "compress" the message into a short noiseless-ID codeword that a channel code further encodes into a noisy-ID codeword. Furthermore, the explanation of block codes in Section III serves us later in Sections V and VII to discuss the additional roles of block codes in creating noisy-ID codes, cf. Fig. 3.

In Section IV, we describe how noisy-ID codes address the identification problem more efficiently than block (channel) codes (for message transmission) by using an encoding specific to the ID goal. In contrast to block codes, a noisy-ID code has two functions: (i) it enables reliable communication via a noisy channel, and (ii) it provides ID-specific encoding and verification. Section IV provides information on the properties that a noisy-ID code should offer but does not explain how to explicitly construct such a noisy-ID code. Thereby, Section IV formulates the problem of constructing a suitable noisy-ID code.

Next, Section V describes the first step for one possible method to construct a noisy-ID code: the separation principle. It is possible to create a noisy-ID code by concatenating two separate codes: a linear block (channel) code (for message transmission) and a noiseless-ID code (for ID-specific encoding and verification). By the separation principle, the concatenation of a noiseless-ID code and a linear block (channel) code forms an overall noisy-ID code for noisy channels.

After Section III already addressed the first code family (i.e., linear block codes) that constitutes a separation-principle noisy-ID code, Section VI explains the second code family, i.e., the noiseless-ID codes. Readers who are not interested in noisy-ID codes but only in the noiseless-ID codes can skip Secs. III. to V. and start from Section VI. Similarly, readers who prefer to learn about noiseless-ID codes before noisy ID-codes can start from Section VI, learn about how noisy-ID codes can be constructed by concatenating a block code with a noiseless-ID code in Section V, and only then read Section IV about noisy-ID codes.

Next, Section VII describes an explicit construction of noiseless-ID codes following the tagging code construction that repurposes linear block codes to become part of reliable noiseless-ID codes. Tagging codes can also be represented as constant-weight codes (CWCs) as Section VIII explains. Section IX continues the explanation of noisy-ID codes by detailing properties of noisy-ID codes that are constructed using the concatenation of a linear block (channel) code with a noiseless-ID code. Finally, Section X provides a summary and an outlook.

TABLE III
SUMMARY OF MAIN NOTATIONS.

Messages	
$\mathcal{U}$	Set of messages
$\boldsymbol{a}$	$\in \mathcal{U}$, Message selected by the sender
$\boldsymbol{b}$	$\in \mathcal{U}$, Message selected by the receiver
q	Symbol size and field size of code
k	Size of the message (in q-ary symbols)
k'	Size of the message (in bit)
N	$= \lvert\mathcal{U}\rvert = q^k$, Number of messages
Equality and Equality Estimate	
v	$\in \{v_\mathrm{p}, v_\mathrm{n}\}$, Boolean equality whether messages match
$\hat{v}$	$\in \{\hat{v}_\mathrm{p}, \hat{v}_\mathrm{n}\}$, Binary estimate of the receiver
$\hat{v}_\mathrm{p}$	Positive equality estimate
$\hat{v}_\mathrm{n}$	Negative equality estimate (inequality)
Codes for ID via Noisy Channels	
$\mathcal{X}$	Set of all possible codewords at the sender
$\mathcal{Y}$	Set of all possible codewords at the receiver
$\mathcal{B}$	Codebook of codewords that the encoder maps to
$\boldsymbol{x}$	$\in \mathcal{X}$, Codeword encoding the message $\boldsymbol{a}$ of the sender
$\boldsymbol{y}$	$\in \mathcal{Y}$, Distorted codeword received by the receiver
$\boldsymbol{W}$	Discrete memoryless channel for codewords of size n
n	Size of codeword (in q-ary symbols)
n'	Size of codeword (in bit)
R_id	ID rate of the code
C_id	ID capacity of the (discrete memoryless) channel $\boldsymbol{W}$

II. Message Identification

A. The Identification Problem

Identification is the joint objective of two parties to determine equality between their respective messages. For the two parties, consider a sender (e.g., Alice) and a receiver (e.g., Bob), cf. Fig. 1. The sender and the receiver agree on a common set $\mathcal{U}$ of messages that both can select their respective message from. The sender selects a message $\boldsymbol{a} \in \mathcal{U}$, and the receiver selects a message $\boldsymbol{b} \in \mathcal{U}$.

The mutual goal of the sender and the receiver is for the receiver to correctly answer the question "Are the two messages $\boldsymbol{a}$ and $\boldsymbol{b}$ identical?". In other words, the receiver tries to verify the equality v between the two messages $\boldsymbol{a}$ and $\boldsymbol{b}$, whereby the equality v is the Boolean result of the comparison $\boldsymbol{a} = \boldsymbol{b}$. The equality v can either be true v_p or false v_n. Since the receiver needs information about the message $\boldsymbol{a}$ of the sender, the sender has to communicate with the receiver via the distorting channel $\boldsymbol{W}$ that connects the two parties. For this tutorial, we consider the channel $\boldsymbol{W}$ between

the encoder and the verifier to be a discrete memoryless channel (DMC) as Section II-D explains in detail. For this tutorial, "distortion" always refers to unwanted changes in the codeword symbols due to their transmission via a DMC.

To overcome the distortion of the DMC $\boldsymbol{W}$, the sender encodes the message $\boldsymbol{a}$ into a codeword $\boldsymbol{x}$ and then transmits the selected codeword $\boldsymbol{x}$ via the DMC $\boldsymbol{W}$ to the receiver. The channel $\boldsymbol{W}$ distorts the codeword $\boldsymbol{x}$ into the distorted codeword $\boldsymbol{y}$. Based on the received distorted codeword $\boldsymbol{y}$ and knowledge of its own message $\boldsymbol{b}$, the receiver determines an estimate $\hat{v}$ of the equality v between the two messages $\boldsymbol{a}$ and $\boldsymbol{b}$, i.e., the receiver verifies whether the two messages match or not.

The estimate $\hat{v}$ of the equality v between the two messages is binary and can either be positive verification $\hat{v}_\mathrm{p}$ (estimate of equality), if the receiver estimates that the two messages are identical (match); or negative verification $\hat{v}_\mathrm{n}$ (estimate of inequality), if the receiver estimates that the two messages differ (mismatch). By determining the estimate $\hat{v}$, the receiver accomplishes the communication goal that is identification. Sometimes, the literature refers to a positive verification $\hat{v}_\mathrm{p}$ as an *accept*, and to a negative verification $\hat{v}_\mathrm{n}$ as a *reject*.

For this section, we treat the encoder and the verifier as two black boxes, whereby the encoder maps the message $\boldsymbol{a}$ to a corresponding codeword $\boldsymbol{x}$ and the verifier maps the message $\boldsymbol{b}$ and the received codeword $\boldsymbol{y}$ to an estimate $\hat{v}$. We leave the question of how the encoder and the verifier achieve this to later sections, cf. Fig. 3. The ID process is composed of the encoder, the DMC $\boldsymbol{W}$, and the verifier. The messages $\boldsymbol{a}$ and $\boldsymbol{b}$ are the two inputs into the ID process, and the estimate $\hat{v}$ is the output of the ID process. After an explanation of our choice of terminology for this tutorial in Section II-B, we describe the inputs to the ID process (the messages $\boldsymbol{a}$ and $\boldsymbol{b}$) in Section II-C, the properties of the codewords $\boldsymbol{x}$ and $\boldsymbol{y}$ as well as the properties of the DMC $\boldsymbol{W}$ in Section II-D, and the output of the ID process (the equality estimate $\hat{v}$) in Section II-E. Finally, Section II-F explains communication problems that closely resemble the ID problem but are distinctly different and therefore out of the scope of this tutorial.

B. Terminology

For this tutorial, we aim to avoid introducing new terms for familiar concepts and entities. For reference, we visualize identification using the terminology presented in this subsection in Fig. 1. In the literature, "messages" $\boldsymbol{a}$ and $\boldsymbol{b}$ are often referred to as "identities (IDs)" $\boldsymbol{a}$ and $\boldsymbol{b}$. This terminology aims to highlight that the receiver does not recover a message estimate $\hat{\boldsymbol{a}}$ of the message $\boldsymbol{a}$, but an equality estimate $\hat{v}$ whether the receiver's message $\boldsymbol{b}$ is equal (identical) to the sender's message $\boldsymbol{a}$. However, a message $\boldsymbol{a}$ in the identification problem has exactly the same properties as a message $\boldsymbol{a}$ in the transmission problem. Therefore we consider it more helpful in a tutorial to stay with familiar terminology for a familiar concept, such as a message $\boldsymbol{a}$.

The literature sometimes uses the term "identity" for the true result of the communication goal of message identification, i.e., the correct answer to the question "Is message $\boldsymbol{a}$ equal to message $\boldsymbol{b}$?" In those terms, the receiver Bob determines the "identity of two messages". While the lexicographical relation to "identification" argues for the term "identity", the term "equality" is unambiguously understood as the Boolean result of the operation $\boldsymbol{a} = \boldsymbol{b}$. Hence, we use the term "equality" for this tutorial and have "equality" between the two messages when the messages are truly equal ($\boldsymbol{a} = \boldsymbol{b}$) and "no equality" (inequality) when the messages truly differ ($\boldsymbol{a} \neq \boldsymbol{b}$). The equality v is a Boolean that can either be true or false.

Furthermore, we refer to the output of the identification process at the receiver as the "estimate" $\hat{v}$. The receiver determines the equality estimate $\hat{v}$ that estimates the equality v between message $\boldsymbol{a}$ and message $\boldsymbol{b}$. This terminology mirrors the name of the output of a receiver in the transmission problem where the receiver estimates the message $\boldsymbol{a}$ via a message estimate $\hat{\boldsymbol{a}}$. In identification, the receiver does not estimate the message $\boldsymbol{a}$ of the sender, but the equality v between the messages $\boldsymbol{a}$ and $\boldsymbol{b}$. The literature also refers to the estimate $\hat{v}$ as the "verdict". Since both communication problems (identification and transmission) include an estimate with a probability of error relative to the underlying truth (the message $\boldsymbol{a}$ in transmission, and the equality v in identification), we refrain from using a different term for the familiar concept of an estimate.

We refer to the party that conveys information about its message via the channel as the "sender" (Alice), cf. Fig. 1. Also, we refer to the party that receives information to determine an estimate of the equality as the "receiver" (Bob). We avoid the term "transmitter" (in place of "sender") for its lexicological relation to the "transmission" problem. The sender includes a "source" for the message $\boldsymbol{a}$ and an "encoder" that encodes the message into a codeword. The receiver includes a "source" for the message $\boldsymbol{b}$, a "verifier" that determines the estimate $\hat{v}$ of the equality v, and a "sink" for the estimate $\hat{v}$. We avoid the term "decoder" (in place of "verifier") because the receiver does not "decode" the codeword $\boldsymbol{y}$ into a message estimate $\hat{\boldsymbol{a}}$. Rather, the receiver determines in a verification step an estimate $\hat{v}$ of the equality v between the two messages of the sender and the receiver. Referring to the "verifier" as "decoder" could imply a similarity to decoding in the transmission problem. However, the difference between decoding a codeword into a message estimate $\hat{\boldsymbol{a}}$ and verifying a codeword into an equality estimate $\hat{v}$ is the fundamental difference between the transmission problem and the identification problem.

We use the abbreviation "ID" for the term "identification". Specifically, we refer to identification codes as ID codes, and to the identification problem as ID.

C. Messages

In order to perform identification, the sender first selects the message $\boldsymbol{a}$ from the set $\mathcal{U}$ of possible messages. If the sender selects every message in the set $\mathcal{U}$ with equal likelihood, then the sender needs at least $k' = \log_2(|\mathcal{U}|)$ bits to represent the message in binary form. Using fewer bits makes it impossible to exactly (i.e., without errors) recover the message from the binary representation. This result is known

as Shannon's source coding theorem [2], and the lower bound on the number of bits needed for representation is called the entropy $H = \log_2(|\mathcal{U}|)$ of the source. If the messages are not uniformly likely, then the entropy of the source is lower, and thereby compression of the message is possible. While there is also a notion of an ID entropy [96], we limit our description to the entropy known from classical information theory.

This tutorial focuses on coding for the identification problem, i.e., on noisy-ID codes. The literature does not make any explicit assumptions about the probability distribution of the source. Therefore, noisy-ID coding theory implicitly considers the worst-case in terms of "compression potential", i.e., uniformly likely messages. Hence, with the source coding theorem, the messages cannot be further compressed without introducing errors in the recovery of the messages. The literature also considers ID-specific source coding [96], [97]. ID source coding is out of the scope of this tutorial.

Any message (e.g., the message $\boldsymbol{a}$ of the sender, the message $\boldsymbol{b}$ of the receiver, or an arbitrary message $\boldsymbol{u}$) can be represented as an array of k symbols, whereby each symbol is selected from an alphabet U. In this representation, the set of messages is $\mathcal{U} = U^k$. Throughout this tutorial, we typeset vectors, such as the message $\boldsymbol{a} \in \mathcal{U} = U^k$, in bold font. For the alphabet U, we consider integers in base q, i.e., $U = \{0, 1, \ldots, q-1\}$ with $|U| = q$. Overall, the set of messages is

$$\mathcal{U} = U^k = \{0, 1, \ldots, q-1\}^k. \tag{1}$$

For base $q = 2$, the result is a binary representation using k bits, i.e., an array of k 0s and 1s, and for base $q = 256$, the message is represented by k symbols between 0 and 255. The base q completely defines the alphabet U. Therefore, instead of repeatedly mentioning the alphabet U, we limit our description to the base q of the messages from hereon. The total number of messages that can be represented is

$$N = |\mathcal{U}| = q^k = 2^{k'}. \tag{2}$$

Reciprocally, messages are represented using k symbols, with

$$k = \log_q(N) = \log_q(|\mathcal{U}|). \tag{3}$$

Next to the q-ary message size k, we sometimes use the binary message size

$$k' = \log_2(N) = \log_2(|\mathcal{U}|) = k \log_2(q). \tag{4}$$

The literature often considers the binary case with $q = 2$, and many theorems are therefore stated for the binary case. One benefit of the binary representation lies in the independence of its expressiveness from the choice of base q. For this tutorial, we focus on the q-ary representation because it is practically relevant and generalizes the binary representation. The binary message size k' is additionally useful to measure the size in bits of messages supported by the respective code. For instance, we consider $k' \approx 168$ bit to be more expressive than the corresponding $N = q^k = 127^{24} \approx 3 \cdot 10^{50}$.

D. Codewords and Discrete Memoryless Channel

After selecting the message $\boldsymbol{a}$ from the set $\mathcal{U}$ of possible messages, cf. Section II-C, the encoder encodes the message $\boldsymbol{a}$ into a codeword $\boldsymbol{x}$. The codeword $\boldsymbol{x}$ can be considered a vector of n q-ary symbols, whereby the encoder selects each symbol from the encoder alphabet X. The encoder selects the codeword $\boldsymbol{x} = [x_1, \ldots, x_n]$ from the set $\mathcal{X} = X^n$ of codewords of the encoder. Additionally, we sometimes use the binary codeword size $n' = n \log_2(q)$ analogous to the binary message size k' as defined in Eq. (4).

Next, the DMC $\boldsymbol{W}$ transforms the transmitted codeword $\boldsymbol{x}$ probabilistically into the received codeword $\boldsymbol{y} = [y_1, \ldots, y_n]$. Since the channel is memoryless, each codeword symbol x is transformed independently of other symbols, i.e., the DMC $W(y|x)$ can be described by a $|X| \times |Y|$ matrix of transition probabilities from all possible input symbols $x \in X$ to all possible output symbols $y \in Y$. The sender uses the channel W for each of the n codeword symbols independently, such that the overall channel is

$$\boldsymbol{W}(\boldsymbol{y}|\boldsymbol{x}) = [W(y_1|x_1), \ldots, W(y_n|x_n)]. \tag{5}$$

The verifier uses the received codeword $\boldsymbol{y}$ and the message $\boldsymbol{b}$ of the receiver to determine an estimate $\hat{v}$ of the equality v between the messages $\boldsymbol{a}$ and $\boldsymbol{b}$.

To simplify notation, examples, and visualization, we consider the source alphabet U, the encoder alphabet X, and the decoder alphabet Y to be identical, i.e., $U = X = Y = \{0, 1, \ldots, q-1\}$. In that case, the set $\mathcal{X}$ of codewords of the encoder is given by

$$\mathcal{X} = X^n = \{0, 1, \ldots, q-1\}^n, \tag{6}$$

whereby the set $\mathcal{X}$ holds $|\mathcal{X}| = q^n$ codewords. Also, the sets of codewords of the sender and the receiver are identical for these conditions, i.e., $\mathcal{Y} = \mathcal{X}$. Throughout this tutorial, we write $\mathcal{Y} = \mathcal{X}$ for the set of codewords at the receiver, and simplify this to "set $\mathcal{X}$ of codewords at the receiver" in figures, for clarity.

Furthermore, for the DMC $\boldsymbol{W}$, for simplicity, we only consider DMCs that are "zero mean". In other words, the received codewords $\boldsymbol{y}$ spread around the sent codeword $\boldsymbol{x}$. This restriction is common and simplifies explanation and visualization significantly.

E. Equality Estimate

The verifier can be understood to determine its estimate $\hat{v}$ via binary classification whether the received codeword $\boldsymbol{y}$ indicates two matching messages (positive verification $\hat{v}_\mathrm{p}$) or two mismatched messages (negative verification $\hat{v}_\mathrm{n}$). The estimate $\hat{v}$ is the verifier's imperfect estimation of the equality v, as the hat operator indicates in the symbol for the estimate $\hat{v}$. Hence, the estimate $\hat{v}$ of the verifier does not always match the equality v. The equality v can either be true v_p, if the two messages are truly identical ($\boldsymbol{a} = \boldsymbol{b}$), or false v_n, if the two messages truly differ ($\boldsymbol{a} \neq \boldsymbol{b}$).

The combination of two possible values for the estimate $\hat{v}$ (either positive or negative verification) and two possible values for the equality v (either true or false equality) yields

TABLE IV
POSSIBLE OUTCOMES IN IDENTIFICATION.

Estimate $\hat{v}$ \ Equality v	True Equality v_p ($\boldsymbol{a} = \boldsymbol{b}$)	True Inequality v_n ($\boldsymbol{a} \neq \boldsymbol{b}$)
Positive Verification $\hat{v}_\mathrm{p}$ Estimates Equality	True positive $\hat{v}_\mathrm{tp}$ No error	False positive $\hat{v}_\mathrm{fp}$ FP error
Negative Verification $\hat{v}_\mathrm{n}$ Estimates Inequality	False negative $\hat{v}_\mathrm{fn}$ FN error	True negative $\hat{v}_\mathrm{tn}$ No error

four possible outcomes of the verification step, see Table IV: true-positive estimates TP, false-positive estimates FP, true-negative estimates TN, and false-negative estimates FN. The literature refers to the estimate FN also as an error of the first kind (type I error) and to the estimate FP also as an error of the second kind (type II error).

Alternatively, determining the estimate $\hat{v}$ is also framed as a binary hypothesis test in the literature. In the hypothesis test framing, the receiver hypothesizes equality between the two messages, and that hypothesis is either confirmed or rejected based on the received codeword. In the fundamental definition of ID, the receiver confirms or rejects the hypothesis based on a single received codeword. This is in contrast to the statistical notion of hypothesis testing, where generally an entire data set of evidence is collected and used to confirm or reject a hypothesis. Hence, framing identification as a binary hypothesis test can be misleading since ID is different from verifying a hypothesis about the statistical properties of a population.

F. Related Communication Problems

The messages $\boldsymbol{a}$ and $\boldsymbol{b}$ of the sender and of the receiver can be selected arbitrarily from the common set $\mathcal{U}$ but are fixed. Specifically, identification is defined for a fixed pair $(\boldsymbol{a}, \boldsymbol{b})$ of messages. When this scenario is generalized, the receiver may use the received codeword $\boldsymbol{y}$ to estimate the equality between the sender's message $\boldsymbol{a}$ and several receiver messages $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\beta$; or, multiple receivers that obtain the codeword $\boldsymbol{y}$ may estimate the equality for their single (or multiple) messages. Hence, the receiver(s) either determine in β separate equality estimates whether the received codeword $\boldsymbol{y}$ implies equality for each receiver message individually. Or, a single receiver determines in a single classification into β classes which message $\hat{\boldsymbol{a}}$ from the set of β messages is most likely the one that the sender selected. The literature typically writes K instead of β for the number of messages that the receiver is interested in and refers to the problem as K-ID [98]. For $\beta = |\mathcal{U}|$ messages, the classification into β classes is exactly the message transmission problem. Thus, extending the number of message pairs under investigation naturally leads to communication problems related to the ID problem. For this tutorial, we focus on message identification for a single ($\beta = 1$) fixed pair of messages.

The sender does not know which message $\boldsymbol{b}$ the receiver selected. Otherwise, the sender could determine locally whether the messages match, i.e., determine the equality v. However, in the ID problem, the equality v is unknown to both parties. If the sender knows the equality v, then the sender can encode the Boolean equality v into a single bit (instead of encoding the message $\boldsymbol{a}$ that is typically larger), and convey the encoded equality (instead of the encoded message) to the receiver. On the other hand, if the receiver knows the equality v, then there is no need for communication and no need to estimate the equality. Furthermore, only the receiver but not the sender determines an estimate $\hat{v}$, i.e., the sender does not gain any information.

Finally, we emphasize that identification is not about assigning similarity scores to different messages. Rather, in identification, the receiver decides whether two messages are exactly the same, or not. If a message differs by a single bit from another one, then the messages are not exactly identical.

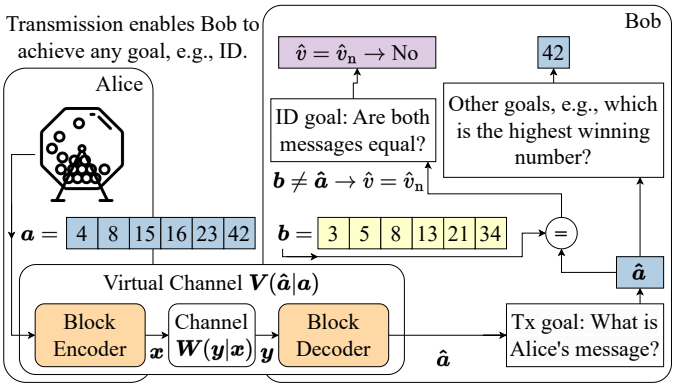


Fig. 4. Linear block codes enable reliable message transmission. Bob obtains an estimate $\hat{\boldsymbol{a}}$ of Alice's message $\boldsymbol{a}$ that Bob can use to accomplish any communication goal, including message identification.

G. Summary

In the ID problem, the receiver estimates whether the message $\boldsymbol{b}$ of the receiver matches the message $\boldsymbol{a}$ of the sender based on a codeword $\boldsymbol{x}$ that the sender transmits to the receiver. Specifically, the receiver *identifies* whether the sender has the same message $\boldsymbol{a} = \boldsymbol{b}$ as the receiver in a binary equality estimate $\hat{v}$, i.e., the receiver determines either a match or a mismatch. A Discrete Memoryless Channel $\boldsymbol{W}$ connects the sender and the receiver and can distort the codeword $\boldsymbol{x}$ such that several symbols of the received codeword $\boldsymbol{y}$ can differ from the codeword $\boldsymbol{x}$ of the sender. The distortion can cause the receiver to determine an incorrect estimate $\hat{v}$ about whether the messages match or not, cf. Table IV. To address the distortion, the sender encodes the message $\boldsymbol{a}$ that consists of k symbols as a codeword $\boldsymbol{x}$ of n symbols. Ideally, the encoding enables reliable ID despite the distortion of the channel.

III. LINEAR BLOCK CODES

A. Overview

When Bob wants to determine whether he won in Alice's lottery (i.e., when Bob wants to address an ID problem),

TABLE V
SUMMARY OF NOTATIONS FOR LINEAR BLOCK CODES.

$\boldsymbol{V}$	Virtual discrete error-free channel
$\hat{\boldsymbol{a}}$	Message estimate by the receiver
$\mathcal{D}_{\boldsymbol{u}}$	$\subset \mathcal{Y}$, Decoder subset of codewords of message $\boldsymbol{u}$
R_{fec}	Code rate of a linear block (FEC) code
C_{DMC}	Transmission capacity of a discrete memoryless channel
$D(\boldsymbol{x_1}, \boldsymbol{x_2})$	Hamming distance between $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$
δ	Minimum Hamming distance between all codeword pairs

Alice can transmit the winning lottery numbers $\boldsymbol{a}$ to Bob. Bob obtains an estimate $\hat{\boldsymbol{a}}$ of the winning numbers $\boldsymbol{a}$ and compares the estimate $\hat{\boldsymbol{a}}$ to his own numbers $\boldsymbol{b}$. Since Bob gets a copy (via the estimate $\hat{\boldsymbol{a}}$) of Alice's message $\boldsymbol{a}$, he can accomplish his goal of identification. In other words, it is possible to address the ID problem using message transmission.

A linear block code enables the reliable transmission of the message $\boldsymbol{a}$ from the sender (Alice) to the receiver (Bob) via a noisy channel $\boldsymbol{W}$. Thereby, a block code is a channel code for message transmission. Channel codes are also known as forward error correction (FEC) codes. The linear block encoder at the sender maps the sender's message $\boldsymbol{a}$ to a block codeword $\boldsymbol{x}$ as Fig. 4 visualizes. The DMC $\boldsymbol{W}(\boldsymbol{y}|\boldsymbol{x})$ distorts the linear block codeword $\boldsymbol{x}$ such that the receiver obtains a distorted block codeword $\boldsymbol{y}$. Based on the received distorted block codeword $\boldsymbol{y}$, the block decoder at the receiver determines an estimate $\hat{\boldsymbol{a}}$ of the message $\boldsymbol{a}$ of the sender.

Any communication goal can be accomplished by transmitting the message $\boldsymbol{a}$ of the sender from the sender to the receiver. This discovery is one of the major contributions of Claude Shannon's Mathematical Theory of Communication [2] that directly led to the development of modern communication systems. Thereby, also the ID problem can be accomplished using message transmission.

In this section, we describe how linear block codes enable reliable message transmission over a distorting (noisy) channel. We limit our explanation to linear block codes (such as Reed-Solomon codes, Reed-Muller codes, and Polar codes) and leave out other channel codes [such as convolutional codes (e.g., LDPC codes) and rateless (fountain) codes [99]]. Furthermore, the field of efficient decoding of linear block codewords is mostly out of the scope of this tutorial and we limit our explanation to basic principles, omitting more advanced methods, such as guessing random additive noise decoding (GRAND) [100].

A linear block code can be used to implement the encoder-verifier pair that addresses the ID problem as Section II-A explained and Fig. 1 visualized. Specifically, the block encoder acts as the encoder that maps the sender's message $\boldsymbol{a}$ to the sender's codeword $\boldsymbol{x}$. The verifier (that determines the equality estimate $\hat{v}$ based on the received codeword $\boldsymbol{y}$) consists of the block decoder and the comparison operation between the estimated message $\hat{\boldsymbol{a}}$ of the sender and the known message $\boldsymbol{b}$ of the receiver.

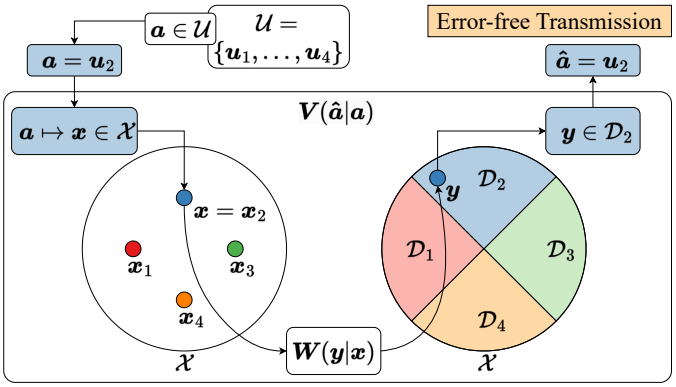


Fig. 5. Schematic view of block encoding and decoding. The block encoder encodes the message $\boldsymbol{a}$ deterministically to a codeword $\boldsymbol{x}$, cf. Eq. (7). For a suitable block code, the receiver can recover the message $\boldsymbol{a}$ from the received codeword $\boldsymbol{y}$ such that the block code can be understood to form from the noisy DMC $\boldsymbol{W}$ a virtual channel $\boldsymbol{V}(\hat{\boldsymbol{a}}|\boldsymbol{a})$ that enables reliable message transmission.

B. Linear Block Encoding

A block code can be considered to form a virtual, approximately noiseless channel $\boldsymbol{V}(\hat{\boldsymbol{a}}|\boldsymbol{a})$ from the noisy channel $\boldsymbol{W}(\boldsymbol{y}|\boldsymbol{x})$. Ideally, the block code forms a perfectly noiseless channel $\boldsymbol{V}(\hat{\boldsymbol{a}}|\boldsymbol{a})$ that always correctly reproduces any input message $\boldsymbol{a}$ as the correct message estimate $\hat{\boldsymbol{a}} = \boldsymbol{a}$. This noiseless channel is also known as identity channel. The virtual channel $\boldsymbol{V}(\hat{\boldsymbol{a}}|\boldsymbol{a})$ can be considered an abstraction that hides the details of the interaction between the block code and the noisy channel $\boldsymbol{W}$ to the outside. Thereby, the sender and the receiver leave the formation of the virtual channel $\boldsymbol{V}$ (that enables reliable message transmission) to the block code, and interface with the formed virtual channel $\boldsymbol{V}$ instead of interfacing with the noisy channel $\boldsymbol{W}$, cf. Fig. 4. This abstraction is fundamental to the development of modern communication systems as it enables the separation of more complex communication problems into sub-problems that can be addressed separately. For example, in the OSI model, the physical layer provides a virtual channel for the upper layers to communicate over. Thereby, the problem of reliable message transmission is that of forming a virtual, ideally noiseless channel $\boldsymbol{V}(\hat{\boldsymbol{a}}|\boldsymbol{a})$.

Next, we explain how the block code forms the virtual, approximately noiseless channel $\boldsymbol{V}(\hat{\boldsymbol{a}}|\boldsymbol{a})$ and begin with the block encoder. We visualize the elements of the noiseless channel $\boldsymbol{V}(\hat{\boldsymbol{a}}|\boldsymbol{a})$ in Fig. 5 and the integration of the noiseless channel into the ID process in Fig. 4.

First, the sender selects a message $\boldsymbol{a}$ from a set $\mathcal{U}$ of messages. In the example in Fig. 5, there are four messages in the set $\mathcal{U}$: $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3$, and $\boldsymbol{u}_4$. The block encoder maps the selected message $\boldsymbol{a} \in \mathcal{U}$ to its corresponding block codeword $\boldsymbol{x} = \boldsymbol{x}_{\boldsymbol{a}}$, according to the block (forward error correction) encoding function

$$f_{\text{fec}} : \boldsymbol{a} \mapsto \boldsymbol{x} = \boldsymbol{x}_{\boldsymbol{a}} \in \mathcal{X} \quad \forall \boldsymbol{a} \in \mathcal{U}. \tag{7}$$

Thereby, the block encoding function f_{fec} maps from the *domain* $\mathcal{U}$ (the set of messages) to the *codomain* $\mathcal{X}$ (the set of all block codewords). In the example, the block encoder

maps the message $\boldsymbol{a} = \boldsymbol{u}_2$ to its corresponding block codeword $\boldsymbol{x} = \boldsymbol{x_2}$. Throughout this tutorial we abbreviate the subscript for codewords (and sets) associated with a message, i.e., we write $\boldsymbol{x_2}$ for the codeword associated with message $\boldsymbol{u}_2$ instead of $\boldsymbol{x}_{\boldsymbol{u}_2}$. The encoding function f_{fec} is deterministic, since f_{fec} maps each message $\boldsymbol{a}$ to the one codeword $\boldsymbol{x_a}$ that is associated with the message $\boldsymbol{a}$. In other words, the block codeword $\boldsymbol{x_a}$ is the image of the message $\boldsymbol{a}$.

The four codewords $\boldsymbol{x_1}, \ldots, \boldsymbol{x_4}$ of the four possible messages $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_4$ are elements of the set $\mathcal{X}$ of possible codewords as defined in Section II-D. In the block encoder, not every possible codeword $\boldsymbol{x}$ in the set $\mathcal{X}$ is associated with a message $\boldsymbol{a}$, cf. the white areas surrounding the four codewords $\boldsymbol{x_1}, \ldots, \boldsymbol{x_4}$ in Fig. 5. Rather, most codewords $\boldsymbol{x}$ in the set $\mathcal{X}$ are not valid encodings of any message. We refer to codewords that are valid encodings of a message (possible outputs of the function $f_{\text{fec}}(\boldsymbol{a})$) as *message encoding codewords* $\boldsymbol{x_a}$. The set of all message encoding codewords is the codebook $\mathcal{B}$, whereby the codebook $\mathcal{B}$ is the *image* of the block encoding function f_{fec} and a subset of the codomain $\mathcal{X}$. In the example in Fig. 5, the codebook $\mathcal{B}$ is $(\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \boldsymbol{x_4})$, i.e., the set of images that the block encoding function f_{fec} can map to. Since every message encoding codeword is associated with exactly one message, the *preimage* of any message encoding codeword consists of exactly one element, i.e., the associated message.

Both the message $\boldsymbol{a} \in \mathcal{U}$ and the block codeword $\boldsymbol{x}$ can be represented as an array of symbols in base q, see Section II-C and Section II-D. The block code can represent message sizes up to k symbols using codeword sizes of n symbols. Thereby, the block code supports up to $N = q^k$ messages, cf. Section II-C. To enable recovery of the original message $\boldsymbol{a}$ despite the distortion of the channel $\boldsymbol{W}$, the block codeword $\boldsymbol{x}$ typically includes redundant information in the form of extra symbols in addition to the k symbols of the message $\boldsymbol{a}$. Since redundant information adds an overhead to the codeword, the size $n = |\boldsymbol{x}|$ of the transmitted block codeword $\boldsymbol{x}$ typically exceeds the size k of the selected message $\boldsymbol{a}$, i.e., $n \geq k$. The noiseless channel $\boldsymbol{V}(\hat{\boldsymbol{a}}|\boldsymbol{a})$ that the block code virtually creates conveys $k = |\boldsymbol{V}|$ symbols and the noisy channel $\boldsymbol{W}(\boldsymbol{y}|\boldsymbol{x})$ conveys $n = |\boldsymbol{W}|$ symbols. The size k of the selected message $\boldsymbol{a}$ and the size n of the transmitted codeword $\boldsymbol{x}$ determine the rate R_{fec} of the block (forward error correction) code. Specifically,

$$R_{\text{fec}} = \frac{\log_2(|\mathcal{U}|)}{\log_2(|\mathcal{X}|)} = \frac{\log_2(q^k)}{\log_2(q^n)} = \frac{k}{n} \leq C \leq 1. \tag{8}$$

A high rate R_{fec} is desirable as it corresponds to an efficient use of the channel $\boldsymbol{W}$, i.e., the block encoder adds only a few additional symbols to allow the block decoder to recover the message $\boldsymbol{a}$ despite the distortion of the channel $\boldsymbol{W}$. However, the higher the rate R_{fec}, the more susceptible the transmission of the message becomes to distortions of the channel $\boldsymbol{W}$, because the block code adds less redundancy to the codeword $\boldsymbol{x}$. In typical communication systems, the block code operates at a rate that allows the correction of almost all errors introduced by the channel [8], [11], [101].

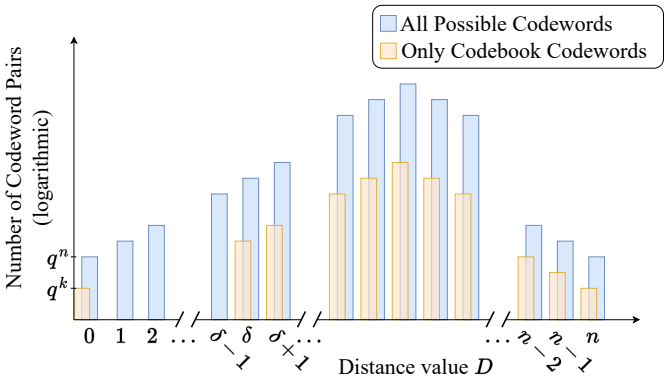


Fig. 6. Schematic example of a histogram of the distances between codewords of a linear block code. The codebook codewords have a Hamming distance D of at least δ. The Hamming distance of any codeword (including codebook codewords) to itself is $D = 0$.

The number of messages that the block code can represent can be reformulated using the block code rate to

$$N = q^k = 2^{k'} = 2^{n' R_{\text{fec}}} = |\mathcal{X}|^{R_{\text{fec}}} \leq |\mathcal{X}|, \tag{9}$$

i.e., the block code scales exponentially in the codeword size n (the block length).

C. Distance

The Hamming distance D measures the number of positions that differ for a pair of arrays, such as messages or codewords. For example, a pair $(\boldsymbol{x_1}, \boldsymbol{x_2})$ of codewords (each codeword of length n) differs in $D(\boldsymbol{x_1}, \boldsymbol{x_2})$ positions, and the codewords share the same symbol in $n - D(\boldsymbol{x_1}, \boldsymbol{x_2})$ positions.

The distance D can take on values between 0 and $n - 1$, i.e., the range of distance values depends on the size n of the codeword. The distance values of the linear block codebook $\mathcal{B}$ (that is the set of message encoding codewords) characterize properties of the underlying linear block code. There are $N = q^k$ different messages that the block code maps to N unique block codewords. Hence, for a linear block code, there are $q^k(q^k - 1)/2$ pairs of differing message encoding codewords, and q^k pairs of identical message encoding codewords. The pairs of identical message encoding codewords have a Hamming distance of $D = 0$, and all other pairs have a non-zero Hamming distance. Because there are typically significantly more pairs of message encoding codewords than the number k of distance values, multiple pairs of message encoding codewords share the same distance value D.

It is possible to visualize the number of pairs of message encoding codewords that share the same distance value D as a histogram, cf. Fig. 6. The linear block encoder does not map the messages in $\mathcal{U}$ to all possible codewords in the set $\mathcal{X}$. Rather, the linear block encoder maps the messages in $\mathcal{U}$ only to codewords in the codebook $\mathcal{B} \subset \mathcal{X}$. Thereby, the block code can guarantee a minimum distance δ between all pairs of message encoding codewords. In other words, the linear block code purposefully excludes possible codewords and allows only codewords into the codebook $\mathcal{B}$ that have a high distance from other codebook codewords. The codebook $\mathcal{B}$ holds $|\mathcal{B}| = N = q^k$ codewords, one for each message, whereas

there are $|\mathcal{X}| = q^n > q^k$ possible codewords overall. In Fig. 6, the minimum distance bound δ is a tight bound. In general, the minimum distance bound δ does not have to be a tight bound, i.e., it is possible that the smallest Hamming distance between codebook codewords exceeds the bound. Similarly, it is possible that no codebook codeword pairs exhibit the maximum distance $D = n$ in contrast to the visualization in Fig. 6. For the pairs of all possible codewords, the histogram of distance values D follows a binomial distribution and the mean of the distance values D in the histogram depends on the symbol size q of the codeword [95, Eq. 21].

With the minimum distance bound δ, an $(n, k, \delta)_q$ linear block code maps length k messages in base q to length n codewords in base q while guaranteeing a minimum distance δ between all pairs of message encoding codewords. A linear block code can correct up to $\delta/2$ errors, i.e., the channel can distort up to $\delta/2$ symbols in the length n codeword $\boldsymbol{x}$ and the block decoder can still recover the encoded message. In other words, as long as the distance $D(\boldsymbol{x}, \boldsymbol{y})$ between the transmitted codeword $\boldsymbol{x}$ and the received codeword $\boldsymbol{y}$ is smaller or equal than the minimum distance δ, the decoder can recover the message $\boldsymbol{a}$. In general, a high minimum distance bound δ corresponds to a more reliable channel code.

D. Linear Block Decoding and Verification

The DMC $\boldsymbol{W}(\boldsymbol{y}|\boldsymbol{x})$ (cf. Section II-A) distorts the transmitted block codeword $\boldsymbol{x}$, such that the receiver receives a distorted block codeword $\boldsymbol{y} \in \mathcal{Y}$. Based on the received distorted block codeword $\boldsymbol{y}$ (that is an element of the set $\mathcal{Y} = Y^n$ of codewords of the block decoder) the block decoder determines the estimated message $\hat{\boldsymbol{a}}$. The block decoder can use the block codewords that are not message encoding codewords at the encoder to mitigate the effects of the channel distortion.

To create a block code decoding rule, large distortions are considered to be less likely than small distortions. One possible block code decoding rule is maximum likelihood decoding. In maximum likelihood decoding, the block decoder determines the message encoding codeword $\hat{\boldsymbol{x}}_{\boldsymbol{a}}$ from the codebook $\mathcal{B}$ that has the smallest Hamming distance D to the received block codeword $\boldsymbol{y}$ (given that the channel has an error probability of less than 50%). In the example in Fig. 5, the received block codeword $\boldsymbol{y}$ is closest to $\boldsymbol{x_2}$ that is the block codeword associated with message $\boldsymbol{u}_2$. Therefore, the decoder decides that the received block codeword $\boldsymbol{y}$ is most likely a distorted version of $\boldsymbol{x_2}$, and correctly decodes the received block codeword into the estimated message $\hat{\boldsymbol{a}} = \boldsymbol{u}_2$ that is associated with the message-encoding codeword $\boldsymbol{x_2}$. For simplicity, we do not discuss how to handle cases in which the received block codeword $\boldsymbol{y}$ has an identical distance to two or more message encoding codewords.

With the distance metric, the block decoder can assign every received block codeword $\boldsymbol{y} \in \mathcal{Y}$ to the closest message encoding codeword. In other words, every message encoding codeword $\boldsymbol{x}_{\boldsymbol{a}}$ (i.e., $\boldsymbol{x_1}$ through $\boldsymbol{x_4}$ in the example) generates a decoding subset $\mathcal{D}_{\hat{\boldsymbol{a}}} \subset \mathcal{Y}$ of block codewords that are all associated with the same estimated message $\hat{\boldsymbol{a}}$. The block decoder can be considered to create a partition of the set $\mathcal{Y}$ of codewords into $N = |\mathcal{U}|$ subsets $\mathcal{D}_{\hat{\boldsymbol{a}}}$ (i.e., four subsets in the example in Fig. 5). In Fig. 5, the subsets $\mathcal{D}_{\hat{\boldsymbol{a}}}$ have a similar hue to their respective generating message encoding codeword $\boldsymbol{x}_{\boldsymbol{a}}$.

As every codeword $\boldsymbol{y}$ is an element of exactly one subset $\mathcal{D}_{\hat{\boldsymbol{a}}}$, the block decoding is not ambiguous, i.e., the block decoding is deterministic. Thereby, creating a block code is similar to sphere packing problems. Each sphere corresponds to a subset of block codewords that decode to a single message. The larger the number of messages that the block code supports, the more non-overlapping small spheres (i.e., disjoint subsets of block codewords) have to fit into the large sphere (i.e., the set $\mathcal{Y}$ of all possible block codewords).

In message transmission, the block decoder can also be considered to perform multi-class classification of the received block codeword $\boldsymbol{y}$ into one of $N = |\mathcal{U}|$ different classes, cf. Section II-F. The block decoder decides which message $\hat{\boldsymbol{a}}$ out of all N messages is most likely the one the sender transmitted. In other words, the block decoder has to classify into which of the N decoding subsets $\mathcal{D}_{\hat{\boldsymbol{a}}}$ the received block codeword $\boldsymbol{y}$ belongs:

$$\hat{\boldsymbol{a}} = \begin{cases} \boldsymbol{u}_1 \text{ if } \boldsymbol{y} \in \mathcal{D}_1, \\ \dots, \\ \boldsymbol{u}_N \text{ if } \boldsymbol{y} \in \mathcal{D}_N. \end{cases} \tag{10}$$

In the example in Fig. 5, the mild distortion does not distort the transmitted block codeword $\boldsymbol{x}$ too much. Therefore, despite the distortion, the received block codeword $\boldsymbol{y}$ is an element of $\mathcal{D}_2$ and therefore the block decoder maps the received block codeword $\boldsymbol{y}$ to the message estimate $\hat{\boldsymbol{a}} = \boldsymbol{u}_2$. This describes one method to implement a mapping rule of the block decoder function:

$$\boldsymbol{y} \mapsto \hat{\boldsymbol{a}} \in \mathcal{U} \quad \forall \boldsymbol{y} \in \mathcal{Y}. \tag{11}$$

For a typical modern communication system, the received message $\hat{\boldsymbol{a}} = \boldsymbol{a}$, i.e., the message is transmitted without errors.

In addition to any information that the receiver had access to before the transmission (e.g., knowledge of its own selected message $\boldsymbol{b}$), the receiver now has access to an estimate $\hat{\boldsymbol{a}}$ of the message $\boldsymbol{a}$ of the sender, and can therefore perform any operation (e.g., computation) that involves that message estimate $\hat{\boldsymbol{a}}$. Thereby, the receiver can now accomplish any communication goal, including the ID problem, i.e., verifying whether $\boldsymbol{a} = \boldsymbol{b}$. The receiver determines its equality estimate $\hat{v}$ by comparing the message estimate $\hat{\boldsymbol{a}}$ with the receiver's message $\boldsymbol{b}$:

$$\hat{v} = \begin{cases} \hat{v}_\mathrm{p} \text{ if } \hat{\boldsymbol{a}} = \boldsymbol{b}, \\ \hat{v}_\mathrm{n} \text{ if } \hat{\boldsymbol{a}} \neq \boldsymbol{b}. \end{cases} \tag{12}$$

Thereby, when addressing the ID problem with linear block codes, the block decoder and a comparison operation together constitute the "verifier" that Fig. 1 visualizes.

In conclusion, the block code is able to address the ID problem via a noisy channel. The block code achieves this by providing reliable message transmission over the noisy channel $\boldsymbol{W}$. The receiver can use the message estimate $\hat{\boldsymbol{a}}$ to determine the equality estimate $\hat{v}$.

TABLE VI
POSSIBLE EFFECTS OF CHANNEL DISTORTION ON MESSAGE ESTIMATE AND EQUALITY ESTIMATE IN ID VIA TRANSMISSION CHANNEL CODES.

Channel $\boldsymbol{W} \rightarrow$		No distortion	Minor distortion	Distortion exceeds block code's correction capability	
Block codeword $\boldsymbol{y}$:		$\boldsymbol{y} = \boldsymbol{x}$	$\boldsymbol{y} \neq \boldsymbol{x}$	$\boldsymbol{y} \neq \boldsymbol{x}$	
Channel effect on message est. $\hat{\boldsymbol{a}}$:		No distortion	$\hat{\boldsymbol{a}}$-neutral distortion	$\hat{\boldsymbol{a}}$-changing distortion	
Channel effect on equality est. $\hat{v}$:		No distortion	$\hat{v}$-neutral distortion	$\hat{v}$-neutral distortion	$\hat{v}$-changing distortion
Equality $v \downarrow$		$\downarrow$		$\downarrow$	$\downarrow$
True equality v_p ($\boldsymbol{a} = \boldsymbol{b}$)	Message est. $\hat{\boldsymbol{a}}$:	Correct message est. ($\hat{\boldsymbol{a}} = \boldsymbol{a}$)		-	False message est. ($\hat{\boldsymbol{a}} \neq \boldsymbol{a}$)
	Verification:	$\hat{\boldsymbol{a}} = \boldsymbol{b} \rightarrow \hat{v} = \hat{v}_\mathrm{p}$		-	$\hat{\boldsymbol{a}} \neq \boldsymbol{b} \rightarrow \hat{v} = \hat{v}_\mathrm{n}$
	Equality est. $\hat{v}$:	TP equality est. $\hat{v}_\mathrm{p} = v_\mathrm{p}$		-	FN equality est. $\hat{v}_\mathrm{n} \neq v_\mathrm{p}$
True inequality v_n ($\boldsymbol{a} \neq \boldsymbol{b}$)	Message est. $\hat{\boldsymbol{a}}$:	Correct message est. ($\hat{\boldsymbol{a}} = \boldsymbol{a}$)		False message est. ($\hat{\boldsymbol{a}} \neq \boldsymbol{a}$)	False message est. ($\hat{\boldsymbol{a}} \neq \boldsymbol{a}$)
	Verification:	$\hat{\boldsymbol{a}} \neq \boldsymbol{b} \rightarrow \hat{v} = \hat{v}_\mathrm{n}$		$\hat{\boldsymbol{a}} \neq \boldsymbol{b} \rightarrow \hat{v} = \hat{v}_\mathrm{n}$	$\hat{\boldsymbol{a}} = \boldsymbol{b} \rightarrow \hat{v} = \hat{v}_\mathrm{p}$
	Equality est. $\hat{v}$:	TN equality est. $\hat{v}_\mathrm{n} = v_\mathrm{n}$		TN equality est. $\hat{v}_\mathrm{n} = v_\mathrm{n}$	FP equality est. $\hat{v}_\mathrm{p} \neq v_\mathrm{n}$

E. Transmission Errors

The block code is not always able to correctly recover the message $\boldsymbol{a}$ when the channel distortion exceeds the block code's capability of forward error correction. In those cases, the message estimate $\hat{\boldsymbol{a}}$ differs from the message $\boldsymbol{a}$ of the sender, i.e., $\hat{\boldsymbol{a}} \neq \boldsymbol{a}$. When addressing the ID problem with linear block (channel) codes, then the channel distortion is the only possible cause for an erroneous equality estimate $\hat{v}$. This is in contrast to noisy-ID codes that are more efficient than channel codes at the cost of an additional error cause (due to a "hashing" step) as Section IV will explain. We present an overview of all combinations of distortion effects on the block codeword $\boldsymbol{x}$, the message estimate $\hat{\boldsymbol{a}}$, and the equality estimate $\hat{v}$ (and by extension an overview of all error causes when addressing the ID problem with linear block [channel] codes) in Table VI. There are three alternatives for the effect of the distortion on the block codeword $\boldsymbol{x}$ and on the message estimate $\hat{\boldsymbol{a}}$: no distortion, message-estimate-neutral ($\hat{\boldsymbol{a}}$-neutral) distortion, and message-estimate-changing ($\hat{\boldsymbol{a}}$-changing) distortion. Furthermore, since the message estimate $\hat{\boldsymbol{a}}$ leads to the equality estimate $\hat{v}$, the distortion can have three effects on the equality estimate $\hat{v}$: no distortion, equality-estimate-neutral ($\hat{v}$-neutral) distortion, and equality-estimate-changing ($\hat{v}$-changing) distortion. In the following, we explain the distortion effects in detail.

1) Correct Recovery of the Message: If the channel does not distort the block codeword $\boldsymbol{x}$ at all, then the receiver obtains the undistorted block codeword $\boldsymbol{y} = \boldsymbol{x}$ and can recover the message $\boldsymbol{a}$ of the sender by the message estimate $\hat{\boldsymbol{a}} = \boldsymbol{a}$. If the distortion of the channel is not severe (is minor), then the block code can overcome the *message-estimate-neutral* distortion. While the received codeword $\boldsymbol{y}$ is distorted, i.e., $\boldsymbol{y} \neq \boldsymbol{x}$, the block decoder can still recover the message $\boldsymbol{a}$ of the sender such that $\hat{\boldsymbol{a}} = \boldsymbol{a}$. When addressing the ID problem with linear block codes, the verifier at the receiver indirectly determines the result of $\boldsymbol{a} = \boldsymbol{b}$ when it computes $\hat{\boldsymbol{a}} = \boldsymbol{b}$ in the no-distortion and estimate-neutral-distortion cases (as the block decoder recovers the correct message $\hat{\boldsymbol{a}} = \boldsymbol{a}$ for both). Hence, the verifier computes the equality estimate $\hat{v}$ correctly and yields a true positive equality estimate $\hat{v}_\mathrm{p} = v_\mathrm{p}$ for matching messages (with the true equality v_p, i.e., $\boldsymbol{a} = \boldsymbol{b}$) or a true negative equality estimate $\hat{v}_\mathrm{n} = v_\mathrm{n}$ for mismatching messages (with "true inequality" v_n, i.e., $\boldsymbol{a} \neq \boldsymbol{b}$).

2) Incorrect Recovery of the Message: If the channel $\boldsymbol{W}$ distorts the block codeword $\boldsymbol{x}$ significantly, then the block decoder is not able to recover the message $\boldsymbol{a}$ from the heavily distorted codeword $\boldsymbol{y} \neq \boldsymbol{x}$. In the case of *message-estimate-changing* distortion, the message estimate $\hat{\boldsymbol{a}}$ is incorrect, i.e., $\hat{\boldsymbol{a}} \neq \boldsymbol{a}$. Thereby, the comparison operation in the verifier does not determine whether $\boldsymbol{a} = \boldsymbol{b}$, but whether a different message $\hat{\boldsymbol{a}} \neq \boldsymbol{a}$ matches the receiver's message $\boldsymbol{b}$, i.e., whether $\hat{\boldsymbol{a}} = \boldsymbol{b}; \hat{\boldsymbol{a}} \neq \boldsymbol{a}$.

For matching messages $\boldsymbol{a} = \boldsymbol{b}$, the message-estimate-changing distortion ($\hat{\boldsymbol{a}} \neq \boldsymbol{a}$) always leads to a message estimate $\hat{\boldsymbol{a}}$ that differs from the message $\boldsymbol{b}$ of the receiver. That is because the receiver is only interested in one specific message $\boldsymbol{b}$, and the distortion changes the estimate $\hat{\boldsymbol{a}}$ from that specific message $\boldsymbol{a} = \boldsymbol{b}$ to a different message $\boldsymbol{a} \neq \boldsymbol{b}$ that the receiver therefore is never interested in. Hence, for true equality v_p and a message-estimate-changing distortion, the verifier always determines a false-negative equality estimate $\hat{v}_\mathrm{fn}$, i.e., misses the true equality between the messages. The distortion causes an error.

On the other hand, for mismatching messages $\boldsymbol{a} \neq \boldsymbol{b}$, message-estimate-changing distortion can lead to one of two results: The incorrect message estimate $\hat{\boldsymbol{a}}$ is (like the original message $\boldsymbol{a}$) different from the receiver's message $\boldsymbol{b}$. In this case, the message-estimate-changing distortion does not result in a false equality estimate but in a TN equality estimate $\hat{v}_\mathrm{tn}$, because the estimated message $\hat{\boldsymbol{a}}$ and the receiver's message $\boldsymbol{b}$ mismatch despite the incorrect message estimate. Thereby, the distortion is message-estimate-changing, but equality-estimate-neutral, as the distortion does not interfere with the overall goal that is determining the equality estimate $\hat{v}$.

For the other possible result of a message-estimate-changing distortion given mismatching messages $\boldsymbol{a} \neq \boldsymbol{b}$, the message estimate $\hat{\boldsymbol{a}}$ accidentally matches the message $\boldsymbol{b}$ of the receiver. In that case, the verifier concludes a false positive equality estimate $\hat{v}_\mathrm{fp}$. In other words, the verifier incorrectly estimates that $\boldsymbol{a} = \boldsymbol{b}$ (matching messages) even though $\boldsymbol{a} \neq \boldsymbol{b}$.

The FP case (for mismatching messages $\boldsymbol{a} \neq \boldsymbol{b}$) is typically less likely than the FN case because there are many message estimates $\hat{\boldsymbol{a}}$ (specifically, $N-1$ message estimates $\hat{\boldsymbol{a}}$) that the block decoder can erroneously decode to that differ from the message $\boldsymbol{b}$ of the receiver. In contrast, there is only a single message estimate $\hat{\boldsymbol{a}}$ that equals the message $\boldsymbol{b}$ of the receiver such that the receiver concludes an FP equality estimate $\hat{v}_{\text{fp}}$. For mismatching messages $\boldsymbol{a} \neq \boldsymbol{b}$, it is typically much more likely that the distortion is equality-estimate-neutral, i.e., the message is recovered incorrectly, but the equality estimate $\hat{v}$ is still a true negative estimate $\hat{v}_{\text{n}}$.

3) Summary: In summary, if there is either no distortion or the linear block (channel) code overcomes the distortion, then the equality estimate $\hat{v}$ is always correct. This is because when addressing the ID problem with channel codes, then the channel distortion is the only cause for errors. In contrast, addressing the ID problem with noisy-ID codes is more efficient but adds a second cause for errors due to a "hashing" step as Section IV will explain. If the distortion exceeds the block code's capability of forward error correction, then the result depends on whether the messages of the sender and the receiver ($\boldsymbol{a}$ and $\boldsymbol{b}$) match or not. If messages match (true equality v_{p}), then the (uncorrectable) distortion always leads to an incorrect equality estimate $\hat{v}$. If the messages mismatch (true inequality v_{n}), then most likely the (uncorrectable) distortion has no negative effect, i.e., does not cause a false equality estimate $\hat{v}$. However, sometimes the channel can distort the message estimate $\hat{\boldsymbol{a}}$ to accidentally match the message $\boldsymbol{b}$ of the receiver such that the verifier falsely concludes matching messages, i.e., a false positive equality estimate $\hat{v}_{\text{fp}}$.

F. Transmission Capacity

The efficiency of addressing the ID problem with linear block codes is limited by the transmission capacity of the channel that connects the sender with the receiver. The transmission capacity C_{DMC} of the DMC $\boldsymbol{W}$ is an upper-bound on the rate R_{fec} of the block code:

$$R_{\text{fec}} = \frac{k}{n} = \frac{k'}{n'} \leq C_{\text{DMC}} \leq 1. \tag{13}$$

For rates R_{fec} beyond the capacity C_{DMC} of a DMC, reliable message transmission (i.e., message transmission with an error probability that approaches zero) is impossible [2]. In other words, the error probability cannot be arbitrarily small beyond the capacity C_{DMC} and the transmission will likely be subject to errors. This fundamental limit to communication is referred to as the noisy channel coding theorem, or as the Shannon limit in reference to Claude Shannon who first described this limit [2]. Due to the Shannon limit, the block code cannot achieve arbitrary rates R_{fec}, and therefore the block code cannot support an infinite number of messages while also providing reliable transmission with low error probability. Furthermore, the best achievable scaling of the rate, i.e., the achievable ratio between the size k of the message and the size n of the block codeword, is the exponential scaling from Eq. (8). For the ID problem, noisy-ID codes can achieve double-exponential scaling in their code rate R_{id} as the following Section IV will explain. This enables noisy-ID codes to address the ID problem much more efficiently than linear block codes.

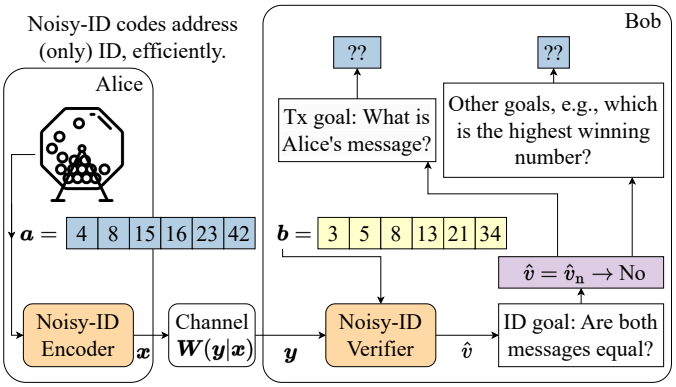


Fig. 7. Example of message identification via noisy-ID codes. Bob can use the received noisy-ID codeword $\boldsymbol{y}$ to determine that the messages of Alice and Bob are different, i.e., to determine a negative equality estimate $\hat{v}_{\text{n}}$. Other communication goals cannot be achieved using noisy-ID codes. Rather, a block code would be required, cf. Fig. 4.

G. Summary

A linear block code can address the distortion of a channel to enable reliable transmission. For this, the linear block encoder maps a message $\boldsymbol{a}$ of length k to a codeword $\boldsymbol{x}$ of length $n > k$, thereby adding redundant symbols. This redundancy enables the receiver to recover the encoded message even from a distorted version $\boldsymbol{y}$ of the sender's codeword $\boldsymbol{x}$. This is possible because valid codewords $\boldsymbol{x}$ (that are part of the codebook) have a high Hamming distance from each other, i.e., each pair of codebook codewords does not have too many symbols in common. If a received codeword $\boldsymbol{y}$ is not part of the codebook, then the receiver determines which valid codeword has the smallest distance from the received codeword $\boldsymbol{y}$. This valid codeword corresponds to a message that the decoder determines as the message estimate $\hat{\boldsymbol{a}}$.

Using a linear block code, the sender can reliably transmit its message $\boldsymbol{a}$ to the receiver. In ID, the receiver can compare the decoded message estimate $\hat{\boldsymbol{a}}$ to its own message $\boldsymbol{b}$ and determine whether there is a match, i.e., whether $\hat{\boldsymbol{a}} = \boldsymbol{b}$. Thereby, linear block codes can enable reliable ID via noisy channels. As Section IV will explain, noisy-ID codes can enable reliable ID much more efficiently than channel codes, such as linear block codes.

IV. Noisy-ID Codes

A. Overview

When addressing the ID problem with linear block codes, cf. Section III, the sender transmits the message to the receiver. In the lottery example, this corresponds to Alice sending the winning numbers to Bob. However, to address the lottery example, Bob does not need to know *which* lottery numbers $\boldsymbol{a}$ won, but only *whether* Bob's numbers won. In other words, it is not necessary to recover the message $\boldsymbol{a}$ of the sender via the message estimate $\hat{\boldsymbol{a}}$. Rather, it is sufficient to determine from the received codeword $\boldsymbol{y}$ the equality estimate $\hat{v}$ with low

TABLE VII
SUMMARY OF NOTATIONS FOR RANDOMIZED NOISY-ID CODES.

$Q_{\boldsymbol{a}}$	Encod. probab. distr. of message $\boldsymbol{a}$ of the sender on $\mathcal{X}$
$\mathcal{E}_{\boldsymbol{a}}$	$\subset \mathcal{X}$, Encoder subset of codewords of message $\boldsymbol{a}$
$f_{\text{noisy-id}}$	Encod. function mapping message $\boldsymbol{a}$ to codeword $\boldsymbol{x}$
$\mathcal{V}_{\boldsymbol{b}}$	$\subset \mathcal{Y} = \mathcal{X}$, Verifier subset of codewords of message $\boldsymbol{b}$
$\boldsymbol{x}_{\text{tp}}$	Codeword that results in a true-positive estimate $\hat{v}_{\text{tp}}$
$\boldsymbol{x}_{\text{fp}}$	Codeword that results in a false-positive estimate $\hat{v}_{\text{fp}}$
$\boldsymbol{x}_{\text{tn}}$	Codeword that results in a true-negative estimate $\hat{v}_{\text{tn}}$
$p_{\text{fn}}(\boldsymbol{a})$	Prob. of a false-negative error for message $\boldsymbol{a}$
$p_{\text{fp}}(\boldsymbol{a}, \boldsymbol{b})$	Prob. of a false-positive error for pair $(\boldsymbol{a}, \boldsymbol{b})$
λ_{fn}	Limit of FN error probs. for all messages
λ_{fp}	Limit of FP error probs. for all pairs

error probability. Instead of transmitting the winning lottery numbers to Bob, in the noisy-ID coding paradigm, Alice sends only a smaller codeword that suffices to enable Bob to reliably determine whether Bob won the lottery. Thereby, noisy-ID codes include ID-specific encoding. Since the encoding is specific to the ID problem, a noisy-ID code is a goal-oriented code.

The noisy-ID encoder maps the message $\boldsymbol{a}$ of the sender to a noisy-ID codeword $\boldsymbol{x}$, cf. Fig. 7. The channel $\boldsymbol{W}$ distorts the noisy-ID codeword $\boldsymbol{x}$ and the receiver obtains the distorted noisy-ID codeword $\boldsymbol{y}$. The noisy-ID verifier computes the equality estimate $\hat{v}$ by determining whether the received noisy-ID codeword $\boldsymbol{y}$ is associated with the message $\boldsymbol{b}$ of the receiver. Thereby, the noisy-ID verifier performs the verification step directly with the received noisy-ID codeword $\boldsymbol{y}$. This integrated verification step contrasts the verification with linear block codes that Section III-D explained. When addressing the ID problem with linear block codes, the receiver (unnecessarily) estimates the message $\boldsymbol{a}$ of the sender and only then determines the equality estimate $\hat{v}$ by comparing the estimated message $\hat{\boldsymbol{a}}$ and the message $\boldsymbol{b}$ of the receiver, cf. Fig. 4.

When addressing the ID problem with linear block codes, the receiver's block decoder has to decide between N different message estimates $\hat{\boldsymbol{a}}$, cf. Eq. (10); in contrast, in noisy-ID coding, the noisy-ID verifier selects one of only two possible results (equality estimates): match $\hat{v}_{\text{p}}$ and mismatch $\hat{v}_{\text{n}}$. Deciding between only the two options for the equality estimate $\hat{v}$ is a simpler communication problem than message transmission as Ja Ja found in [27]. Inspired by Ja Ja's investigation, Ahlswede and Dueck showed in their award-winning paper Identification via Channels [1] that the specific communication goal of identification can be accomplished more efficiently than by message transmission via block codes. More specifically, Ahlswede and Dueck [1] proved the existence of (goal-oriented) noisy-ID codes that have up to exponential gains over (goal-agnostic) block codes to address the ID problem over noisy channels. Since noisy-ID codes are optimized for ID, they can outperform the general-purpose block codes in the ID problem. However, because the noisy-ID verifier determines the binary estimate $\hat{v}$ of the equality v (and not an estimate $\hat{\boldsymbol{a}}$ of the sender's message $\boldsymbol{a}$), the receiver can only address the ID problem and no other communication goal, as Fig. 7 visualizes.

Noisy-ID codes are tailored for the ID problem and can be more efficient than block codes at accomplishing the communication goal that is ID. Noisy-ID codes with randomized encoding (i.e., randomized noisy-ID codes) can have up to exponential gains in efficiency over block codes at addressing the ID problem. In this context, efficiency refers to the number N of messages that the noisy-ID code supports (i.e., the benefit) by transmitting noisy-ID codewords $\boldsymbol{x}$ of size n, whereby the noisy-ID codeword size n corresponds to the incurred traffic (i.e., the cost). An efficient code supports a large number N of messages and only sends codewords of a small size n. Because a noisy-ID code can be more efficient than a block code at addressing the ID problem, the noisy-ID code can support more messages (i.e., larger N) than a block code given the same codeword size n. Alternatively, the noisy-ID code can support the same number N of messages as a block code but requires only a smaller codeword size n to identify a message reliably, or the noisy-ID code supports a higher number N of messages *and* uses a smaller codeword size n than the block code. The efficiency gain comes at the cost of an additional, limited error probability. The ID encoding is tailored to the ID problem (i.e., goal-oriented). Despite the loss of information in the encoding, reliable identification is possible. Section IV-D explains the error probabilities in detail.

Noisy-ID codes that include randomization in the noisy-ID encoder have the potential to outperform block codes up to exponentially, and are therefore the main focus of practical research on noisy-ID codes. Noisy-ID codes without randomization (i.e., deterministic noisy-ID codes) are limited to only logarithmic gains over block codes, and have therefore received less attention so far. Channel codes do not benefit from randomization [102] and are therefore typically deterministic. We leave the description of deterministic noisy-ID codes (that have received less attention) to Section IV-H.

In this section, we describe principles of randomized noisy-ID encoding and verification in contrast to the deterministic block coding approach that we explained in Section III. We leave the details of how to explicitly construct such randomized noisy-ID codes to Sections V and VII, as visualized in Fig. 3. As Section V will explain in detail, it is possible (and practical) to construct a noisy-ID code by concatenating a goal-agnostic channel code (such as a linear block code) with a noiseless-ID code that performs the lossy ID-specific encoding. This Section IV focuses on the overall noisy-ID code, i.e., a code that jointly performs channel coding and the ID-specific encoding.

B. Noisy-ID Encoding

The randomized noisy-ID encoder selects a random noisy-ID codeword $\boldsymbol{x}$ that encodes the message $\boldsymbol{a}$ of the sender. The selection of the random codeword can be framed as selecting a random noisy-ID codeword from a subset of codewords, or equivalently as sampling from a probability distribution. We first explain the randomized noisy-ID encoding process in the subset representation and visualize this framing in Fig. 8.

1) Subset Representation: The sender first selects a message $\boldsymbol{a}$ from the set $\mathcal{U}$ of messages. The set $\mathcal{U}$ of messages

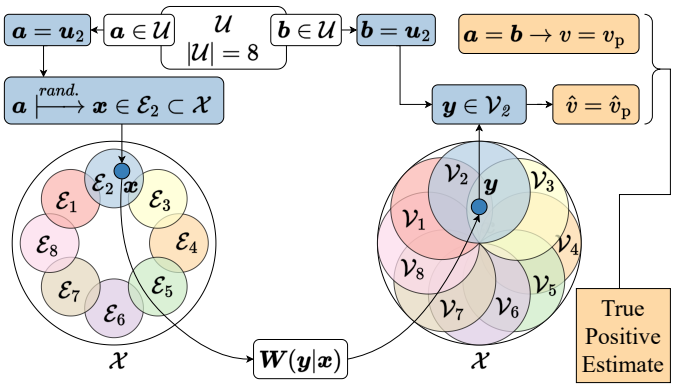


Fig. 8. Schematic view of randomized noisy-ID encoding and verification. To address the ID problem, the sender selects a random codeword according to the randomized encoding function, cf. Eq. (14). The DMC $\boldsymbol{W}$ distorts the selected codeword $\boldsymbol{x}$. The receiver uses the received codeword $\boldsymbol{y}$ to determine the equality estimate via Eq. (19), whether the two messages match or not. Note that codeword subsets overlap in contrast to block codes, cf. Fig. 5.

supported by the noisy-ID code in the example in Fig. 8 holds significantly more messages than the set of messages supported by the linear block code in the example in Fig. 5 to highlight that noisy-ID codes can be more efficient than block codes at addressing the ID problem. Generally, the randomized ID encoder selects random noisy-ID codewords $\boldsymbol{x}$ from the set $\mathcal{X}$ of noisy-ID codewords. Each message $\boldsymbol{u} \in \mathcal{U}$ is associated with a subset $\mathcal{E}_{\boldsymbol{u}}$ of noisy-ID codewords. For example, the message $\boldsymbol{u}_2$ is associated with the subset $\mathcal{E}_2$ of noisy-ID codewords whereby we write $\mathcal{E}_2$ instead of $\mathcal{E}_{\boldsymbol{u}_2}$ to keep the notation concise. The *subset of noisy-ID codewords* contrasts the *single linear block codeword* that deterministic block codes associate with each message. Each subset $\mathcal{E}_{\boldsymbol{u}}$ of noisy-ID codewords associated with an arbitrary message $\boldsymbol{u} \in \mathcal{U}$ is a subset of the set $\mathcal{X}$ of all possible noisy-ID codewords, i.e., $\mathcal{E}_{\boldsymbol{u}} \subset \mathcal{X}$. From the subset $\mathcal{E}_{\boldsymbol{a}}$ associated with the message $\boldsymbol{a}$ of the sender, the noisy-ID encoder randomly selects a noisy-ID codeword $\boldsymbol{x}$, according to a randomized noisy-ID encoding function

$$f_{\text{noisy-id}} : \boldsymbol{a} \overset{\text{rand.}}{\longmapsto} \boldsymbol{x} \in \mathcal{E}_{\boldsymbol{a}} \subset \mathcal{X} \quad \forall \boldsymbol{a} \in \mathcal{U}, \tag{14}$$

whereby the operator $\overset{\text{rand.}}{\longmapsto}$ describes a random mapping. Just as the linear block encoding function f_fec, cf. Eq. (7), the randomized noisy-ID encoding function $f_{\text{noisy-id}}$ maps from the domain $\mathcal{U}$ (the set of messages) to the codomain $\mathcal{X}$ (the set of codewords). Formally, the randomized noisy-ID encoding function is not a function because its output (the noisy-ID codeword $\boldsymbol{x}$) is not deterministic. However, (imperfectly) framing noisy-ID encoding as a function facilitates comparison to the linear block encoding function.

The union of all subsets $\mathcal{E}_{\boldsymbol{u}}$ of noisy-ID codewords is the noisy-ID encoder codebook $\mathcal{B}$ that holds all noisy-ID codewords that are encoding images of messages $\boldsymbol{u} \in \mathcal{U}$, i.e., the encoding codebook $\mathcal{B}$ is the image of the noisy-ID encoding function. The subsets $\mathcal{E}_{\boldsymbol{u}}$ of different messages $\boldsymbol{u}$ are not disjoint as Fig. 8 visualizes. Thereby, the randomized noisy-ID encoder can randomly map different messages to the same noisy-ID codeword. Conversely, the encoding preimage of each noisy-ID codeword (that is part of the noisy-ID codebook $\mathcal{B}$) consists of several different messages. The ambiguity of the encoding preimage can (indirectly via the verification preimage that Section IV-C explains and that is related to the encoding preimage) cause errors in the equality estimate $\hat{v}$ (in addition to errors due to the channel distortion) but enables the efficiency gain of noisy-ID codes over block codes when addressing the ID problem. Section IV-D will explain the different error types in detail.

Overall, the randomized noisy-ID encoder performs two steps: 1) a deterministic mapping from the selected message $\boldsymbol{a}$ to the associated subset $\mathcal{E}_{\boldsymbol{a}}$ of noisy-ID codewords, and 2) random selection of a noisy-ID codeword $\boldsymbol{x}$ from the associated noisy-ID codeword subset $\mathcal{E}_{\boldsymbol{a}}$, cf. Fig. 8.

2) Probabilistic Representation: It is also possible (and mathematically rigorous in contrast to the "randomized function" $f_{\text{noisy-id}}$) to represent the noisy-ID encoding via a family of probability distributions Q, whereby the probabilistic representation is equivalent to the subset representation in Section IV-B1. In the probabilistic representation, each message $\boldsymbol{a}$ is associated with a different probability distribution $Q_{\boldsymbol{a}}$ on the noisy-ID encoder's set $\mathcal{X}$ of noisy-ID codewords. Instead of encoding the message $\boldsymbol{a}$ into a deterministic codeword $\boldsymbol{x}$ as done by block encoders, the noisy-ID encoder samples a noisy-ID codeword from the probability distribution $Q_{\boldsymbol{a}}$ on the set $\mathcal{X}$ of noisy-ID codewords. The probability distribution $Q_{\boldsymbol{a}}$ assigns a non-zero probability to some noisy-ID codewords, and a probability of 0 to the other noisy-ID codewords.

The subset $\mathcal{E}_{\boldsymbol{a}}$ of noisy-ID codewords that are associated with the message $\boldsymbol{a}$ is equivalent to the support of the probability distribution $Q_{\boldsymbol{a}}$. In other words, the union of all noisy-ID codewords that have a non-zero probability in the probability distribution $Q_{\boldsymbol{a}}$ forms a subset $\mathcal{E}_{\boldsymbol{a}}$.

3) Noisy-ID Code Rate: The codeword size n of a noisy-ID codeword is typically smaller than the codeword size n of a block code that supports the same number of messages. The noisy-ID code rate $R_{\text{noisy-id}}$ reflects this efficiency gain of randomized noisy-ID codes over block codes in comparison with the block code rate R_fec as defined in Eq. (8). Ahlswede and Dueck [1] proved that randomized noisy-ID codes can achieve the following noisy-ID rate $R_{\text{noisy-id}}$ and still approach error-free identification:

$$R_{\text{noisy-id}} = \frac{\log_2\left(\log_2(N)\right)}{n'} = \frac{\log_2\left(k'\right)}{n'} \leq 1. \tag{15}$$

This noisy-ID rate is also referred to as a second-order rate, because the ID rate scales with two concatenated logarithms as opposed to the single logarithm scaling of the block code rate R_fec, cf. Eq. (8). A randomized noisy-ID code can therefore have a double-exponential scaling of the number N of supported messages when increasing the size n of the transmitted noisy-ID codeword $\boldsymbol{x}$:

$$N = 2^{2^{n' R_{\text{noisy-id}}}}, \tag{16}$$

whereas the number N of messages supported by a block code only scales single-exponentially, cf. Eq. (9).

Since we describe block coding and noisy-ID coding using q-ary messages and q-ary codewords, we reformulate the

noisy-ID rate for the q-ary representation. The noisy-ID code supports $N = q^k$ messages and has a noisy-ID code rate of

$$R_{\text{noisy-id}} = \frac{\log_2\left(k \log_2(q)\right)}{n \log_2(q)} \leq 1. \tag{17}$$

Note that depending on the rate $R_{\text{noisy-id}}$ of the noisy-ID code, the received codeword $\hat{\boldsymbol{x}}$ could theoretically enable the receiver to decode the message $\boldsymbol{a}$ of the sender. However, for every "good" noisy-ID code, i.e., noisy-ID codes with reasonably high noisy-ID rates, the received codeword $\boldsymbol{y}$ has too little entropy to reliably decode the message $\boldsymbol{a}$ of the sender. "Good" noisy-ID codes operate beyond the Shannon limit, i.e., their transmission rate $R_{\text{fec}} = k/n > C$. In other words, typically, the size k of the message exceeds the size n of the noisy-ID codeword, i.e., $k > n$. Therefore, it is impossible to reliably *decode* the message $\boldsymbol{a}$ from the codeword $\boldsymbol{x}$. Rather, the received codeword suffices for reliable *identification*.

C. Noisy-ID Verification

After randomly selecting a noisy-ID codeword $\boldsymbol{x}$, the sender transmits the randomly selected codeword $\boldsymbol{x}$ over the discrete memoryless channel $\boldsymbol{W}(\boldsymbol{y}|\boldsymbol{x})$ (as introduced in Section II-A) to the receiver, cf. Fig. 8. In the transmission process, the channel $\boldsymbol{W}(\boldsymbol{y}|\boldsymbol{x})$ distorts the codeword $\boldsymbol{x}$ such that the receiver receives a distorted codeword $\boldsymbol{y}$.

The receiver selects its own message $\boldsymbol{b}$ from the set $\mathcal{U}$ of messages, cf. Fig. 8. Based on the received distorted codeword $\boldsymbol{y}$, the receiver wants to verify whether the sender also selected the message $\boldsymbol{b} \in \mathcal{U}$, i.e., whether $\boldsymbol{b} = \boldsymbol{a}$. To this end, the receiver creates a verifier noisy-ID codeword subset $\mathcal{V}_{\boldsymbol{b}} \subset \mathcal{Y} = \mathcal{X}$ associated with the selected message $\boldsymbol{b}$, cf. Fig. 8. The verifier determines whether the received distorted codeword $\boldsymbol{y}$ is part of the verifier noisy-ID codeword subset $\mathcal{V}_{\boldsymbol{b}}$ associated with the selected message $\boldsymbol{b}$.

For the "zero-mean" DMC, cf. Section II-D, distortion-free transmission of the codeword $\boldsymbol{x}$, i.e., $\boldsymbol{y} = \boldsymbol{x}$, is the baseline, and distorted received codewords $\boldsymbol{y}$ spread around the transmitted codeword $\boldsymbol{x}$. Therefore, similar to the decoding subsets $\mathcal{D}_{\boldsymbol{u}}$ in linear block codes that are centered on their associated block codeword $\boldsymbol{x}_{\boldsymbol{u}}$, cf. Fig. 5, it is a reasonable choice to center each verifier subset $\mathcal{V}_{\boldsymbol{u}}$ on its associated encoding subset $\mathcal{X}_{\boldsymbol{u}}$. Each verifier subset $\mathcal{V}_{\boldsymbol{u}}$ (for example in Fig. 8) is roughly centered on its corresponding encoding subset $\mathcal{X}_{\boldsymbol{u}}$. To mitigate the channel's distortion, in addition to all codewords that are part of the corresponding encoding subset $\mathcal{E}_{\boldsymbol{u}}$, the verifier subsets $\mathcal{V}_{\boldsymbol{u}}$ include codewords that are slightly distorted versions of the codewords in the corresponding encoding subset $\mathcal{E}_{\boldsymbol{u}}$. In other words, the verifier subsets $\mathcal{V}_{\boldsymbol{u}}$ include codewords that have a small Hamming distance D from codewords in the corresponding encoding subset $\mathcal{E}_{\boldsymbol{u}}$. Thereby, for any message $\boldsymbol{u}$, the encoding subset $\mathcal{E}_{\boldsymbol{u}}$ is a subset of the associated verifier subset $\mathcal{V}_{\boldsymbol{u}}$:

$$\mathcal{E}_{\boldsymbol{u}} \subset \mathcal{V}_{\boldsymbol{u}} \subset \mathcal{X} \; \forall \boldsymbol{u} \in \mathcal{U}. \tag{18}$$

We visualize this property also in Fig. 8.

This way, for all codewords $\boldsymbol{y}$ that correspond to undistorted transmission of any codeword $\boldsymbol{x} \in \mathcal{E}_{\boldsymbol{b}}$ in the encoding subset of the *receiver message* $\boldsymbol{b}$, the verifier concludes a "match" equality estimate $\hat{v}_{\text{p}}$, i.e., $\hat{v}_{\text{p}}$ if $\boldsymbol{y} \in \mathcal{E}_{\boldsymbol{b}}$. Additionally, to mitigate the effects of possible distortion in the channel $\boldsymbol{W}$, codewords $\boldsymbol{y}$ that have a small Hamming distance D from any codeword $\boldsymbol{x} \in \mathcal{E}_{\boldsymbol{b}}$ also lead to a positive equality estimate $\hat{v}_{\text{p}}$. Thereby, if the received codeword $\boldsymbol{y}$ is part of the subset $\mathcal{V}_{\boldsymbol{b}}$ of the selected message $\boldsymbol{b}$, then the receiver concludes that the messages $\boldsymbol{a}$ and $\boldsymbol{b}$ are identical, i.e., its estimate $\hat{v} = \hat{v}_{\text{p}}$ is positive, cf. Table IV. Otherwise, the receiver concludes that the messages must be different, i.e., its estimate $\hat{v} = \hat{v}_{\text{n}}$ is negative. Overall, in contrast to the equality estimate via block codes in Eq. (12), the receiver determines its estimate $\hat{v}$ as:

$$\hat{v} = \begin{cases} \hat{v}_{\text{p}} \text{ if } \boldsymbol{y} \in \mathcal{V}_{\boldsymbol{b}}, \\ \hat{v}_{\text{n}} \text{ if } \boldsymbol{y} \notin \mathcal{V}_{\boldsymbol{b}}. \end{cases} \tag{19}$$

This highlights that identification is a simpler communication problem than transmission, as explained in Section II-F. For identification, the received codeword $\boldsymbol{y}$ is classified only into one of two classes, whereas for transmission the received codeword $\boldsymbol{y}$ is classified into one of $N = |\mathcal{U}|$ classes, cf. Eq. (10). For closer alignment with the formulation in Eq. (10), we reformulate Eq. (19) to

$$\hat{v} = \begin{cases} \hat{v}_{\text{p}} \text{ if } \boldsymbol{y} \in \mathcal{V}_{\text{p}} = \mathcal{V}_{\boldsymbol{b}}, \\ \hat{v}_{\text{n}} \text{ if } \boldsymbol{y} \in \mathcal{V}_{\text{n}} = \mathcal{Y} \setminus \mathcal{V}_{\boldsymbol{b}} = \mathcal{X} \setminus \mathcal{V}_{\boldsymbol{b}}, \end{cases} \tag{20}$$

whereby $\mathcal{V}_{\text{p}}$ is the set of received codewords that results in a positive verification $\hat{v}_{\text{p}}$ and $\mathcal{V}_{\text{n}}$ is the set of received codewords that results in a negative verification $\hat{v}_{\text{n}}$.

In terms of framing the creation of a code as a sphere packing problem (cf. Section III-B), the overlap in the noisy-ID verifier codeword subsets $\mathcal{V}_{\boldsymbol{u}}$ allows packing the spheres (subsets) of more messages into the codeword set $\mathcal{X}$ than block codes allow. Instead of looking for decoding subsets $\mathcal{D}_{\boldsymbol{u}}$ of block codewords that at most "touch" each other; in noisy-ID codes, the verifier subsets $\mathcal{V}_{\boldsymbol{u}}$ can overlap as long as the overlap is not too large. Thereby, each noisy-ID codeword is an element of the verifier subsets $\mathcal{V}_{\boldsymbol{u}}$ of many messages $\boldsymbol{u}$. In other words, the verification preimage of each noisy-ID codeword holds many messages $\boldsymbol{u}$. The (ideally small) overlap of subsets $\mathcal{V}_{\boldsymbol{u}}$ between different messages $\boldsymbol{u}$ enables the efficiency gain of noisy-ID codes by making more messages identifiable compared to block codes while using the same resources (codeword size n). The cost for this efficiency gain in terms of codeword size n lies in additional errors that the overlap of verifier subsets $\mathcal{V}_{\boldsymbol{u}}$ can introduce.

D. Error Types

There are two sources for errors when addressing the ID problem with randomized noisy-ID codes: i) the distortion of the codeword $\boldsymbol{x}$ in the channel $\boldsymbol{W}$ as known from block codes, cf. Table VI, and ii) the random choice of a noisy-ID codeword $\boldsymbol{x}$ from possibly overlapping noisy-ID codeword subsets $\mathcal{E}_{\boldsymbol{u}}$ (in contrast to linear block codes). For either case, one has to distinguish between the false-negative error $\hat{v}_{\text{fn}}$ associated with matching messages (true equality v_{p}) and the false-positive error $\hat{v}_{\text{fp}}$ associated with mismatching messages (true inequality v_{n}).

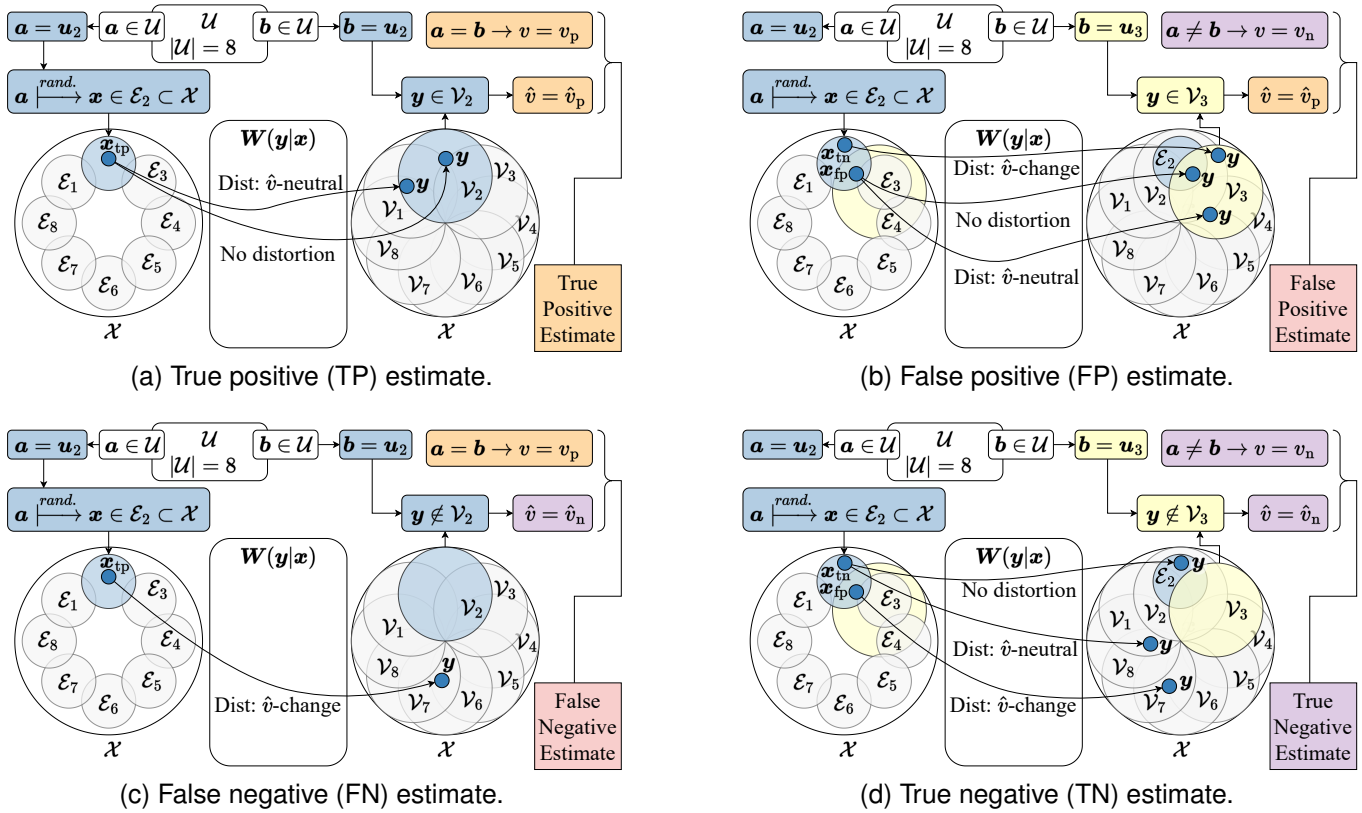


Fig. 9. Schematic view of encoding and verifying with randomized noisy-ID codes: four possible estimates, cf. Table IV. The channel $\boldsymbol{W}$ can have three different effects on the transmitted codeword $\boldsymbol{x}$: no distortion, distortion that results in a change of the estimate $\hat{v}$ compared to undistorted transmission of the codeword $\boldsymbol{x}$, and distortion that does not change the estimate $\hat{v}$, i.e., an estimate-neutral distortion. When the message pair mismatches (Figs. 9b and 9d), the codeword $\boldsymbol{x}$ that the encoder randomly selects is either a false-positive codeword $\boldsymbol{x}_{\text{fp}}$ that results in an FP estimate $\hat{v}_{\text{fp}}$ if transmitted without distortion, or a true-negative codeword $\boldsymbol{x}_{\text{tn}}$ that results in a TN estimate $\hat{v}_{\text{tn}}$ if transmitted without distortion.

Section III-E explained the effect of the channel distortion on codewords and the equality estimate $\hat{v}$. In this subsection, next to the distortion, the choice of a random noisy-ID codeword $\boldsymbol{x}$ is a secondary cause for errors. We will explain the interaction between distortion and the random choice of the noisy-ID codeword $\boldsymbol{x}$, but first consider only distortion-free transmission of the codeword $\boldsymbol{x}$ as a baseline. Based on Fig. 8, Fig. 9 visualizes the four possible equality estimates $\hat{v}$, cf. Table IV, and three distortion types: no distortion, equality-estimate-neutral ($\hat{v}$-neutral) distortion, and equality-estimate-changing ($\hat{v}$-changing) distortion as explained in Section III-E.

Thereby, addressing the ID problem with noisy-ID codes can be structured into nine different cases whereby each case is a combination of i) one of the three distortion types, ii) the Boolean equality value v whether the messages truly match ($\boldsymbol{a} = \boldsymbol{b}$), and iii) (only for mismatching messages $\boldsymbol{a} \neq \boldsymbol{b}$) the choice of the encoding codeword $\boldsymbol{x}$. Each of these nine cases corresponds to one of the labeled arrows in Fig. 9. In contrast, addressing the ID problem with linear block codes can be structured into only six cases that are a combination of i) one of the three distortion types, and ii) the Boolean equality value v whether the messages truly match ($\boldsymbol{a} = \boldsymbol{b}$), cf. Table VI.

1) Distortion-Free Transmission: For matching messages $\boldsymbol{a} = \boldsymbol{b}$, the noisy-ID encoder selects a random codeword $\boldsymbol{x}$ from the encoding subset $\mathcal{E}_{\boldsymbol{a}}$ associated with the sender's message $\boldsymbol{a}$. The receiver creates the verifier subset $\mathcal{V}_{\boldsymbol{b}} = \mathcal{V}_{\boldsymbol{a}}$ associated with the receiver's message $\boldsymbol{b} = \boldsymbol{a}$. Since the encoding subset $\mathcal{E}_{\boldsymbol{a}}$ is a subset of the verifier subset $\mathcal{V}_{\boldsymbol{a}}$, cf. Eq. (18), if the codeword $\boldsymbol{x}$ is transmitted without distortion, then the received codeword $\boldsymbol{y} = \boldsymbol{x}$ is an element of $\mathcal{V}_{\boldsymbol{a}}$. Thereby, the receiver concludes a positive equality estimate $\hat{v}_{\text{p}}$ (that is a true positive equality estimate $\hat{v}_{\text{tp}}$ because the messages match), cf. Fig. 9a. It does not matter which codeword $\boldsymbol{x}$ the sender selects randomly from the encoding set $\mathcal{E}_{\boldsymbol{a}}$, because the sender can only select codewords $\boldsymbol{x} \in \mathcal{E}_{\boldsymbol{a}} \subset \mathcal{V}_{\boldsymbol{a}}$ that lead to a TP equality estimate $\hat{v}_{\text{tp}}$. We refer to codewords $\boldsymbol{x}$ (selected by the sender) that yield a TP equality estimate $\hat{v}_{\text{tp}}$ when transmitted without distortion as *TP codewords* $\boldsymbol{x}_{\text{tp}}$. No codewords $\boldsymbol{x}$ result in an FN estimate $\hat{v}_{\text{fn}}$ when transmitted without distortion (i.e., in Fig. 9a there is no "no distortion" case that yields an FN estimate $\hat{v}_{\text{fn}}$ as in Fig. 9c), because the sender can only select TP codewords $\boldsymbol{x}_{\text{tp}}$ by design. Thereby, for matching messages, there are no errors if there is no distortion (similar to there being no errors in linear block coding if there is no distortion).

For mismatching messages $\boldsymbol{a} \neq \boldsymbol{b}$, the pairwise overlap of codeword subsets can cause errors despite distortion-free transmission of the noisy-ID codeword $\boldsymbol{x}$. Specifically, the codeword $\boldsymbol{x}$ that the sender selects could either be a codeword

that causes a true negative equality estimate $\hat{v}_{\text{tn}}$, or a codeword that causes a false positive equality estimate $\hat{v}_{\text{fp}}$, i.e., an error. Analogous to TP codewords $\boldsymbol{x}_{\text{tp}}$ in the matching messages case, we refer to such codewords as *TN codewords* $\boldsymbol{x}_{\text{tn}}$ and as *FP codewords* $\boldsymbol{x}_{\text{fp}}$, respectively. Since the messages of sender and receiver differ, the receiver checks whether the (undistorted) received codeword $\boldsymbol{y} = \boldsymbol{x}$ is an element of the verifier subset $\mathcal{V}_{\boldsymbol{b}}$ associated with the receiver's message $\boldsymbol{b}$. A codeword $\boldsymbol{x}$ that the sender randomly selects from the encoding subset $\mathcal{E}_{\boldsymbol{a}}$ associated with the sender's message $\boldsymbol{a}$ can also be an element of the verifier subset $\mathcal{V}_{\boldsymbol{b}}$. Thereby, any codeword $\boldsymbol{x}$ is an FP codeword $\boldsymbol{x}_{\text{fp}}$ for the message pair $(\boldsymbol{a}, \boldsymbol{b})$ if the codeword $\boldsymbol{x}$ is an element of the overlap of the encoding subset $\mathcal{E}_{\boldsymbol{a}}$ with the verifier subset $\mathcal{V}_{\boldsymbol{b}}$, i.e., if $\boldsymbol{x} \in (\mathcal{E}_{\boldsymbol{a}} \cap \mathcal{V}_{\boldsymbol{b}})$. The examples in Figs. 9b and 9d visualize this property as the FP codeword $\boldsymbol{x}_{\text{fp}}$ lies in the overlap $(\mathcal{E}_{i_2} \cap \mathcal{V}_{i_3})$. Conversely, any codeword $\boldsymbol{x}$ is a TN codeword $\boldsymbol{x}_{\text{tn}}$ if it is *not* an element of the overlap of the encoding subset $\mathcal{E}_{\boldsymbol{a}}$ with the verifier subset $\mathcal{V}_{\boldsymbol{b}}$, i.e., if $\boldsymbol{x} \in (\mathcal{E}_{\boldsymbol{a}} \setminus \mathcal{V}_{\boldsymbol{b}})$. Since the sender does not know which message $\boldsymbol{b}$ the receiver selected, the sender does not know which codewords are FP codewords and cannot avoid selecting an FP codeword. Selecting an FP codeword corresponds to a hash collision in the terminology of hashes.

In noisy-ID codes, each codeword $\boldsymbol{x}$ has two preimages: the encoding preimage that holds all messages $\boldsymbol{u}$ that include that codeword $\boldsymbol{x}$ in their associated encoding subset $\mathcal{E}_{\boldsymbol{u}}$, and the verifier preimage that holds all messages $\boldsymbol{u}$ that include that codeword $\boldsymbol{x}$ in their associated verifier subset $\mathcal{V}_{\boldsymbol{u}}$. If the encoding preimage of the randomly selected codeword $\boldsymbol{x}$ includes the message $\boldsymbol{a}$ of the sender and the verifier preimage of the same codeword $\boldsymbol{x}$ includes the message $\boldsymbol{b}$ of the receiver (i.e., if $\boldsymbol{x} \in (\mathcal{E}_{\boldsymbol{a}} \cap \mathcal{V}_{\boldsymbol{b}})$), then that codeword $\boldsymbol{x}$ is an FP codeword $\boldsymbol{x}_{\text{fp}}$. Because the encoding and verifier preimages hold many messages and therefore codewords $\boldsymbol{x}$ are associated with many messages, each codeword $\boldsymbol{x}$ can take on different "roles" for different message pairs $(\boldsymbol{a}, \boldsymbol{b})$. In other words, a codeword $\boldsymbol{x}$ might be an FP codeword $\boldsymbol{x}_{\text{fp}}$ for one message pair, and a TN codeword $\boldsymbol{x}_{\text{tn}}$ for another message pair.

Linear block codes always map the message $\boldsymbol{a}$ to the same deterministic block codeword. In other words, linear block codes always select a TP codeword $\boldsymbol{x}_{\text{tp}}$ for matching messages, and always a TN codeword $\boldsymbol{x}_{\text{tn}}$ for mismatching messages. Randomized noisy-ID codes purposefully add the possibility of selecting FP codewords (for mismatching messages) because as long as the probability of selecting an FP codeword is limited, noisy-ID codes can perform reliable identification more efficiently than linear block codes.

2) Matching Messages under Distortion: Next to the no-distortion case that Section IV-D1 explained, the channel $\boldsymbol{W}$ can also distort the codeword $\boldsymbol{x}$. For an *estimate-neutral* distortion, the received codeword $\boldsymbol{y} \neq \boldsymbol{x}$. However, this distortion of the codeword does not result in an incorrect estimate $\hat{v}$. For example, if an estimate-neutral distortion interferes with a TP codeword $\boldsymbol{x}_{\text{tp}}$, then the estimate $\hat{v}$ remains TP as it would be without any distortion, cf. the example in Fig. 9a. The noisy-ID code is able to overcome the distortion of the channel as long as the distortion is not too severe and enables reliable identification (for matching messages).

However, if the distortion is too severe, then the noisy-ID code cannot overcome the noisy channel, much like a linear block code can only correct errors up to a certain level of distortion. In Fig. 9c, the channel $\boldsymbol{W}$ distorts the codeword $\boldsymbol{x}$ in an estimate-changing manner such that the receiver receives a codeword $\boldsymbol{y}$ that is not an element of the decoding subset $\mathcal{V}_2$ of the receiver's message $\boldsymbol{b} = \boldsymbol{u}_2$. The estimate-changing distortion directly causes an FN error.

Because the noisy-ID code selects only TP codewords $\boldsymbol{x}_{\text{tp}}$ by design, only the distortion of the channel $\boldsymbol{W}$ can cause an error for matching messages. The overlap of codeword subsets is irrelevant in this case.

3) Mismatching Messages under Distortion: As Section IV-D1 explained, the overlap of codeword subsets can cause FP errors for mismatching message pairs $(\boldsymbol{a}, \boldsymbol{b})$, $\boldsymbol{a} \neq \boldsymbol{b}$, even when the channel $\boldsymbol{W}$ does not distort the transmitted codeword $\boldsymbol{x}$. For the distortion-free case, an FP error occurs when the sender randomly selects an FP codeword $\boldsymbol{x}_{\text{fp}}$. For noisy-ID codes, the interaction of the channel distortion with the verification of mismatching messages can cause multiple outcomes. Generally, the severity of the distortion does *not* determine whether the receiver determines the equality estimate $\hat{v}$ correctly. A small distortion of a TN codeword $\boldsymbol{x}_{\text{tn}}$ can be an estimate changing distortion, as the top arrow in Fig. 9b visualizes. In this case, the distortion caused an error that would not have occurred without distortion. Conversely, a large distortion of a TN codeword $\boldsymbol{x}_{\text{tn}}$ can be an estimate-neutral distortion, as the middle arrow in Fig. 9d visualizes. The outcome depends on whether the distortion distorts the codeword into the verifier subset $\mathcal{V}_{\boldsymbol{b}}$ of the receiver's message.

If the sender randomly selects an FP codeword $\boldsymbol{x}_{\text{fp}}$, then the distortion can be estimate-neutral. The bottom arrow in Fig. 9b visualizes an example for this case where the distortion does not change the equality estimate. However, since the sender selected an FP codeword, the receiver determines a false estimate. The error is caused by the subset overlap and not by the distortion.

If the sender selects an FP codeword (that corresponds to a hash collision), then the distortion can accidentally *correct* the estimate $\hat{v}$ from an FP to a TN estimate. The bottom arrow in Fig. 9d shows an example of such a distortion. The distortion corrects the collision by changing the hash (that is the noisy-ID codeword) to a different value. This effect is not typical in linear block coding, so we will elaborate on the implications.

While the channel distortion can correct some FP codewords to TN codewords, the channel can also distort TN codewords to FP codewords. Without any additional knowledge of the channel or the code, there is no reason to assume that one effect outweighs the other. In other words, generally, the distortions from TN to FP estimates may compensate for the corrections from FP to TN estimates in a "zero-sum" effect. Yet, it is conceivable that there are noisy-ID codes that create an imbalance between the correction and distortion effects on the codewords for mismatching message pairs, such that overall (in the sum) the channel distortion reduces the FP error probability compared to a baseline without distortion. However, even if such a noisy-ID code could be found, the distortion would still cause FN errors for *matching* message

pairs. In that case, the correction via distortion would not be helpful for the overall performance. It is not meaningful to exclude matching message pairs from the analysis, because if the messages always match, then there is no communication problem, as both parties would know that their messages always mismatch. Hence, a noisy-ID code should always not only minimize the overlap of codeword subsets, but also mitigate the channel distortion (even though the channel distortion sometimes corrects FP codewords). Thus, we consider the potential correction effect of the channel on FP codewords merely an interesting subtlety of randomized noisy-ID codes, and not an avenue for engineering better noisy-ID codes.

4) Discussion of Error Causes: In summary, channel distortion can cause FN estimates for matching message pairs. For mismatching message pairs, the overlap in codeword subsets can cause FP estimates when the encoder randomly selects an FP codeword. This case corresponds to a hash collision in hash terminology. Generally, channel distortion does not increase the overall number of FP errors for mismatching message pairs. While the channel does increase the number of FP errors caused by estimate-changing distortions of TN codewords $\boldsymbol{x}_{\text{tn}}$, the channel also corrects FP errors (caused by selecting an FP codeword) via estimate-changing distortions of FP codewords $\boldsymbol{x}_{\text{fp}}$. Distortion therefore generally has a "zero-sum" effect on the number of FP errors. It is solely the overlap of codeword subsets that determines the number of FP errors. A good noisy-ID code allows for only a small number of FN errors by limiting the channel effects to estimate-neutral distortions, and for only a small number of FP errors by limiting the overlaps of codeword subsets. Distortion errors (TN errors) are associated with matching message pairs, and overlap errors (FP errors) are associated with mismatching message pairs.

E. Error Probabilities

A suitable noisy-ID code limits the number of FN and FP errors by limiting the corresponding error *probabilities*. The FN error probability $p_{\text{fn}}(\boldsymbol{a})$ of each message $\boldsymbol{a} \in \mathcal{U}$ is characterized by the channel $\boldsymbol{W}$, the encoding subset $\mathcal{E}_{\boldsymbol{a}}$, and the verifier subset $\mathcal{V}_{\boldsymbol{a}}$ (because $\boldsymbol{b} = \boldsymbol{a}$ for matching message pairs). Each message $\boldsymbol{a} \in \mathcal{U}$ is associated with a distinct FN error probability $p_{\text{fn}}(\boldsymbol{a})$. The upper bound λ_{fn} on the FN error probabilities $p_{\text{fn}}(\boldsymbol{a})\ \forall \boldsymbol{a} \in \mathcal{U}$ limits the worst case FN error probability. Irrespective of the selected message $\boldsymbol{a} \in \mathcal{U}$, the FN error probability $p_{\text{fn}}(\boldsymbol{a})$ does not exceed the bound λ_{fn}. The bound λ_{fn} is an important metric for proofs and formalizations of noisy-ID codes. In the literature, the bound λ_{fn} is typically referred to as λ_1 since the FN errors are also called type I (type 1) errors.

Analogously, the FP error probability $p_{\text{fp}}(\boldsymbol{a}, \boldsymbol{b})$ is also characterized by the channel $\boldsymbol{W}$, the encoding subset $\mathcal{E}_{\boldsymbol{a}}$, and (in contrast to the FN error probability) by the verifier subset $\mathcal{V}_{\boldsymbol{b}}$ of a message $\boldsymbol{b} \neq \boldsymbol{a}$. Each *pair* $(\boldsymbol{a}, \boldsymbol{b})$ of messages is associated with a distinct FP error probability $p_{\text{fp}}(\boldsymbol{a}, \boldsymbol{b})$. The FP error probability $p_{\text{fp}}(\boldsymbol{a}, \boldsymbol{b})$ of a message pair is mainly caused by and is proportional to the size $|\mathcal{E}_{\boldsymbol{a}} \cap \mathcal{V}_{\boldsymbol{b}}|$ of the overlap of the encoding subset $\mathcal{E}_{\boldsymbol{a}}$ with the verifier subset $\mathcal{V}_{\boldsymbol{b}}$, i.e., a large overlap corresponds to a high FP error probability $p_{\text{fp}}(\boldsymbol{a}, \boldsymbol{b})$. The upper bound λ_{fp} on the FP error probabilities $p_{\text{fp}}(\boldsymbol{a}, \boldsymbol{b})\ \forall \boldsymbol{a} \neq \boldsymbol{b}; \boldsymbol{a}, \boldsymbol{b} \in \mathcal{U}$ limits the worst-case FP error probability of all pairs of mismatching messages. In the literature, the bound λ_{fp} is typically referred to as λ_2 since the FP errors are also called type II (type 2) errors.

Note that the FP error probability is defined strictly for *pairs* of messages. If the receiver verifies the received codeword $\boldsymbol{y}$ for more than one message $\boldsymbol{b}$, then the FP error probability increases because it sums up over all verified messages. The verifier preimage of each codeword $\boldsymbol{y}$ holds a large number of messages, i.e., each codeword $\boldsymbol{y}$ is an element of the verifier subsets $\mathcal{V}_{\boldsymbol{b}}$ of many messages $\boldsymbol{b}$, cf. Fig. 8. Therefore, a single codeword $\boldsymbol{y}$ can cause a large number of FP estimates when the receiver verifies the received codeword with the verifier subsets $\mathcal{V}_{\boldsymbol{b}}$ of multiple messages. Verifying the received codeword $\boldsymbol{y}$ for more than a single message $\boldsymbol{b}$ is outside the scope of the identification problem. Rather, verifying the codeword for several messages is a related communication problem, cf. Section II-F. The main subject of this tutorial is the *pairwise* identification of messages.

F. Formal Definition

To conclude this section, we reproduce and explain the formal definition of randomized noisy-ID codes as originally defined in [1]. Our notation for this definition deviates slightly from the original definition for clarity and consistency with the notation of this tutorial.

Definition 1. A (randomized) noisy-identification (ID) code $(n, N, \lambda_{\text{fn}}, \lambda_{\text{fp}})$ is a family

$$\{(Q_{\boldsymbol{u}}, \mathcal{V}_{\boldsymbol{u}}) \,|\, \boldsymbol{u} \in \mathcal{U}\}$$

of pairs with

$$\mathcal{U} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N\},\ n = |\boldsymbol{W}| = |\boldsymbol{x}|,$$

and $Q_{\boldsymbol{u}}$ a probability distribution on $\mathcal{X}$ for all $\boldsymbol{u} \in \mathcal{U}$, and

$$\mathcal{V}_{\boldsymbol{u}} \subset \mathcal{Y}\ \forall \boldsymbol{u} \in \mathcal{U}, \tag{21}$$

and with errors of the false-negative (respective false-positive) kind satisfying

$$\sum_{\boldsymbol{x} \in \mathcal{X}} Q_{\boldsymbol{a}}(\boldsymbol{x})\ \boldsymbol{W}(\mathcal{V}_{\boldsymbol{a}}|\boldsymbol{x})\ \geq\ 1 - \lambda_{\text{fn}}, \tag{22}$$

and

$$\sum_{\boldsymbol{x} \in \mathcal{X}} Q_{\boldsymbol{a}}(\boldsymbol{x})\ \boldsymbol{W}(\mathcal{V}_{\boldsymbol{b}}|\boldsymbol{x})\ \leq\ \lambda_{\text{fp}}, \tag{23}$$

for all $\boldsymbol{a} \in \mathcal{U}, \boldsymbol{b} \in \mathcal{U}$ with $\boldsymbol{b} \neq \boldsymbol{a}$.

In less formal terms, Definition 1 states that a randomized noisy-ID code is characterized by the size n of the transmitted codeword $\boldsymbol{x}$, the number $N = |\mathcal{U}|$ of supported messages, and the two error probability bounds λ_{fn} and λ_{fp}. Pairs $(Q, \mathcal{V})$ constitute the overall code, whereby each pair consists of an encoding probability distribution Q, and a verifying codeword subset $\mathcal{V}$, with one such pair $(Q_{\boldsymbol{u}}, \mathcal{V}_{\boldsymbol{u}})$ for every supported message $\boldsymbol{u}$. Every encoding probability distribution $Q_{\boldsymbol{u}}$ is defined on the set $\mathcal{X}$ of encoder codewords (and the support

of the distribution $Q_{\boldsymbol{u}}$ corresponds to the encoding subset $\mathcal{E}_{\boldsymbol{u}}$, cf. Section IV-B1). Note that the codeword $\boldsymbol{x}$ is selected from the set $\mathcal{X}$ and consists of n symbols, i.e., $|\boldsymbol{x}| = n$. If the symbols are q-ary, then the cardinality of the set is $|\mathcal{X}| = q^n$, as Section II-D explained. The channel $\boldsymbol{W}$ is of size $|\boldsymbol{W}| = n$ because the channel size and the codeword size n match. Next to the encoding probability distribution $Q_{\boldsymbol{u}}$, each supported message $\boldsymbol{u}$ is also associated with a (verification) set $\mathcal{V}_{\boldsymbol{u}}$ of codewords that is a subset of the set $\mathcal{Y}$ of codewords at the receiver. For this tutorial, we generally consider $\mathcal{Y} = \mathcal{X}$ to facilitate the explanations.

Finally, there are two requirements on the error probabilities. In Eq. (22), the sender selects a codeword $\boldsymbol{x}$ with a probability $Q_{\boldsymbol{a}}(\boldsymbol{x})$ associated with the message $\boldsymbol{a}$. The channel $\boldsymbol{W}$, with some probability, maps the codeword $\boldsymbol{x}$ to (a codeword $\boldsymbol{y}$ that belongs to) the verifying codeword subset $\mathcal{V}_{\boldsymbol{a}}$ of the same message $\boldsymbol{a}$. The overall probability that the encoder selects a codeword $\boldsymbol{x}$ (that represents message $\boldsymbol{a}$) that the channel maps to the verifying codeword subset $\mathcal{V}_{\boldsymbol{a}}$ of the same message $\boldsymbol{a}$ is larger than $1 - \lambda_{\text{fn}}$, i.e., ideally close to 1, for all messages $\boldsymbol{a} \in \mathcal{U}$. In other words, for matching message pairs, the noisy-ID code should ensure that the selected codeword $\boldsymbol{x}_{\boldsymbol{a}}$ falls into the corresponding verification subset $\mathcal{V}_{\boldsymbol{a}}$ with high probability despite the distortion of the channel $\boldsymbol{W}$.

Analogous to Eq. (22), Eq. (23) requires that the overall probability of the encoder selecting a codeword $\boldsymbol{x}$ (that represents message $\boldsymbol{a}$) that the channel maps into the verifying codeword subset $\mathcal{V}_{\boldsymbol{b}}$ of a single, differing message $\boldsymbol{b} \neq \boldsymbol{a}$ to be smaller than λ_{fp}, i.e., ideally close to 0, for all messages $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{U},\ \boldsymbol{a} \neq \boldsymbol{b}$. In other words, for mismatching messages $\boldsymbol{a} \neq \boldsymbol{b}$, the noisy-ID code should ensure that after channel distortion, the selected codeword $\boldsymbol{x}_{\boldsymbol{a}}$ falls into the verification subset of a different message $\mathcal{V}_{\boldsymbol{b}}$ only with small probability. That is, the noisy-ID code should avoid that the receiver concludes an FP estimate $\hat{v}_{\text{fp}}$.

G. Noisy-ID Capacity

Ahlswede and Dueck [1] proved that the noisy-ID capacity of a DMC $\boldsymbol{W}$ coincides with the transmission capacity C_{DMC} that Section III-F explained. Thereby, the transmission capacity C_{DMC} is an upper limit on the rate $R_{\text{noisy-id}}$ that a noisy-ID code can achieve. While the capacity coincides for transmission and noisy identification, the scaling of the corresponding code families does not. As Section III-B explained, the rate R_{fec}, cf. Eq. (8) of a linear block code can scale exponentially in the codeword size (block length) n. In this exponential scaling, a linear block code can asymptotically, i.e., for $n \to \infty$, guarantee virtually error-free transmission. Ahlswede and Dueck [1] proved that the noisy-ID code rate $R_{\text{noisy-id}}$, cf. Eq. (15), can scale *double-exponentially* as Section IV-B explained. Even in double-exponential scaling, a noisy-ID code can asymptotically, i.e., for $n \to \infty$, guarantee virtually error-free *identification via noisy channels*. A noisy-ID code with the following double-exponential rate can reliably address the ID problem:

$$R_{\text{noisy-id}} = \frac{\log_2(\log_2(N))}{n'} \leq C. \tag{24}$$

For the ID problem, noisy-ID codes can support an up to exponentially larger number N of messages than a linear block code for the same codeword length n.

The proof for this achievability includes only a few assumptions. All N encoding subsets, i.e., one set per message, have the same size, i.e., $|\mathcal{E}_1| = |\mathcal{E}_2| = \cdots = |\mathcal{E}_N|$, and the encoding probabilities $Q_{\boldsymbol{u}}$ are all equidistributions on their respective subset $\mathcal{E}_{\boldsymbol{u}}$.

In contrast to the transmission capacity, the ID capacity can be increased, e.g., by feedback and by access to common randomness. This offers the potential to make noisy-ID codes even more efficient for the ID problem. The ID capacity of different channels has been one of the dominant avenues of research on noisy-ID codes [103]–[108]. Channels other than the DMC are out of the scope of this tutorial.

Noisy-ID codes offer a tradeoff between the efficiency of the code, i.e., the codeword size n per message size k, and the reliability, i.e., the FP error probability bound λ_{fp}. It is possible to increase the number N of supported messages, but doing so increases the FP error probability. That is because in terms of the sphere packing problem, the spheres (codeword subsets for encoding and verification) of an increasing number N of messages are stacked into the same set $\mathcal{E}$ of codewords. The overlap ratio increases and the FP error probability bound λ_{fp} rises. Therefore, the noisy-ID code rate $R_{\text{noisy-id}}$ "competes" with the FP error probability bound λ_{fp}. The higher the rate $R_{\text{noisy-id}}$, i.e., the more efficient the noisy-ID code, the higher the achievable FP error probability bound λ_{fp}, i.e., the less reliable the noisy-ID code.

To describe achievable FP error probability bounds λ_{fp}, the literature uses *error exponents*. The error exponent E_{fp} is a different representation of the FP error probability bound λ_{fp}:

$$E_{\text{fp}} = \frac{-\log_2(\lambda_{\text{fp}})}{n'}. \tag{25}$$

The literature often refers to this error exponent as E_2 because FP errors are often referred to as type II errors. A high error exponent E_{fp} corresponds to a small FP error probability bound λ_{fp} and is therefore desirable. A noisy-ID code is characterized by the pair $(R_{\text{noisy-id}}, E_{\text{fp}})$ of rate and FP error exponent. This pair is subject to the ID capacity of the channel [1, Theorem 2]:

$$R_{\text{noisy-id}} + 2E_{\text{fp}} \leq C, \tag{26}$$

which can be reformulated to

$$\frac{\log_2(\log_2(N)) - 2\log_2(\lambda_{\text{fp}})}{n'} \leq C. \tag{27}$$

The bound that is the ID capacity C is generally not a tight bound because the bound additionally depends on the FN error probability bound λ_{fn} and its derived FN error exponent

$$E_{\text{fn}} = \frac{-\log_2(\lambda_{\text{fn}})}{n'}. \tag{28}$$

If the FN error probability bound λ_{fn} approaches zero, then the bound in Eqs. (26) and (27) becomes a tight bound. That is, for $\lambda_{\text{fn}} \to 0$, it is possible to find a noisy-ID code with a pair $(R_{\text{noisy-id}}, E_{\text{fp}})$ of rate and FP error exponent that achieves $R_{\text{noisy-id}} + 2E_{\text{fp}} = C$.

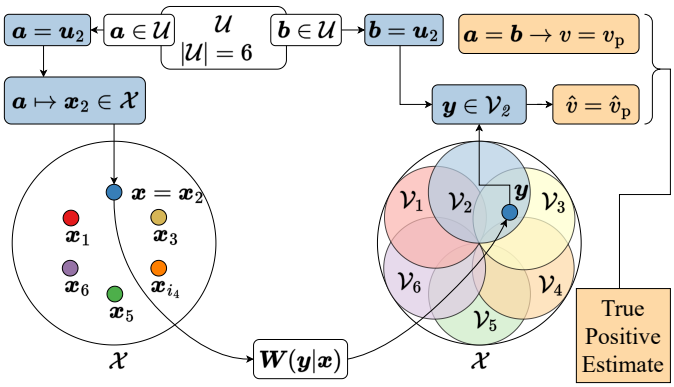


Fig. 10. Schematic view of deterministic noisy-ID encoding and verifying. In contrast to randomized noisy-ID codes as visualized in Fig. 8, the deterministic noisy-ID encoder maps the message $\boldsymbol{a}$ to a deterministic codeword and does not sample a random codeword from a codeword subset.

In less theoretical terms this means that a noisy-ID code that is very efficient (via a high rate $R_{\text{noisy-id}}$) and simultaneously reliable (via a low FP error probability bound λ_{fp}) requires an FN error probability bound λ_{fn} that approaches zero. As explained in Section IV-D4, the FN errors are associated with channel distortion. For $\lambda_{\text{fn}} \to 0$, the noisy-ID code is able to avoid virtually all errors caused by the channel distortion. Thereby, the noisy-ID code constructs a virtually noiseless (reliable) channel. To address the overall ID problem reliably, the noisy-ID code additionally has to ensure that the FP error probability bound λ_{fp} is low. The following Section V will explain that it is possible to construct noisy-ID codes based on this idea to treat the channel distortion and the FP error probability (caused by the random sampling of codewords from overlapping codeword subsets) separately.

H. Deterministic Noisy-ID Codes

Before Section V explains how to construct randomized noisy-ID codes by treating the channel distortion and the ID-specific encoding separately, this subsection explains noisy-ID codes that do not use randomization in the encoding. Such noisy-ID codes are called *deterministic* noisy-ID codes or noisy-ID codes without randomization. Deterministic noisy-ID codes are closely related to the randomized noisy-ID codes that this section explained.

Analogously to the block encoding procedure described in Section III-B, the sender encodes the message $\boldsymbol{a}$ into a noisy-ID codeword $\boldsymbol{x}$, according to a deterministic ID encoding function $f_{\text{enc, detid}}$, cf. Fig. 10:

$$f_{\text{enc, detid}} : \boldsymbol{a} \mapsto \boldsymbol{x} \quad \forall \boldsymbol{a} \in \mathcal{U}. \tag{29}$$

Because the receiver only needs to determine the equality estimate $\hat{v}$, in contrast to block codes, the codeword subsets $\mathcal{V}$ associated with each message $\boldsymbol{u}$ can overlap similar to the way they do for randomized noisy-ID codes, cf. Fig. 8. The verifier subsets $\mathcal{V}$ are centered around their constructing codebook codeword $\boldsymbol{a}$, i.e., in the example in Fig. 10, the verifier subset $\mathcal{V}_2$ associated with message $\boldsymbol{u}_2$ is centered around the codeword $\boldsymbol{x_2}$. Thereby, the deterministic noisy-ID code is able to overcome distortion by the channel $\boldsymbol{W}$.

In contrast to block decoding, the verification subsets $\mathcal{V}$ overlap, thereby loosening the sphere packing problem to a problem of fitting the "spheres" (verification subsets) with small overlap. Since the encoding is deterministic, the deterministic noisy-ID code can fit fewer verification subsets into the set $\mathcal{Y}$ (that equals $\mathcal{X}$ in the example in Fig. 10) than a randomized noisy-ID code can. For ID via a DMC, the scaling of deterministic noisy-ID codes matches the exponential scaling of linear block codes [27], i.e., there is no benefit in using deterministic noisy-ID codes in this setting. However, for Gaussian channels, deterministic noisy-ID codes can achieve code rates of a better scaling than the single exponential scaling of block code rates [93], cf. Eq. (8), but of a worse scaling than the double-exponential scaling of randomized noisy-ID codes, cf. Eq. (15). Specifically, the rate of deterministic noisy-ID codes for ID via a Gaussian channel can be [93]

$$R_{\text{detid}} = \frac{\log_2(N)}{n' \log_2(n')}. \tag{30}$$

Thereby, for Gaussian channels, deterministic noisy-ID codes can support a number N of messages up to logarithmic order higher than block codes:

$$N = 2^{n' R_{\text{id}} \log_2(n')} = n'^{n' R_{\text{id}}}, \tag{31}$$

in contrast to the number of messages supported by block codes, cf. Eq. (9), and in contrast to the number of messages supported by randomized noisy-ID codes, cf. Eq. (16). Reformulated for the size k of the messages, this yields for deterministic ID via Gaussian channels

$$k' = \log_2(N) = n' R_{\text{id}} \log_2(n'). \tag{32}$$

Somewhat misleadingly, the literature refers to some noisy-ID codes *with* randomization at the encoding as deterministic noisy-ID codes. The term "deterministic" only requires that the sender itself does not have access to *local* randomness. However, it is possible that the sender generates the randomness; for example, by an experiment via the noisy channel $\boldsymbol{W}$, and a noiseless feedback channel. In that case, the "deterministic" noisy-ID code can use *non-local* randomness for the encoding, whereby the randomization enables higher efficiency by allowing for higher achievable code rates R_{id}. This corresponds to increasing the ID capacity of the channel $\boldsymbol{W}$ for "deterministic" noisy-ID codes, cf. Section IV-G that explained that feedback and common randomness can increase the ID capacity for randomized noisy-ID codes. Thereby, information-theoretic studies on deterministic (noisy) identification with noiseless feedback have implications also for randomized noisy-ID codes, because in practice it is not always important how the sender acquired the randomness that is necessary for randomizing the encoding.

Deterministic noisy-ID codes (without any kind of randomization) have so far received limited attention from the practical ID research community, mainly because such codes have a smaller potential than randomized noisy-ID codes to outperform block codes. Recently, however, [24], [109] initiated also more practical research by proposing the first explicit construction for deterministic noisy-ID codes, whereby these

constructions are truly deterministic without any randomization at the encoder. Furthermore, deterministic noisy-ID codes can enable reliable communication even at negative dB values of signal-to-noise ratios, i.e., in extremely noisy environments. This is because the deterministic noisy-ID codes address the ID problem more efficiently than channel codes and thereby can achieve lower error probabilities even in resource-constrained scenarios with undesirable channel conditions.

Due to the community's focus on randomized noisy-ID codes, from hereon, the tutorial does not consider deterministic noisy-ID codes.

I. Summary

Noisy-ID codes address the ID problem via channels more efficiently than channel codes can. That is, noisy-ID codes are tailored to the ID problem in contrast to the goal-agnostic channel codes. The efficiency gain comes from a relaxation in the requirements on the verification ("decoding") codeword subsets. Linear block codes rely on disjoint codeword subsets in the decoder because they generally require an unambiguous mapping between the received codeword $\boldsymbol{y}$ and the message estimate $\hat{\boldsymbol{a}}$. However, in ID, the receiver does not need to estimate the message $\boldsymbol{a}$ of the sender (there are N different messages) but only the equality v of the receiver's message $\boldsymbol{b}$ to the sender's message $\boldsymbol{a}$ (i.e, the binary estimate for either match or a mismatch). If the received codeword $\boldsymbol{y}$ is close enough to the codebook codeword associated with the receiver's message $\boldsymbol{b}$, then the received codeword $\boldsymbol{y}$ falls into the corresponding verification subset $\mathcal{V}_{\boldsymbol{b}}$. The receiver estimates matching messages between the sender and the receiver in this case. Since the receiver only determines this binary equality estimate $\hat{v}$, the verification ("decoding") codeword subsets of different messages $\boldsymbol{b}$ can overlap.

Such a noisy-ID code is referred to as a deterministic noisy-ID code, cf. Section IV-H. By adding randomization to the encoding process, the ID problem can be addressed even more efficiently. Each message $\boldsymbol{a}$ does not correspond to a single codeword in the codebook, but to a set $\mathcal{E}_{\boldsymbol{a}}$ of codewords. The encoder randomly selects one of the codewords from the set $\mathcal{E}_{\boldsymbol{a}}$. Differing messages share some of their codewords, i.e., their encoding codeword sets $\mathcal{E}$ overlap. Such codes with randomization at the encoder are referred to as randomized noisy-ID codes. These codes are the focus of this tutorial as they can be more powerful than deterministic noisy-ID codes. One method to construct randomized noisy-ID codes lies in concatenating a linear block code with a second code (a *noiseless-ID code* that enables the ID-specific efficiency gain) as Section V will explain. Other methods are conceivable but—to the best of our knowledge—have not yet been proposed. Thereby, the state of the art in noisy-ID coding lies in concatenating a noiseless-ID code with a linear block code to form a concatenated randomized noisy-ID code.

V. Separation Principle

To address the ID problem reliably and efficiently, noisy-ID codes have to i) overcome the distortion of the channel $\boldsymbol{W}$, and ii) create codeword subsets that purposefully overlap slightly; thereby, striking a balance between efficiency gain and reliability loss, cf. Section IV-D4.

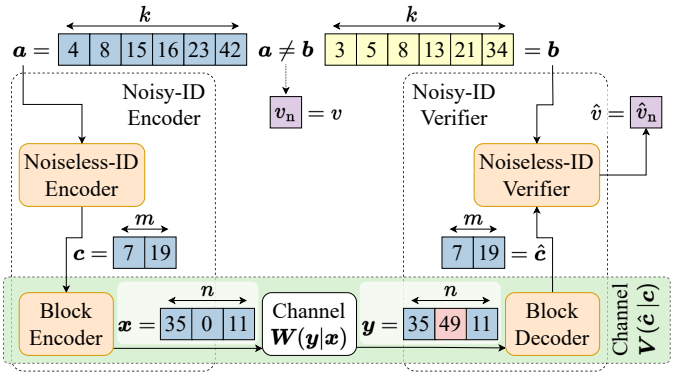


Fig. 11. Example for the encoding and verification of a concatenated noisy-ID code.

A. Joint Coding

In principle, it is conceivable to create a randomized noisy-ID code that *jointly* addresses both causes for errors: those caused by the distortion, and those caused by the overlap of codeword subsets that enables efficient ID. Joint randomized noisy-ID coding is also implied in Section IV in general, and in Fig. 8 in particular. However, to the best of our knowledge, a joint randomized noisy-ID coding approach has not yet been investigated by the research community. Instead, all research on randomized noisy-ID codes considers addressing the two causes for errors separately. Specifically, in the original study [1] that initiated research on message identification, the proof that randomized noisy-ID codes have up to exponential gains over block codes when addressing the ID problem considers a separate approach for each cause for errors (separation principle), as Section IV-G explained. Additionally, the first explicit noisy-ID code construction [21] (from which all other noisy-ID codes have been derived to date) also addresses the two causes for errors separately. Since no joint randomized noisy-ID codes have been investigated yet, for the remainder of this section, we focus on randomized noisy-ID codes that address the two error causes separately.

B. Separate Coding

The separation principle reformulates the communication problem of identification via *noisy* channels into two subproblems: i) forming a virtually distortion-free (noiseless) channel $\boldsymbol{V}$, and ii) identification via *noiseless* channels. Section III explained that linear block codes are able to form a noiseless channel. Codes that address ID via noiseless channels are called *noiseless-ID codes* (in contrast to noisy-ID codes that address ID via *noisy* channels). Concatenating a (deterministic) linear block code with a randomized noiseless-ID code can yield an overall randomized noisy-ID code. The (noiseless-)ID code does not need to address the distortion of the channel $\boldsymbol{W}$ and focuses instead on all remaining aspects that are required to construct an efficient noisy-ID code. Specifically, randomized noiseless-ID codes encode the

message $\boldsymbol{a}$ to a random noiseless-ID codeword $\boldsymbol{c}$ that is smaller than the original message $\boldsymbol{a}$, thereby enabling more efficient ID. Additionally, a randomized noiseless-ID code guarantees that the error probability p_{fp} (that the loss of information in the noiseless-ID encoding incurs) is small and limited, cf. Section IV-D4.

Because the noiseless-ID encoder maps the message $\boldsymbol{a}$ to a shorter codeword $\boldsymbol{c}$, information is lost in the encoding. In other words, the codeword $\boldsymbol{c}$ is not unique to each message. Rather, the codeword $\boldsymbol{c}$ can be understood as an imperfect "fingerprint" of the message whereby the codeword does not differ from the "fingerprints" of all other messages. This is acceptable as long as it is highly unlikely that two different messages are encoded to the same codeword $\boldsymbol{c}$. The information contained in the codeword $\boldsymbol{c}$ suffices to *identify* the sender's message $\boldsymbol{a}$ with high probability, i.e., reliably. We leave detailed explanations of the specific properties of such noiseless-ID codes to Section VI. For this section, we consider to have access to a reliable, efficient noiseless-ID code and focus on its concatenation with a block code to create a noisy-ID code.

Fig. 11 visualizes how a concatenated noisy-ID code addresses the noisy-ID problem. An ID encoder maps the message $\boldsymbol{a}$ of the sender to a (random) noiseless-ID codeword $\boldsymbol{c}$ that is significantly smaller than the message $\boldsymbol{a}$, i.e., the noiseless-ID codeword size m is significantly smaller than the message size k. In other words, the noiseless-ID code "hashes" the message $\boldsymbol{a}$ to a random noiseless-ID codeword $\boldsymbol{c}$ ("hash"). In a second step, the block encoder adds redundancy to the random noiseless-ID codeword $\boldsymbol{c}$ to make the noiseless-ID codeword resilient against the distortion of the channel $\boldsymbol{W}$. Thereby, the block encoder creates the overall random noisy-ID codeword $\boldsymbol{x}$ of length n, whereby typically $m < n < k$. The concatenation of a linear block code with a randomized noiseless-ID code forms a concatenated noisy-ID code for reliable identification and is not to be confused with the concatenation of two linear block codes to form a concatenated linear block code for reliable message transmission, cf. Section VII-F.

The concatenation of the ID encoder with the block encoder forms a noisy-ID encoder that maps the sender's message $\boldsymbol{a}$ to a random noisy-ID codeword $\boldsymbol{x}$ as Section IV explained. The noisy-ID codeword $\boldsymbol{x}$ is the deterministic block codeword $\boldsymbol{x}$ that encodes the random noiseless-ID codeword $\boldsymbol{c}$ that is a "hash" of the message $\boldsymbol{a}$. Thereby, the separation principle enables the construction of a randomized noisy-ID encoder by the concatenation of a noiseless-ID encoder and a block encoder. Analogously, the concatenation of the block decoder with the ID verifier forms a noisy-ID verifier. The input to the noisy-ID verifier is the block codeword $\boldsymbol{y}$ (that corresponds to the noisy-ID codeword $\boldsymbol{y}$ explained in Section IV). From the block codeword $\boldsymbol{y}$ and the message $\boldsymbol{b}$ of the receiver, the noisy-ID verifier concludes the equality estimate $\hat{v}$.

Fundamentally, the separation approach reproduces the notion from Claude Shannon's Theory of Communication [2] to reformulate arbitrary communication problems into the "technical" problem of reliable message transmission as the linear block code creates the noiseless channel $\boldsymbol{V}$ for the noiseless-ID code. For the purpose of creating a concatenated randomized noisy-ID code, it is therefore possible to rely partially on the well-understood problem of creating block codes that are capable of efficiently forming a virtually distortion-free ("noiseless") channel $\boldsymbol{V}$ from a DMC $\boldsymbol{W}$. We consider forming a virtually noiseless channel $\boldsymbol{V}$ with block codes an orthogonal coding problem to the problem of finding a suitable noiseless-ID code. Specifically, using a suitable block code and a suitable noiseless-ID code, it is possible to construct a concatenated noisy-ID code that is *optimal for ID* [1], i.e., the concatenated noisy-ID code rate $R_{\text{noisy-id}}$, cf. Eq. (15), can reach the ID capacity of the channel, cf. Section IV-G. For details on employing block codes to minimize transmission errors over noisy channels, we refer to [7]–[14]. Since the block code addresses the issue of mitigating the distortion, this leaves finding a noiseless-ID code that addresses ID via noiseless channels.

With respect to Fig. 3, the sender encodes its message $\boldsymbol{a}$ for the verification step at the receiver. Thereby, concatenated noisy-ID codes use conventional block coding, and add a noiseless-ID-specific encoding step between the source and the block (channel) coding. Therefore, it typically does not make sense to compare noiseless-ID codes directly with channel codes, such as block codes, cf. Section I.

It is possible to frame noiseless-ID codes as noisy-ID codes with an FN error probability bound $\lambda_{\text{fn}} = 0$ [1]. In other words, given that the FN errors that are associated with the channel distortion do not occur, a noiseless-ID code suffices to address ID via channels because an FN error probability bound $\lambda_{\text{fn}} = 0$ implies a noiseless channel $\boldsymbol{V}$. A block code can mitigate the channel distortion that is associated with FN errors, cf. Section IV-D4. While the block code aims to mitigate all FN errors, the noiseless-ID code introduces a limited FP error probability that is associated with the ID-specific encoding of noisy-ID codes.

If the block code is not able to fully mitigate the distortion of the channel $\boldsymbol{W}$, then the concatenated noisy-ID code will yield distortion errors as explained in Section IV-D and visualized in Fig. 9. However, for the sake of separation, a noiseless-ID code (that in concatenation with a block code forms an overall noisy-ID code) assumes that the block code addresses the distortion and forms a perfectly noiseless channel $\boldsymbol{V}$.

C. Block Codes in Concatenated Noisy-ID Codes

From the perspective of the noiseless-ID code, the serial arrangement of the block encoder, the channel $\boldsymbol{W}$, and the block decoder form a virtual channel $\boldsymbol{V}$, cf. Fig. 11. The ID encoder and ID verifier communicate over the virtual channel $\boldsymbol{V}$ and assume that the channel does not impose any distortion on the transmitted noiseless-ID codeword $\boldsymbol{c}$. Therefore, the block code should ideally fully mitigate the effects of distortion in a channel $\boldsymbol{W}$. In contrast to the block encoding function Eq. (7) that Section III explained, the block code does not encode the message $\boldsymbol{a}$ but the noiseless-ID codeword $\boldsymbol{c}$:

$$f_{\text{fec}} : \boldsymbol{c} \mapsto \boldsymbol{x} \in \mathcal{X} \quad \forall \boldsymbol{c} \in \mathcal{C}. \tag{33}$$

The size n of the block codeword $\boldsymbol{x}$ (that matches the size of the noisy channel $\boldsymbol{W}$) is unchanged from the explanation of message transmission in Section III. However, the size of the input to the block code differs. The block code encodes the noiseless-ID codeword $\boldsymbol{c}$ of size $m < k$. Thereby, the block code rate is $R_{\text{fec}} = m/n$ in this case. The transmission capacity C of the noisy channel $\boldsymbol{W}$ limits the block code rate $R_{\text{fec}} = m/n$, cf. Section III-F.

From the distorted block codeword $\boldsymbol{y}$ (that differs from the sent block codeword $\boldsymbol{x}$ in one symbol due the channel distortion in the example in Fig. 11), the block decoder determines an estimate $\hat{\boldsymbol{c}}$ of the noiseless-ID codeword $\boldsymbol{c}$ that the ID encoder randomly selected. Thereby, the noiseless channel $\boldsymbol{V}$ (that the block code forms) maps the noiseless-ID codeword $\boldsymbol{c}$ to an estimate $\hat{\boldsymbol{c}}$ of the noiseless-ID codeword $\boldsymbol{c}$, i.e., we write noiseless channel $\boldsymbol{V}(\hat{\boldsymbol{c}}|\boldsymbol{c})$. The noiseless channel $\boldsymbol{V}$ has size m. Based on the noiseless-ID codeword estimate $\hat{\boldsymbol{c}}$ and the receiver's message $\boldsymbol{b}$, the ID verifier determines the equality estimate $\hat{v}$. In the example in Fig. 11, the equality estimate is correct because the block code provides a correct noiseless-ID codeword estimate $\hat{\boldsymbol{c}}$ to the ID verifier despite the distorted symbol in the receiver block codeword $\boldsymbol{y}$.

D. Summary

To create a (randomized) noisy-ID code, it is possible to separate the code construction into two subcodes. To address the distortion of the channel $\boldsymbol{W}$, a channel code, such as a linear block code can be used. The channel code adds redundancy to the message and can recover an estimate of the message reliably, thereby creating a virtual noiseless channel $\boldsymbol{V}$. To achieve the up to exponential efficiency gains of noisy-ID codes over channel codes in the ID problem, a *noiseless-ID code* performs a "compression" of the message that reduces the size of the message significantly to a small noiseless-ID codeword (that can be considered a "hash"). Thereby, the noiseless-ID encoder maps the message $\boldsymbol{a}$ of size k to a smaller "hash" $\boldsymbol{c}$ of size m. The linear block code encodes not the message itself but the "hash" $\boldsymbol{c}$. Hence, the redundant overhead that the linear block code provides is relative to the "hash" $\boldsymbol{c}$ and not relative to the original message $\boldsymbol{a}$. Therefore, the size n of the block (channel) codeword $\boldsymbol{x}$ can be smaller than the size k of the message $\boldsymbol{a}$, cf. Fig. 11.

In other words, utilizing a channel code can transform the problem of ID via noisy channels to the problem of ID via noiseless channels. Noiseless-ID codes address the problem of ID via noiseless channels. We leave the impact of the noise on the performance of the overall concatenated noisy-ID code to Section IX and explain noiseless-ID codes in the following section.

VI. Noiseless-ID Codes

A. Principle

Noiseless-ID codes can enable *efficient, reliable* identification via noiseless channels. Specifically, noiseless-ID codes encode messages to improve ID efficiency while still guaranteeing low error probabilities to ensure reliable ID. In contrast to the problem of ID via noisy channels (as explained in Section II-A), noiseless-ID codes address ID via noiseless channels. The channel that connects the sender and the receiver is noiseless and does not distort the transmitted noiseless-ID codeword. Such a noiseless channel can be formed by a linear block code, cf. Section III-B, and thereby transform ID via noisy channels into ID via noiseless channels. For an overview of noiseless-ID code notation, refer to Table VIII.

TABLE VIII
Summary of notations for randomized noiseless-ID codes.

$\mathcal{U}$	Set of messages
$\boldsymbol{u}, \boldsymbol{a}, \boldsymbol{b}$	Message (arbitrary, sender's, and receiver's) in $\mathcal{U}$
$\mathcal{C}$	Set of noiseless-ID codewords
$\mathcal{C}_{\boldsymbol{u}}$	Noiseless-ID codeword subset of message $\boldsymbol{u}$, subset of $\mathcal{C}$
$\boldsymbol{c}$	Noiseless-ID codeword ("hash") in $\mathcal{C}$
N	Size of message set $\mathcal{U}$, $N = \lvert\mathcal{U}\rvert$
S	Size of noiseless-ID codeword set $\mathcal{C}$, $S = \lvert\mathcal{C}\rvert$
M	Size of N noiseless-ID codeword subsets $\mathcal{C}_{\boldsymbol{u}}$, $M = \lvert\mathcal{C}_{\boldsymbol{u}}\rvert$
k	Size of message $\boldsymbol{u}$ in q-ary symbols
m	Size of noiseless-ID codeword $\boldsymbol{c}$ in q-ary symbols
$K(\boldsymbol{a}, \boldsymbol{b})$	Overlap size $\lvert\mathcal{C}_{\boldsymbol{a}} \cap \mathcal{C}_{\boldsymbol{b}}\rvert$ of subset pair
κ	Upper bound on overlap sizes K

The inputs to the noiseless-ID problem are the message $\boldsymbol{a} \in \mathcal{U}$ of the sender and the message $\boldsymbol{b} \in \mathcal{U}$ of the receiver, whereby both messages are from the set $\mathcal{U}$ of messages. The output of noiseless-ID problem is the equality estimate $\hat{v}$. The noiseless channel $\boldsymbol{V}$ transmits the noiseless-ID codeword $\boldsymbol{c} \in \mathcal{C}$ from the sender to the receiver. Messages $\boldsymbol{u}, \boldsymbol{a}, \boldsymbol{b}$ are represented in $\mathbb{F}_q^k$, and noiseless-ID codewords in $\mathbb{F}_q^m$. Therefore, each message consists of k symbols, and each noiseless-ID codeword $\boldsymbol{c}$ consists of m symbols in base q.

With a suitable code, if the noiseless-ID codeword size m equals or exceeds the message size k, then the receiver can always recover the message $\boldsymbol{a}$ from the codeword $\boldsymbol{c}$ and thereby always correctly determine the equality estimate $\hat{v}$. An *efficient* noiseless-ID code "hashes" the message $\boldsymbol{a}$ to a noiseless-ID codeword $\boldsymbol{c}$ that is smaller than the message $\boldsymbol{a}$, i.e., $m < k$. Because the "hashing" removes information from the message, it is impossible to avoid that different messages collide in the same noiseless-ID codeword. With the pigeonhole principle, if the number q^k of messages exceeds the number q^m of noiseless-ID codewords, then several different messages must map to the same noiseless-ID codeword. A *reliable* noiseless-ID code guarantees that the collision probability between messages is small.

Finding a noiseless-ID code that is reliable and efficient corresponds to striking a balance between how much data the "hashing" removes from the message $\boldsymbol{a}$ (efficiency gain) and how small the incurred collision probability is (reliability loss). For the noiseless channel, a noiseless-ID code can "hash" the message $\boldsymbol{a}$ to a size m that is only a logarithm of the size k of the message and still maintain vanishing collision probability if the noiseless-ID codeword size m (that is also referred to as the block length) grows toward infinity [1]. At finite noiseless-ID codeword sizes m, the collision probability can only be zero if the noiseless-ID codeword is not smaller than the message, i.e., for $k \geq m$. However, it is possible to find noiseless-ID codes that guarantee a *low* collision probability

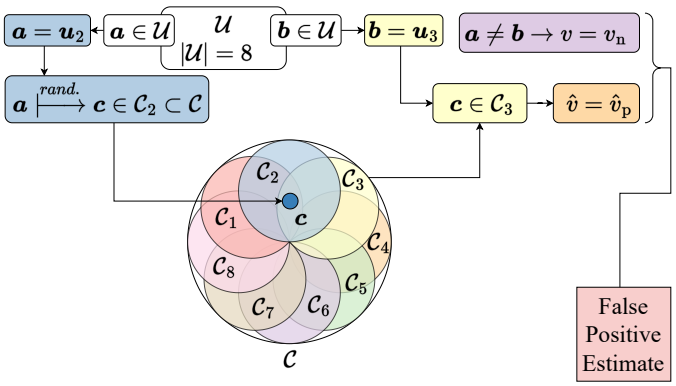


Fig. 12. Schematic view of noiseless-ID encoding and verification. In any randomized noiseless-ID coding scheme, the sender maps the message $\boldsymbol{a}$ to a randomly selected "hash" $\boldsymbol{c}$ from a set $\mathcal{C}_{\boldsymbol{a}}$ of "hashes" that can collide with several "hashes" (that constitute the subset $\mathcal{C}_{\boldsymbol{b}}$) that are associated with a differing message $\boldsymbol{b} \neq \boldsymbol{a}$ of the receiver. In this example, the sender selects a "hash" that collides with a "hash" of the receiver's message, thereby causing a false equality estimation at the receiver. Since the channel that connects the sender and receiver is noiseless, the encoding and verification subsets are identical, in contrast to noisy-ID codes, cf. Fig. 8.

at finite block lengths m. We leave further details about the limits on efficiency and reliability to Section VI-D.

Hashing the message $\boldsymbol{a}$ to a size m that is only a logarithm of the original message size k corresponds to an exponential efficiency gain and is only possible if the noiseless-ID encoding uses randomization. That is, each message $\boldsymbol{u}$ is associated with a set $\mathcal{C}_{\boldsymbol{u}}$ of noiseless-ID codewords $\boldsymbol{c}$ and the noiseless ID-encoding randomly selects a noiseless-ID codeword $\boldsymbol{c} \in \mathcal{C}_{\boldsymbol{u}}$. In hashing terminology, this closely resembles universal hashing as Section VI-E will explain.

B. Randomized Encoding and Verification

An encoder-verifier pair constitutes a noiseless-ID code. That is, the encoder defines how to encode the message into a noiseless-ID codeword $\boldsymbol{c}$. Whereas, the verifier defines how to use the received noiseless-ID codewords to determine the estimate $\hat{v}$ of the equality v between the messages $\boldsymbol{a}$ and $\boldsymbol{b}$ of the sender and the receiver.

Each message $\boldsymbol{u}$ is associated with a single subset $\mathcal{C}_{\boldsymbol{u}}$ of noiseless-ID codewords (in contrast the pair of one encoding subset $\mathcal{E}_{\boldsymbol{u}}$ and one verification subset $\mathcal{V}_{\boldsymbol{u}}$ for each message $\boldsymbol{u}$ in noisy-ID codes, cf. Section IV). In other words, the noiseless-ID encoding subset $\mathcal{C}_{\boldsymbol{u}}$ and the noiseless-ID verification subset $\mathcal{C}_{\boldsymbol{u}}$ associated with each message $\boldsymbol{u} \in \mathcal{U}$ are identical [1]. It is conceivable to create a noiseless-ID code that does not use identical subsets for encoding and verification. However, such a mismatch would (undesirably) introduce FN errors without any advantage that we currently know of. Hence, this tutorial focuses on using identical subsets.

Figure 12 visualizes the encoding and verification in a schematic example. Similar to the noisy-ID encoding Eq. (14), noiseless-ID encoding can be considered a two-step process. For the first step, the sender deterministically maps the message $\boldsymbol{u}$ to its associated noiseless-ID codeword subset $\mathcal{C}_{\boldsymbol{u}}$ that is a subset of the set $\mathcal{C}$ of all possible noiseless-ID codewords. In the second step, the encoder selects a random noiseless-ID codeword $\boldsymbol{c}$ from the set $\mathcal{C}_{\boldsymbol{a}}$ of noiseless-ID codewords associated with the message $\boldsymbol{a}$ of the sender. Thereby, the randomized noiseless-ID encoding function for an arbitrary message $\boldsymbol{u}$ is

$$f_\mathrm{id} : \boldsymbol{a} \overset{\text{rand.}}{\longmapsto} \boldsymbol{c} \in \mathcal{C}_{\boldsymbol{u}} \subset \mathcal{C} \quad \forall \boldsymbol{u} \in \mathcal{U}. \tag{34}$$

Each subset $\mathcal{C}_{\boldsymbol{u}}$ has the same size $M = |\mathcal{C}_{\boldsymbol{u}}|$, whereby the subset size M is smaller than the size S of the set $\mathcal{C}$ of all noiseless-ID codewords.

For verification, the receiver determines whether the noiseless-ID codeword $\boldsymbol{c}$ is an element of the noiseless-ID codeword set $\mathcal{C}_{\boldsymbol{b}}$ associated with the receiver's message $\boldsymbol{b}$. In contrast to verification via noisy-ID codes, cf. Section IV-C, noiseless-ID codes do not specify a specific verification subset. Rather, there is a single noiseless-ID codeword subset $\mathcal{C}_{\boldsymbol{u}}$ that is associated with message $\boldsymbol{u}$. The noiseless-ID codeword subset $\mathcal{C}_{\boldsymbol{u}}$ replaces the two subsets of noisy-ID codes. In contrast to the noisy-ID verification rule in Eq. (19), the receiver of the noiseless-ID codeword $\boldsymbol{c}$ determines its equality estimate $\hat{v}$ as:

$$\hat{v} = \begin{cases} \hat{v}_\mathrm{p} \text{ if } \boldsymbol{c} \in \mathcal{C}_{\boldsymbol{b}}, \\ \hat{v}_\mathrm{n} \text{ if } \boldsymbol{c} \notin \mathcal{C}_{\boldsymbol{b}}. \end{cases} \tag{35}$$

C. Overlap of Noiseless-ID Codeword Subsets

Noiseless-ID codes are subject to FP errors but not to FN errors. That is because each message $\boldsymbol{u}$ is associated with a single noiseless-ID codeword subset $\mathcal{C}_{\boldsymbol{u}}$ for both the encoding and verification step. If the message pair $(\boldsymbol{a}, \boldsymbol{b})$ matches, i.e., $\boldsymbol{a} = \boldsymbol{b}$, then the equality estimate $\hat{v}$ is always correct and no FN errors occur. For message pairs $(\boldsymbol{a}, \boldsymbol{b})$ of differing messages $\boldsymbol{a} \neq \boldsymbol{b}$, the "hashing" of the message $\boldsymbol{a}$ can result in FP errors $\hat{v}_\mathrm{fp}$ because the codeword subsets $\mathcal{C}_{\boldsymbol{u}}$ overlap.

Specifically, there are $K(\boldsymbol{a}, \boldsymbol{b})$ noiseless-ID codewords that the subset $\mathcal{C}_{\boldsymbol{a}}$ shares with subset $\mathcal{C}_{\boldsymbol{b}}$:

$$K(\boldsymbol{a}, \boldsymbol{b}) = |\mathcal{C}_{\boldsymbol{a}} \cap \mathcal{C}_{\boldsymbol{b}}|, \tag{36}$$

whereby the minimum number of shared codewords is 0, and the maximum number is the size M of all noiseless-ID codeword subsets $\mathcal{C}_{\boldsymbol{u}}$, i.e., $0 \leq K \leq M$. The pairwise overlap size $K(\boldsymbol{a}, \boldsymbol{b})$ of their respective noiseless-ID codeword subsets is a property of each pair $(\boldsymbol{a}, \boldsymbol{b})$ of messages. Note that for the identical pair $(\boldsymbol{a}, \boldsymbol{a})$, the overlap size $K(\boldsymbol{a}, \boldsymbol{a})$ is the subset size M.

Commonly, each noiseless-ID codeword $\boldsymbol{c}$ is equally likely to be randomly selected in the encoding process [1]. In other words, the probability distribution over the subset $\mathcal{C}_{\boldsymbol{u}}$ associated with any message $\boldsymbol{u}$ is uniform. In the uniform case, the collision probability (i.e., the probability of selecting a noiseless-ID codeword $\boldsymbol{c} \in \mathcal{C}_{\boldsymbol{a}} \cap \mathcal{C}_{\boldsymbol{b}}$) is directly proportional to the overlap size $K(\boldsymbol{a}, \boldsymbol{b})$. Specifically, the FP error probability $p_\mathrm{fp}(\boldsymbol{a}, \boldsymbol{b})$ of the pair $(\boldsymbol{a}, \boldsymbol{b})$ of messages is given by

$$p_\mathrm{fp}(\boldsymbol{a}, \boldsymbol{b}) = \frac{K(\boldsymbol{a}, \boldsymbol{b})}{M}, \tag{37}$$

with $M = |\mathcal{C}_{\boldsymbol{a}}| = |\mathcal{C}_{\boldsymbol{b}}|$. The noiseless-ID FP error probability $p_\mathrm{fp}(\boldsymbol{a}, \boldsymbol{b})$ is the property of each *pair* $(\boldsymbol{a}, \boldsymbol{b})$ of messages,

cf. Section IV-E for noisy-ID codes. For noiseless-ID codes, investigating the overlap $K(\boldsymbol{a},\boldsymbol{b})$ provides all necessary information to determine the collision probability and thereby all error probabilities of noiseless-ID codes.

Since the FP error probability $p_{\text{fp}}(\boldsymbol{a},\boldsymbol{b})$ is specific to each pair $(\boldsymbol{a},\boldsymbol{b})$ of messages, it is helpful to have metrics that summarize the FP error probabilities $p_{\text{fp}}(\boldsymbol{a},\boldsymbol{b})$ of all message pairs $(\boldsymbol{a},\boldsymbol{b})$. Specifically, the FP error probability bound λ_{fp} is an upper bound on the FP error probabilities of all message pairs that can be selected from the set $\mathcal{U}$ of messages.

$$\lambda_{\text{fp}} = \max_{\boldsymbol{a},\boldsymbol{b}\in\mathcal{U},\boldsymbol{a}\neq\boldsymbol{b}} p_{\text{fp}}(\boldsymbol{a},\boldsymbol{b}). \tag{38}$$

With Eq. (37), this can be reformulated to use the upper bound κ on the overlap sizes $K(\boldsymbol{a},\boldsymbol{b})$ of all message pairs $(\boldsymbol{a},\boldsymbol{b})$:

$$\lambda_{\text{fp}} = \frac{1}{M} \max_{\boldsymbol{a},\boldsymbol{b}\in\mathcal{U},\boldsymbol{a}\neq\boldsymbol{b}} K(\boldsymbol{a},\boldsymbol{b}) = \frac{\kappa}{M}. \tag{39}$$

Next to the upper bound λ_{fp} on the collision probabilities $p_{\text{fp}}(\boldsymbol{a},\boldsymbol{b})$, it is possible to characterize the collision probabilities by their mean $\overline{p_{\text{fp}}}$ and quantiles. Specifically, the mean collision probability $\overline{p_{\text{fp}}}$ is often the inverse of the field size q, i.e.,

$$\overline{p_{\text{fp}}} = \frac{1}{q}. \tag{40}$$

The quantiles of collision probabilities generalize the upper bound λ_{fp}. While the upper bound λ_{fp} is a collision probability that is larger than the collision probability of all message pairs $(\boldsymbol{a},\boldsymbol{b})$, a threshold probability λ_α is the quantile collision probability that is larger than the collision probability of $1-10^{\alpha}$ of all message pairs $(\boldsymbol{a},\boldsymbol{b})$ [95, Eq. (15)]. As $\alpha\to\infty$, the quantile collision probability λ_α approaches the collision probability bound λ_{fp}.

D. ID Code Rate and Capacity

The noiseless-ID code rate R_{id} describes the message size k relative to the noiseless-ID codeword size m. The rate R_{id} can scale double-exponentially (similar to the noisy-ID code rate, cf. Section IV-G) and still allow asymptotically for an error probability bound λ_{fp} that approaches 0. Specifically, the noiseless-ID code rate is

$$\begin{aligned} R_{\text{id}} &= \frac{\log_2\left(\log_2(|\mathcal{U}|)\right)}{\log_2(|\mathcal{C}|)} = \frac{\log_2\left(\log_2(N)\right)}{\log_2(S)} \\ &= \frac{\log_2\left(k\log_2(q)\right)}{m\log_2(q)} = \frac{\log_2\left(k'\right)}{m'} \leq 1. \end{aligned} \tag{41}$$

The noiseless-ID code rate R_{id} is not only limited by the ID capacity C of the channel (whereby $C=1$ for the noiseless channel). The desired *error exponent* E_{fp} that is a representation of the collision probability bound λ_{fp} limits the noiseless-ID code rate as well. Specifically, and analogously to noisy-ID codes, cf. Eq. (26),

$$R_{\text{id}} + 2E_{\text{fp}} \leq 1, \tag{42}$$

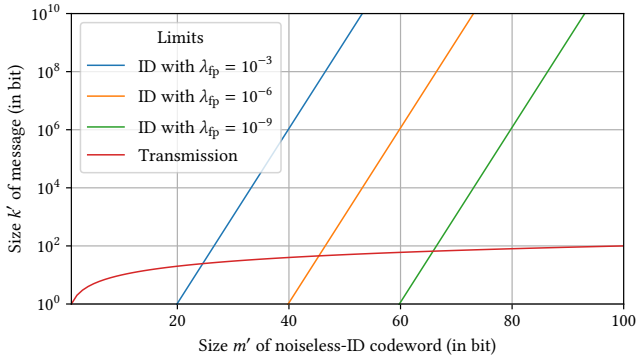


Fig. 13. Capacity of message transmission and noiseless identification. Over a noiseless channel (the identity channel), the code rate of noiseless-ID codes can scale exponentially faster than the rate of a "channel code" for message transmission. Each coordinate in the figure corresponds to a rate that is a ratio of message size k' and codeword size m'. The gain in terms of the rate $(\log_2(k'))/m'$ comes at the cost of a non-zero error probability. The smaller the desired error probability, the larger the required codeword size m', cf. Eq. (44).

with the error exponent adapted to noiseless-ID codeword length m' from Eq. (25)

$$E_{\text{fp}} = -\frac{\log_2(\lambda_{\text{fp}})}{m'}. \tag{43}$$

As shown in the Appendix, Eq. (42) can be reformulated to

$$k' \leq 2^{m'}\lambda_{\text{fp}}^2 = S\lambda_{\text{fp}}^2. \tag{44}$$

This is a fundamental limit of noiseless-ID codes and describes a tradeoff between the size k' of supported messages in bit, the size m' of the transmitted noiseless-ID codeword in bit, and the FP error probability bound λ_{fp} that limits the error probability.

This means that it is possible to find an noiseless-ID code that supports message sizes k' that are up to exponentially larger than the noiseless-ID codeword size m' and still enable reliable noiseless-ID. However, a practical noiseless-ID code can never reach the full exponential scaling because guaranteeing a small error probability bound prohibits that the ID code rate equals 1. In other terms, for a fixed "budget" m' of bits that are transmitted as the noiseless-ID codeword $\boldsymbol{c}$, the noiseless-ID code commits some bits to increasing the code rate R_{id} (for efficiency), and the remaining bits to reducing the error exponent E_{fp} (for reliability).

Figure 13 visualizes the capacity of noiseless-ID codes. A desirable code lies in the top left corner of the figure, supporting large messages while maintaining small codeword sizes. The capacity of (collision-free) message transmission is depicted for reference. Noiseless-ID codes can support up to exponentially more messages than message transmission at the cost of allowing for a small collision probability. The smaller the collision probability bound of a noiseless-ID code should be, the larger the codeword must be. For example, if the noiseless-ID codeword should be $m'=60$ bit long, then it is possible to find a noiseless-ID code that guarantees a collision probability bound $\lambda_{\text{fp}}=10^{-6}$ and supports messages up to $k'=10^6$ bit. Finding a noiseless-ID code that guarantees a collision probability bound $\lambda_{\text{fp}}=10^{-9}$ (ceteris paribus)

is impossible because it exceeds the capacity by violating Eq. (42).

Randomized noiseless-ID codes (the focus of this tutorial) rely on randomness. If Alice and Bob have an efficient method to agree on common randomness [110]–[113] (that the encoding "consumes"), then this common randomness can increase the ID capacity. If Alice and Bob have access to a feedback link, then this can also increase the capacity [114] as Alice and Bob can use the feedback link, for example, to generate common randomness. In message transmission, feedback does not increase the capacity [115]. The usefulness of common randomness to the capacity opens new possibilities unknown from message transmission, such as generating common randomness during night time and using the stored randomness during day time.

E. Noiseless-ID Codes and Universal Hashing

Randomized noiseless-ID codes are related to ϵ-almost universal hash functions [116]. Additionally, geometric codes are known to be useful in providing guarantees about collision resistance for universal hash functions and message authentication codes [117]–[119]. There are also universal hash functions that use the properties of polynomial codes to guarantee collision resistance [120], [121].

Much like universal hash functions, noiseless-ID encoding functions map from a set $\mathcal{U}$ of messages to a smaller set $\mathcal{C}$ of noiseless-ID codewords ("hashes"), thereby improving efficiency. The set $\mathcal{C}_{\boldsymbol{a}}$ of noiseless-ID codewords corresponds to the union of the outputs of all hash functions in the family of hash functions in universal hashing given the input message $\boldsymbol{a}$. A randomized noiseless-ID code provides strict worst-case guarantees on the "hash collision" probability. Due to these guarantees, a "good" noiseless-ID code asymptotically enables virtually error-free noiseless-ID despite a severe information loss in the encoding ("hashing").

Universal hashes also perform a random encoding but aim to distribute data evenly among the finite number of hashes. Providing worst-case guarantees (via the collision probability bound λ_{fp}) is generally not the principal design requirement of universal hash functions. In noiseless-ID codes, a guaranteed upper bound λ_{fp} on the probability of false positive ID is essential.

Generally, hash functions have arbitrary input sizes, whereas noiseless-ID codes limit the size of the input. This is because for an arbitrary input size, there are potentially infinitely many messages that collide in each noiseless-ID codeword (that is the hash). However, a noiseless-ID code that supports messages that are larger than several TB can practically be considered to support virtually arbitrary file sizes.

Generally, noiseless-ID codes are not cryptographic. A survey of different families of hash functions and a comprehensive comparison to noiseless-ID codes is out of the scope of this tutorial.

F. Summary

Noiseless-ID codes aim to address the ID problem via noiseless channels efficiently and reliably and can be used in concatenation with channel codes to construct noisy-ID codes to address ID via noisy channels. A noiseless-ID encoder "hashes" the sender's message $\boldsymbol{a}$ to a *random* short codeword $\boldsymbol{c}$ (the "hash") whereby each message $\boldsymbol{u}$ is associated with a set $\mathcal{C}_{\boldsymbol{u}}$ of codewords. That is, the noiseless-ID encoder selects a random codeword $\boldsymbol{c}$ from the set $\mathcal{C}_{\boldsymbol{u}}$ of the encoded message $\boldsymbol{u}$. The mapping between each message and its random codeword is not unique. Rather, many messages are associated with each codeword. A good noiseless-ID code provides guarantees about the fraction of noiseless-ID codewords that is shared among any pair of messages. Thereby, if the overlap between two codeword sets is small, then also the probability to identify a message in error is small because the encoder is unlikely to select one of the few shared codewords. This way, noiseless-ID codes provide reliability guarantees despite the lossy encoding that increases the efficiency of the scheme. If the codeword size grows large, then the error probability (that the lossy encoding entails) can approach zero. However, for practical codeword lengths, noiseless-ID encoding always entails a finite non-zero error probability. This is one of the costs associated with the ID-specific efficiency of noiseless-ID codes.

The literature has proposed two related coding methods to explicitly construct noiseless-ID codes: tagging codes that Section VII explains, and constant-weight codes that Section VIII explains. All CWCs that the literature proposes for the noiseless-ID problem are based on tagging codes, i.e., all proposed ID-CWCs are derived from tagging codes. Thereby, tagging codes are considered the state-of-the-art noiseless-ID codes.

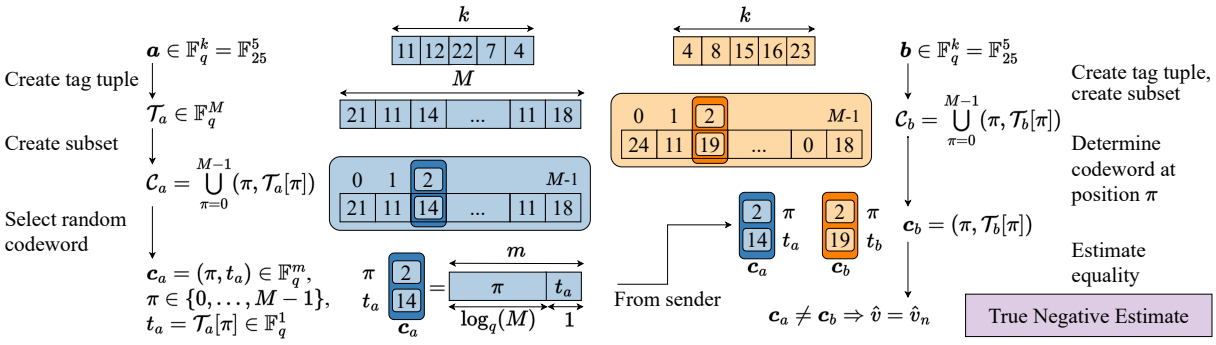
VII. Tagging Codes

A. Principle

1) Overview: Tagging codes are an explicit randomized noiseless-ID code construction and can enable efficient noiseless-ID while maintaining strong error probability guarantees. While [21] first proposed the method to explicitly construct noiseless-ID codes, the term *tagging code* was only later coined in [26]. Tagging codes comprise a randomized encoding function and define how to determine an equality estimate $\hat{v}$ by directly comparing the noiseless-ID codeword $\boldsymbol{c_a}$ (that is a "hash" of the sender's message $\boldsymbol{a}$) with the noiseless-ID codeword $\boldsymbol{c_b}$ (that is a "hash" of the receiver's message $\boldsymbol{b}$). We first explain tagging codes in a relatively standalone fashion in this Section VII-A, while we leave the description of how tagging codes implement the sets and functions of noiseless-ID codes that Section VI explained to Section VII-B.

Tagging codes are defined by two components: the tagging encoder and the tagging verification rule. First, we explain the tagging encoder. Tagging codes repurpose linear block codes for the encoding step. Block codes with a high Hamming distance are well-suited for tagging code construction because they guarantee low collision probabilities $p_{\text{fp}}(\boldsymbol{a}, \boldsymbol{b})$ as Section VII-C will explain.

2) Tagging Encoder: Both the sender and the receiver use a tagging encoder. Therefore, we describe the tagging encoding for an arbitrary message $\boldsymbol{u}$ that could be either the message $\boldsymbol{a}$



(a) Tagging encoding. (b) Tagging verification.

Fig. 14. Overview and example of tagging encoding and verification steps.

TABLE IX
SUMMARY OF NOTATIONS FOR TAGGING CODES.

$\mathcal{T}_{\boldsymbol{u}}$	Tuple of tags of message $\boldsymbol{u}$
t	Tag
π	Position of tag in tuple $\mathcal{T}_{\boldsymbol{u}}$ of tags
$\boldsymbol{c} = (t, \pi)$	Noiseless-ID codeword
$D(\boldsymbol{a}, \boldsymbol{b})$	Short for Hamming distance $D(\mathcal{T}_{\boldsymbol{a}}, \mathcal{T}_{\boldsymbol{b}})$
δ	Lower bound on Hamming distance between all pairs
T	Number of transmitted tags

of the sender or the message $\boldsymbol{b}$ of the receiver. We visualize the encoding process in Fig. 14a.

As the first of two elements of the tagging encoder, a block encoder maps the message $\boldsymbol{u} \in \mathcal{U}$ of size $\log_q(|\mathcal{U}|) = k$ to a block codeword $\mathcal{T}_{\boldsymbol{u}}$ of size $\log_q(|\mathcal{T}_{\boldsymbol{u}}|) = M \geq k$. The encoding function $f_{\mathcal{T}}$ that formalizes this step operates in finite fields, i.e.,

$$f_{\mathcal{T}} : \boldsymbol{u} \mapsto \mathcal{T}_{\boldsymbol{u}} \in \mathbb{F}_q^M \quad \forall \boldsymbol{u} \in \mathcal{U} = \mathbb{F}_q^k. \tag{45}$$

Detailed explanation of finite field arithmetic is outside the scope of this tutorial and we refer to [23] instead. As the second tagging encoding step, the tagging encoder selects the symbol (that is called a *tag* t) at a random position π from the block codeword $\mathcal{T}_{\boldsymbol{u}}$. Hence, the block codeword $\mathcal{T}_{\boldsymbol{u}}$ can be considered to be an M-tuple $\mathcal{T}_{\boldsymbol{u}}$ of tags. In other words, the tagging encoder selects a random tag t from the tag tuple $\mathcal{T}_{\boldsymbol{u}}$ that is associated with message $\boldsymbol{u}$. Together with the random position π, this tag t forms the randomized noiseless-ID codeword $\boldsymbol{c} = (\pi, t)$. In the example in Fig. 14a, the sender randomly selects position $\pi = 2$ in the tuple $\mathcal{T}_{\boldsymbol{a}}$ and determines the tag $t = 14$. In parts of the literature, the random tag position π is referred to as a *coloring number*, the tag value t as the *color*, and the message $\boldsymbol{u}$ a *coloring*.

Note that, in principle, the tag tuple encoding function $f_{\mathcal{T}}$ in Eq. (45) is identical to the block encoding function f_{fec} in Eq. (7). However, we choose the alternate notation to highlight the entirely different use of the respective output of the two functions: The output of the block encoding function f_{fec} is a size n block (channel) codeword $\boldsymbol{x} \in \mathcal{X} = \mathbb{F}_q^n$ that the sender transmits over the channel. On the other hand, the output of the tag tuple encoding function $f_{\mathcal{T}} \in \mathbb{F}_q^M$,is further processed and only one symbol of the "block codeword" $\mathcal{T}_{\boldsymbol{u}}$ (that is the tag tuple) is transmitted, i.e., the tag t is the only element of the "block codeword" $\mathcal{T}_{\boldsymbol{u}}$ that the sender transmits. For the size of the tag tuple $\mathcal{T}_{\boldsymbol{u}}$, we write the size M instead of the size n of a block (channel) codeword. Thereby, the tagging code employs an $(M, k, \delta)_q$ block code, whereas a channel code is an $(n, k, \delta)_q$ block code.

The tagging encoder obtains the random tag t by selecting a random position

$$\pi \in \{0, \ldots, M-1\} \tag{46}$$

in the tuple $\mathcal{T}_{\boldsymbol{u}}$ of tags. The position π can be considered the index of the selected tag t in the tuple $\mathcal{T}_{\boldsymbol{u}}$:

$$t = \mathcal{T}_{\boldsymbol{u}}[\pi] \in \mathbb{F}_q. \tag{47}$$

The (position, tag) pair constitutes the tagging (ID) codeword $\boldsymbol{c}$. In other words, a tagging codeword $\boldsymbol{c} = (\pi, t)$ is an ordered pair of a position π within the tuple $\mathcal{T}_{\boldsymbol{u}}$, and the tag value $t = \mathcal{T}_{\boldsymbol{u}}[\pi]$ at that position π. The encoding function of tagging codes is

$$f_{\text{tag}} : \boldsymbol{u} \xmapsto{\text{rand.}} \boldsymbol{c} = (\pi, \mathcal{T}_{\boldsymbol{u}}[\pi]) \quad \forall \boldsymbol{u} \in \mathcal{U} = \mathbb{F}_q^k. \tag{48}$$

3) Tagging Verification Rule: Next to the tagging encoder, the tagging verification rule is the second element that constitutes a tagging code. We visualize the verification in Fig. 14b. Tagging codes verify equality using the tagging (ID) codewords $\boldsymbol{c_a}$ and $\boldsymbol{c_b}$ of the respective messages of the sender and the receiver. Specifically, the sender determines its tagging codeword $\boldsymbol{c_a} = (\pi, \mathcal{T}_{\boldsymbol{a}}[\pi])$ via $f_{\text{tag}}(\boldsymbol{u} = \boldsymbol{a})$ and transmits the tagging codeword via the noiseless channel $\boldsymbol{V}$ to the receiver. The receiver extracts the random position π (that the sender determined) from the received tagging codeword $\boldsymbol{c_a} = (\pi, \mathcal{T}_{\boldsymbol{a}}[\pi])$. Recall that, for noiseless-ID coding, we consider the channel to be noiseless and to always perfectly reproduce the transmitted tagging (ID) codeword $\boldsymbol{c_a}$ at the receiver. We leave the discussion of the implications of tagging coding via noisy channels to Section IX.

Using the extracted random position π, the receiver determines its own tagging codeword $\boldsymbol{c_b} = (\pi, \mathcal{T}_{\boldsymbol{b}}[\pi])$ via $f_{\text{tag}}(\boldsymbol{u} = \boldsymbol{b})$ at the same position π as the codeword $\boldsymbol{c_a}$ of the sender. In the final step, the receiver compares whether the two codewords are identical, i.e., whether $\boldsymbol{c_a} = \boldsymbol{c_b}$. Since both tagging codewords $\boldsymbol{c_a}$ and $\boldsymbol{c_b}$ share the same position π

by design of the tagging code (the verifier uses the received position π to create its tagging codeword $\boldsymbol{c_b}$), only the two tags need to be compared. Thereby, the tagging verification rule is

$$\hat{v} = \begin{cases} \hat{v}_\text{p} \text{ if } t_{\boldsymbol{a}} = t_{\boldsymbol{b}} \Leftrightarrow \mathcal{T}_{\boldsymbol{a}}[\pi] = \mathcal{T}_{\boldsymbol{b}}[\pi] \Leftrightarrow \boldsymbol{c_a} = \boldsymbol{c_b}, \\ \hat{v}_\text{n} \text{ if } t_{\boldsymbol{a}} \neq t_{\boldsymbol{b}} \Leftrightarrow \mathcal{T}_{\boldsymbol{a}}[\pi] \neq \mathcal{T}_{\boldsymbol{b}}[\pi] \Leftrightarrow \boldsymbol{c_a} \neq \boldsymbol{c_b}. \end{cases} \tag{49}$$

B. Tagging Codeword Sets

Since each tagging code is a noiseless-ID code, it is possible to define the subset $\mathcal{C}_{\boldsymbol{u}}$ of noiseless-ID codewords of each message $\boldsymbol{u}$ and the set $\mathcal{C}$ of all possible noiseless-ID codewords for tagging codes. The subset $\mathcal{C}_{\boldsymbol{u}}$ of tagging codewords is the union of all tagging codewords $\boldsymbol{c_u} = (\pi, \mathcal{T}_{\boldsymbol{u}}[\pi])$ that are associated with message $\boldsymbol{u}$:

$$\mathcal{C}_{\boldsymbol{u}} = \bigcup_{\pi=0}^{M-1} (\pi,\, \mathcal{T}_{\boldsymbol{u}}[\pi]). \tag{50}$$

Thereby, there are $M = |\mathcal{T}_{\boldsymbol{u}}| = |\mathcal{C}_{\boldsymbol{u}}|$ tagging codewords in each subset $\mathcal{C}_{\boldsymbol{u}}$ associated with message $\boldsymbol{u}$. The set $\mathcal{C}$ of all tagging codewords is defined as the union over all possible (position, tag)-pairs:

$$\mathcal{C} = \bigcup_{\substack{\pi=0,\dots,M-1 \\ t=0,\dots,q-1}} (\pi, t). \tag{51}$$

The block encoder $f_{\mathcal{T}}$ that generates the tag tuple $\mathcal{T}_{\boldsymbol{u}}$ of message $\boldsymbol{u}$ can be considered part of an overall *subset encoder* that generates the noiseless-ID codeword subset $\mathcal{C}_{\boldsymbol{u}}$ for message $\boldsymbol{u}$. After the subset encoder generates the subset $\mathcal{C}_{\boldsymbol{u}}$ for the message $\boldsymbol{u}$ (i.e., the first step of the encoding procedure), the tagging encoder randomly selects a tagging codeword $\boldsymbol{c}$ by selecting a random position $\pi \in \{0, \dots, M-1\}$ in the second step of the encoding procedure. The tagging codeword is determined as

$$\boldsymbol{c} = \mathcal{C}_{\boldsymbol{u}}[\pi] = (\pi, \mathcal{T}_{\boldsymbol{u}}[\pi]), \tag{52}$$

and the tagging encoding function Eq. (48) can be extended to

$$f_\text{tag} : \boldsymbol{u} \xrightarrow{\text{rand.}} \boldsymbol{c} = (\pi, \mathcal{T}_{\boldsymbol{u}}[\pi]) \in \mathcal{C}_{\boldsymbol{u}} \subset \mathcal{C} \quad \forall \boldsymbol{u} \in \mathcal{U} = \mathbb{F}_q^k. \tag{53}$$

Note that the block encoder *deterministically* creates the tag tuple $\mathcal{T}_{\boldsymbol{u}}$ from message $\boldsymbol{u}$. Thereby, also the subset $\mathcal{C}_{\boldsymbol{u}}$ of tagging codewords is deterministic. By sampling a random element $\boldsymbol{c}$ from the deterministic subset $\mathcal{C}_{\boldsymbol{u}}$, the tagging encoding f_tag becomes a randomized noiseless-ID encoding. This separation matches the explanation in Section IV-B1 that describes noiseless-ID encoding as a two-step procedure: i) a deterministic subset encoding, and ii) a random sampling.

Because each tagging codeword $\boldsymbol{c}$ consists of the random position $\pi \in \{0, \dots, M-1\}$, and the tag $t = \mathcal{T}_{\boldsymbol{u}}[\pi] \in \{0, \dots, q-1\}$, there are $S = |\mathcal{C}| = M \cdot q$ possible tagging codewords $\boldsymbol{c}$. To stay consistent with the representation of the noiseless-ID codeword $\boldsymbol{c}$ as m-tuple in base q, cf. Section IV, it is possible to represent the tagging codeword $\boldsymbol{c} = (\pi, \mathcal{T}_{\boldsymbol{u}}[\pi])$ as an m-tuple in base q. In other words, a single array of m symbols in base q represents the tagging codeword $\boldsymbol{c_u}$, cf. Fig. 14a, in place of the pair $(\pi, \mathcal{T}_{\boldsymbol{u}}[\pi])$. Specifically, $\log_q(M)$ symbols in base q represent the position π, and a single symbol represents the tag t (that is in base q in either representation). Hence, the size m of the q-ary tagging codeword $\boldsymbol{c}$ is

$$m = \log_q(M) + 1. \tag{54}$$

With this, the number of tagging codewords that the tagging code can represent is

$$S = |\mathcal{C}| = q^m = q^{\log_q(M)+1} = M \cdot q. \tag{55}$$

In a real implementation, the size m must be an integer.

Furthermore, the tagging code supports $N = |\mathcal{U}| = |\mathbb{F}_q^k| = q^k$ messages. Typically $m < k$, i.e., the tagging (ID) codeword $\boldsymbol{c}$ is smaller than the input message $\boldsymbol{u}$. The noiseless-ID code rate, cf. Eq. (41), of a tagging code is

$$R_\text{tag} = \frac{\log_2(k')}{m'} = \frac{\log_2(k \log_2(q))}{\log_2(M) + \log_2(q)} \leq 1. \tag{56}$$

In contrast to the noiseless-ID coding verification rule in Eq. (35), the tagging verification rule in Eq. (49) only implicitly tests whether the received tagging codeword $\boldsymbol{c_a}$ of the sender is an element of the codeword subset $\mathcal{C}_{\boldsymbol{b}}$ of the receiver's message $\boldsymbol{b}$. The codeword subset $\mathcal{C}_{\boldsymbol{b}}$ of the receiver's message $\boldsymbol{b}$ consists of (position, tag)-pairs, cf. Eq. (50). Each position π "generates" one element of the codeword subset $\mathcal{C}_{\boldsymbol{b}}$ and each position value $\pi \in \{0, \dots, M-1\}$ is used exactly once in each codeword subset $\mathcal{C}_{\boldsymbol{b}}$. Thereby, for a given receiver message $\boldsymbol{b}$, the codeword subset $\mathcal{C}_{\boldsymbol{b}}$ holds exactly one tagging codeword $\boldsymbol{c}$ that includes the position π that the receiver extracted from the received tagging codeword $\boldsymbol{c_a}$. In other words, if any tagging codeword in the codeword subset $\mathcal{C}_{\boldsymbol{b}}$ matches the received tagging codeword $\boldsymbol{c_a}$, then it must be the tagging codeword at the same position π as the received tagging codeword. Thereby, comparing the tag value $t_{\boldsymbol{b}} = \mathcal{T}_{\boldsymbol{b}}[\pi]$ with the received tag value $t_{\boldsymbol{a}}$ implicitly corresponds to testing whether the received tagging codeword $\boldsymbol{c_a}$ is an element of the codeword subset $\mathcal{C}_{\boldsymbol{b}}$ of the receiver's message $\boldsymbol{b}$.

C. Collision Probability Guarantees

As Section VI-C explained, noiseless-ID codes, such as tagging codes, are subject only to FP errors due to codeword collision; whereas, FN errors cannot occur. The collision probability $p_\text{fp}(\boldsymbol{a}, \boldsymbol{b})$ is determined by the size of the overlap $K(\boldsymbol{a}, \boldsymbol{b})$ between two codeword subsets of messages $\boldsymbol{a}$ and $\boldsymbol{b}$, i.e., by the number $K(\boldsymbol{a}, \boldsymbol{b})$ of codewords that both subsets share. Tagging codes are built from linear block codes that often have known Hamming distance properties. Recall that Section III-C explained the Hamming distance. For the noiseless-ID codeword subsets of two messages, the overlap $K(\boldsymbol{a}, \boldsymbol{b})$ and the Hamming distance $D(\boldsymbol{a}, \boldsymbol{b})$ are inverses of each other, i.e.,

$$D(\boldsymbol{a}, \boldsymbol{b}) = M - |\mathcal{C}_{\boldsymbol{a}} \cap \mathcal{C}_{\boldsymbol{b}}| = M - K(\boldsymbol{a}, \boldsymbol{b}). \tag{57}$$

Analogously, the collision probability $p_\text{fp}(\boldsymbol{a}, \boldsymbol{b})$, cf. Eq. (37), can be reformulated to

$$p_\text{fp}(\boldsymbol{a}, \boldsymbol{b}) = \frac{K(\boldsymbol{a}, \boldsymbol{b})}{M} = 1 - \frac{D(\boldsymbol{a}, \boldsymbol{b})}{M} = p_\text{fp}[D(\boldsymbol{a}, \boldsymbol{b})]. \tag{58}$$

Note that $p_{\text{fp}}[D(\boldsymbol{a},\boldsymbol{b})]$ takes on a finite number of possible values as there are a finite number of distance values between the codeword subsets of two messages.

For example, in Fig. 14, the subsets $\mathcal{C}_{\boldsymbol{a}}$ and $\mathcal{C}_{\boldsymbol{b}}$ of noiseless-ID codewords of the pair $(\boldsymbol{a},\boldsymbol{b})$ of messages overlap in (at least) two positions ($\pi = 1$ and $\pi = M-1$), and they differ in (at least) three positions. If we only consider the five shown positions, then $M = 5$, $K = 2$, $D = 3$, and $p_{\text{fp}}(\boldsymbol{a},\boldsymbol{b}) = 0.4$.

For the upper bound λ_{fp} on the collision probability, Eq. (39) can be reformulated to use the distance D instead of the overlap K:

$$\begin{aligned} \lambda_{\text{fp}} &= \max_{\boldsymbol{a},\boldsymbol{b}\in\mathcal{U},\boldsymbol{a}\neq\boldsymbol{b}} p_{\text{fp}}(\boldsymbol{a},\boldsymbol{b}) \\ &= \frac{\max_{\boldsymbol{a},\boldsymbol{b}\in\mathcal{U},\boldsymbol{a}\neq\boldsymbol{b}} K(\boldsymbol{a},\boldsymbol{b})}{M} = \frac{\kappa}{M} \\ &= 1 - \frac{\min_{\boldsymbol{a},\boldsymbol{b}\in\mathcal{U},\boldsymbol{a}\neq\boldsymbol{b}} D(\boldsymbol{a},\boldsymbol{b})}{M} = 1 - \frac{\delta}{M}. \end{aligned} \tag{59}$$

An $(M,k,\delta)_q$ block code with a high minimum Hamming distance δ can be suitable to create a tagging code with a low collision probability bound λ_{fp}. That is because the minimum Hamming distance δ property of the underlying block code corresponds to the minimum Hamming distance δ between the tag tuples $\mathcal{T}_{\boldsymbol{u}}$ of pairs of messages.

The minimum Hamming distance δ of a noiseless-ID code is a lower bound on the Hamming distance for all pairs of noiseless-ID codeword subsets in the noiseless-ID code. Section III-C explained the distribution of distances for all message pairs. The distance distribution can provide additional metrics to evaluate noiseless-ID codes, e.g., via quantile collision probabilities [95], cf. Section VI-C.

D. Transmit Multiple Tags

The sender can compute and transmit multiple tags of the same message $\boldsymbol{a}$ to reduce the collision probability. The tagging encoder determines the tag values at $T > 1$ positions in the tag tuple $\mathcal{T}$.

In the example in Fig. 14, if the sender randomly selects position $\pi = 1$, then the resulting tag value $t = 11$ matches the tag value of the receiver. If the sender additionally transmits the tag at position $\pi = 0$, then the collision is avoided because the tags of sender and receiver do not collide in position $\pi = 0$.

Sending multiple tags increases the amount of data that the sender has to transmit, thereby reducing the code rate and the efficiency of the scheme. However, each additional tag reduces the mean collision probability by a factor $1/q$ thus improving reliability. Sending multiple tags can also reduce the collision probability bound λ_{fp}. The exact benefit can be computed if the distance distribution of the underlying block code is known [95]. Asymptotically, as the number T of transmitted tags approaches the size k of the message, the sender transmits enough information for the receiver to *decode* the message $\boldsymbol{a}$ of the sender instead of just identifying it. Thereby, the number of tags describes a sliding tradeoff between ID and message transmission.

E. Role of Randomness

If the receiver can reproduce random symbols of the sender (i.e., reproduce part of the random position π) without using the channel, then higher rates and lower collision probability bounds are possible. Identical randomness (or, random data) that the sender and receiver both agree upon is called common randomness (CR) [110]–[113]. CR increases the ID capacity of a channel [122], [123]. The CR rate is added on top of the ID capacity of the channel. If the CR rate is high enough and generates all $\log_q(M)$ symbols that represent the random position π in the tag tuple $\mathcal{T}_{\boldsymbol{u}}$, then only the tag t has to be transmitted to the receiver. The noiseless-ID code conveys only one q-ary symbol (that is the tag t) instead of conveying m q-ary symbols. Therefore, the code rate scales according to

$$R_{\text{id,CR}} = \frac{\log_q(k')}{\log_2(q)} = \frac{\log_2(k')}{(\log_2(q))^2}. \tag{60}$$

A simple method to produce the same random numbers in two parties lies in using a pseudo-random number generator (PRNG) that both parties initialize with the same seed. When encoding, the sender uses a pseudo-random number from the PRNG as the random position π to generate a tag t from the message $\boldsymbol{a}$. Because the receiver uses a PRNG that is identical to the PRNG of the sender, it generates the same pseudo-random number and uses it to compute the tag t of its message $\boldsymbol{b}$. The receiver then compares the received tag with its own tag. This way, no randomness has to be transmitted. Both parties only have to agree on a common seed of the PRNG ahead of the transmission.

If adversaries intercept the common seed for the PRNG, then they are able to reliably identify the message $\boldsymbol{a}$ of the sender. Therefore, other methods of creating CR between the sender and receiver that are not interceptable can ensure the secrecy of the communication [124], [125]. If the random position π is known only to the sender and the receiver, then intercepting the transmitted tag does not provide an eavesdropper with any information about the message of the sender. This holds not only for noiseless-ID codes but also for noisy-ID codes. For noisy-ID codes, this property is sometimes framed in the context of physical-layer security [126], [127]. The ID capacity of a channel coincides with the secret ID capacity of a channel, i.e., no additional bits are needed to make the communication information-theoretically secret. This property is sometimes referred to as secrecy for free [128].

If the sender and receiver do not face any adversaries, then it is not always necessary to reroll the random position π for each new ID round. Rather, in practice, it may suffice to agree on one random position and to use it for several communication rounds. This way, the random position does not need to be changed with each round. This could save the transmission cost of sending the random position, save the processing time of generating the next pseudo-random number from the PRNG, or save the use of securely generated CR. However, the rerandomization in each round serves the additional purpose of breaking up collisions. If the tags of a message pair $(\boldsymbol{a},\boldsymbol{b})$ collide for the selected random position π, then each round that observes the same message pair and the same random position will produce false-positive equality

estimates at the receiver. In other words, rerolling the random position π in each round ensures that each round has a non-zero probability of generating a non-colliding tag (hash).

F. Explicit Tagging Code Construction

The literature proposes several different tagging code families. A "good" tagging code should provide high reliability via low collision probabilities including the collision probability bound λ_{fp}, the mean collision probability $\overline{p_{\text{fp}}}$, and quantile collision probabilities λ_α. Simultaneously, the tagging code should be efficient, i.e., achieve a high code rate R_{id} that corresponds to large message sizes k and small codeword sizes m.

Reed-Solomon (RSID) codes achieve the Singleton bound and thereby offer the highest possible distance bound δ on the tag tuples $\mathcal{T}$. By "concatenating" two RSID codes (of purposefully chosen parameters) it is possible to construct a noiseless-ID code that approaches the ID capacity as the codeword size m approaches infinity [21]. We refer to this "concatenated" RSID code as an RS2ID code. The "concatenation" is not to be confused with the concatenation of error correction codes. Specifically, multiple instances of the second RSID encoder encode each output symbol of the first RSID encoder in parallel and concatenate the resulting symbol arrays. A detailed explanation of the "concatenation" is out of the scope of this tutorial; instead, we refer to [94, Fig. 1].

Reed-Muller (RMID) codes do not achieve the Singleton bound, i.e., they have higher (worse) collision probability bounds than comparable RSID codes. Yet, the encoding is computationally cheaper than RSID codes [129]. RSID, RS2ID, and RMID codes have been compared in [94].

Random linear (RLID) codes can achieve rates close to the ID capacity also at finite codeword lengths m [130]. RLID codes very likely have a high distance if the symbol size q is large. Tagging codes naturally benefit from large symbol sizes q because larger symbol sizes correspond to smaller collision probabilities.

It is possible to concatenate tagging codes with constant-weight codes (CWCs), thereby creating an overall CWC for ID, as explained in Section VIII.

G. Implementation

The literature investigates how to efficiently implement tagging codes [26], [131], [132]. In tagging codes, it is not necessary to compute the full tuple $\mathcal{T}_{\boldsymbol{u}}[0:M-1]$ of tags before sampling the random tag $t = \mathcal{T}_{\boldsymbol{u}}[\pi]$. Rather, [26] proposes to directly compute the required tag $t = \mathcal{T}_{\boldsymbol{u}}[\pi]$, thereby avoiding computing the remaining $M-1$ tags only to discard them.

Limiting valid field sizes q to binary extension fields allows for further optimization. Specifically, for $q = 2^8$ and $q = 2^{16}$, the g2p library [133] can speed up the necessary finite field multiplications via lookup tables [132]. For larger field sizes, lookup tables become prohibitive in terms of memory. Instead, for $q = 2^{32}$ and $q = 2^{64}$, modern processors (including amd64 and aarch64) offer carryless multiply instruction sets (CLMU). After the polynomial multiplication into a 64-bit or 128-bit polynomial, respectively, a Barrett reduction via CLMU determines the modulus to map the result back into the 32-bit or 64-bit field, [132], [134].

Consider the message $\boldsymbol{u}$ that consists of k symbols, whereby u_i is the i'th symbol in the message. The tagging encoder determines the tag tuple $\mathcal{T}$ that consists of M symbols, whereby the tag $\mathcal{T}[\pi]$ is the π'th symbol in the tag tuple $\mathcal{T}$. For RSID codes, the RSID polynomials can be computed via increasing powers $g^{\pi i}$ of the RSID code's generator polynomial g. Specifically,

$$\mathcal{T}[\pi] = \sum_{i=0}^{k} u_i g^{\pi i}. \tag{61}$$

A detailed explanation of polynomials over finite fields is out of scope for this tutorial, and we refer to [23].

Rather than repeatedly multiplying polynomials g^π to determine their power $g^{\pi i}$, the g2p library offers a square-and-multiply method that improves the efficiency. To be even more efficient, instead of computing the power $g^{\pi i}$ of the underlying generator polynomial g from scratch in each iteration i, it is possible to compute the next generator power $g^{\pi(i+1)}$ at the end of each iteration i [132].

Further optimization is possible by avoiding computations that involve zero-padded symbols u_i if the input message $\boldsymbol{u}$ is smaller than the supported message size k. Since such computations always result in zero, the result can be determined immediately [132].

Since the result of a finite field operation (such as the addition or multiplication of two polynomials) can overflow the size of the finite field; generally, after each operation, a Barrett reduction (that is a form of modulo operation) "reduces" the result to be representable in the finite field. Further optimization of the computation of polynomials is possible by avoiding these computationally expensive Barrett reductions as much as possible [132]. Specifically, since the result of a polynomial addition cannot exceed the output memory of the CPU, it is not necessary to Barrett reduce after each addition of two polynomials. Instead, there is no limit on the number of polynomials that can be summed before the Barrett reduction. Only a single Barrett reduction after the summations is necessary to "return" the result into the finite field of 32 bit or 64 bit.

For multiplications (with the generator exponents $g^{\pi i}$), Barrett reductions are necessary immediately after two 64-bit polynomials (or three 32-bit polynomials) have been multiplied to avoid overflow. To reduce the computational complexity, the multiplication can be split into smaller subtasks. Specifically, the computation can follow an inner and an outer loop that correspond to two summations:

$$\mathcal{T}[\pi] = \sum_{i=0}^{k} u_i g^{\pi i} = \sum_{h=0}^{k/l} \left(\sum_{j=0}^{l} u_{(hl+j)} g^{\pi j} \right) g^{\pi h l}. \tag{62}$$

The input message $\boldsymbol{u}$ can be split into k/l fragments of length l. Then, in the inner loop, each symbol of index j in the message fragment is multiplied only with a *partial* generator exponent $g^{\pi j}$. By selecting a reasonable fragment length l of, e.g., 4096, it is possible to compute the partial generator

powers $g^{\pi j}, j = 0, \ldots, l-1$, only once and store them in a lookup table. Thereby, the Barrett reduction for the partial generator powers only needs to be computed once to store the result in the lookup table. Afterwards, the result can be retrieved from the table. Since the terms $u_{(hl+j)}g^{\pi j}$ are only a fragment of the field size q, the resulting product cannot overflow and does not require a Barrett reduction. Therefore, the operation in the inner loop requires only a single CLMU instruction [132].

The summation $\sum_{j=0}^{l} u_{(hl+j)}g^{\pi j}$ in the inner loop cannot overflow because summations never overflow. Therefore, only after summing up the l multiplications of the inner loop, a Barrett reduction is necessary for multiplication with the missing part $g^{\pi hl}$ of the generator power (in the outer loop). With this optimization, tag encoding is possible at 1.5GB/s for 32 bit, and at 3GB/s for 64 bit [132]. While [132] proposes these methods for RSID based tagging codes, [135] applies them to Reed-Muller based tagging codes.

H. Summary

Tagging codes define explicit code constructions to create noiseless-ID codes with tractable reliability guarantees. To this end, tagging codes repurpose linear block codes. The Hamming distance between the codewords of linear block codes is an established property. A tagging encoder maps the message $\boldsymbol{a}$ of the sender to a linear block codeword. From this linear block codeword, the encoder randomly selects one symbol that is the tag. The randomly selected position π and the value t of the selected symbol (i.e., the tag t) together constitute the tagging codeword $\boldsymbol{c}$, cf. Fig. 14. Because the Hamming distance of the underlying linear block code is known, the distance between the sender's linear block codeword (that constitutes the sender's noiseless-ID codeword subset $\mathcal{C}_{\boldsymbol{a}}$) and the receiver's linear block codeword (that constitutes the receiver's noiseless-ID codeword subset $\mathcal{C}_{\boldsymbol{b}}$) is known. Thereby, a high-distance linear block code (such as a Reed-Solomon code) provides guarantees about the overlap of the codeword subsets of message pairs as is required of a noiseless-ID code. For verification, the receiver computes the tag value at the random position π that the sender selected. If the tag values match, then the receiver concludes that both messages match.

Tagging codes are the state-of-the-art noiseless-ID codes. Selecting a specific tagging code depends on the use case as the choice of the employed code construction (such as Reed-Solomon, Reed-Muller, or Random Linear codes) and its respective parameters changes the performance, significantly. Specifically, RSID and RS2ID codes (based on Reed-Solomon codes) have the lowest worst-case collision probability bound. RS2ID codes are typically considered state-of-the-art tagging codes because they asymptotically achieve the ID capacity even for finite field sizes q. However, RS2ID codes have a higher coding complexity than RMID codes (based on Reed-Muller codes). RMID codes achieve the ID capacity only if the field size q also grows asymptotically towards infinity and RMID codes require larger tag sizes to achieve the same worst-case collision probability guarantees as RS2ID codes. However, RMID codes benefit from lower coding complexity. RLID codes rely on large field sizes q to achieve high-distance properties that govern the collision probability guarantees, whereby large field sizes q typically increase the computational complexity. However, RLID codes offer greater flexibility in contrast to the relatively rigid RS and RM based ID codes. The requirements on the worst-case collision probability can be softened such that the noiseless-ID code offers only "many 9s" in terms of the quantiles of collision probabilities (in contrast to the strict 100% guarantee of the worst-case collision probability bound λ_{fp}). For the "many 9s" case, no clear state of the art is established.

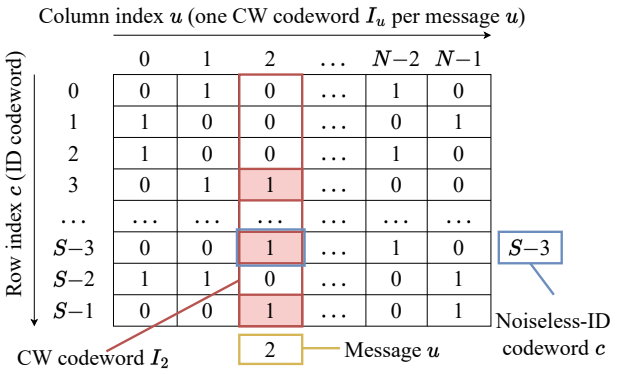


Fig. 15. The overall constant-weight code matrix can be interpreted as an incidence matrix between messages u and noiseless-ID codewords c. Selecting a random codeword corresponds to selecting a position at which the constant-weight codeword $I_{\boldsymbol{u}}$ holds a 1.

VIII. Constant-Weight Codes

Constant-weight codes (CWCs) are a variation of tagging codes and can form noiseless-ID codes. The first explicit noiseless-ID code construction (RS2ID, as proposed by [21]) framed the RS2ID code as a CWC. Until now, all ID-CWCs are binary, and concatenations of tagging codes with a (binary) CWC extension. Since the code rate R_{id}, the collision probability $p_{\text{fp}}(\boldsymbol{a}, \boldsymbol{b})$, and the collision probability bound λ_{fp} are chiefly determined by the underlying tagging code [94], the CWC extension is mainly a different representation of the same encoding principle. A complete explanation of ID-CWCs is out of the scope of this tutorial as the principles of these codes are already covered by the explanation of noiseless-ID codes in Section VI and tagging codes in Section VII. For a more detailed explanation of ID-CWCs, we refer to [94].

CWCs represent messages $\boldsymbol{u}$ and codewords $\boldsymbol{c}$ in scalar form in contrast to the vector representation that ID codes otherwise employ. Therefore, for this Section VIII, we typeset messages u and codewords c in non-bold font if they explicitly relate to the CWC representation of ID coding.

A. Encoding

CWCs map each message u to a constant-weight codeword (CW codeword) I_u, as Fig. 15 visualizes. A CWC supports a number N of messages, whereby the number N is called the dimension N of the CWC. A CW codeword I_u is the

TABLE X
CONSTANT-WEIGHT CODE CHARACTERISTICS AND CORRESPONDING NOISELESS-ID CODE CHARACTERISTICS.

CWC Characteristic	Corresponding ID Code Characteristic
CWC block length S	Number $S = \lvert\mathcal{C}\rvert$ of noiseless-ID codewords in set $\mathcal{C}$ of all noiseless-ID codewords
CWC dimension N	Number $N = \lvert\mathcal{U}\rvert$ of messages
CWC Hamming Weight M	Number $M = \lvert\mathcal{C}_{\boldsymbol{u}}\rvert$ of noiseless-ID codewords in subset $\mathcal{C}_{\boldsymbol{u}}$ of message $\boldsymbol{u}$
Upper Bound κ on CW-Codeword Overlap	Maximum number $(M - \delta)$ of overlapping noiseless-ID codewords

indicator vector for the subset $\mathcal{C}_u$ of noiseless-ID codewords that are associated with message u. In other words, for each noiseless-ID codeword c in the set $\mathcal{C}$ of all possible noiseless-ID codewords, the CW codeword I_u holds a 1 if the codeword belongs to the subset $\mathcal{C}_u$ of noiseless-ID codewords that are associated with message u, and 0 otherwise. Hence, a CW codeword I_u is a binary tuple of length $S = |\mathcal{C}| = q^m$. This length is referred to as the block length S of the CWC. Each CW codeword I_u holds $M = |\mathcal{C}_u|$ 1s, and $S - M$ 0s. This property gives CW codewords their name, as each CW codeword has a constant Hamming weight M, i.e., each CW codeword holds the same number of 1s.

The block length S of the CWC is not to be confused with the block length M of a noiseless-ID code. Rather, the block length S of the CWC corresponds to the size $|\mathcal{C}| = q^m$ of set $\mathcal{C}$ of all noiseless-ID codewords, cf. Table X.

The example in Fig. 15 shows an $(N \times S)$ matrix that holds the CW codewords for all N messages, whereby each column corresponds to one CW codeword. The message $u = 2$ (in yellow) is encoded into its corresponding CW codeword I_2 (in red). Each CW codeword I_u is binary.

From the CW codeword I_u, the sender selects a random noiseless-ID codeword c (sometimes called a *cue*) where the CW codeword I_u holds a 1. This corresponds to selecting a random position π in tagging codes. Recall that the random position π is selected from $\{0, \ldots, M-1\}$, with the length M of the tag tuple $\mathcal{T}_{\boldsymbol{u}}$. In CWCs, the random noiseless-ID codeword c is selected from one of the M positions for which the CW codeword I_u takes on a value of 1. The random sampling step in tagging codes and CWCs is equivalent, i.e., only the representation differs.

In the example in Fig. 15, there are $M = 3$ positions to choose from in the CW codeword I_2 of message $u = 2$. The sender randomly selects one of them (highlighted in blue), specifically $c = S - 3$. The noiseless-ID codeword $c = S - 3$ is a "hash" of the message $u = 2$. Next, the sender transmits the noiseless-ID codeword c via the noiseless channel $\boldsymbol{V}$ to the receiver.

B. Equivalence to Tagging Code Representation

The CWC representation and the tagging code representation can be equivalent if the CWC is constructed using a one-hot encoding (also referred to as pulse-position modulation PPM) of the tag tuple $\mathcal{T}$ created by a tagging code. For the PPM extension, a binary indicator vector of Hamming weight 1 sets the position in the vector that corresponds to the tag value to 1. In other words, the indicator vector must have a length q to be able to represent the q-ary tag value t by setting the t'th position of the indicator vector to 1.

In this case, the CW codeword I_u corresponds in size to the set $\mathcal{C}$ of all noiseless-ID codewords, cf. Eq. (51). The number of 1s in the CW codeword I_u, i.e., the weight M, equals the number M of tags in each tag tuple $\mathcal{T}$. Therefore, the CW codeword I_u can be understood as an indicator vector that states which (position, tag) tuples in the set $\mathcal{C}$ of all noiseless-ID codewords belong to the message u. By selecting a random position in the CW codeword I_u that equals 1, the encoder selects a (position, tag) tuple $(\pi, \mathcal{T}[\pi])$ that is one of the noiseless-ID codewords of the message u. Specifically, the noiseless-ID codeword c corresponds to the (position, tag) tuple $(\pi, \mathcal{T}[\pi])$ according to

$$c = \pi q + \mathcal{T}[\pi] = \pi q + t. \tag{63}$$

Thereby, c of CWCs is a different representation of the tagging codeword $\boldsymbol{c} = (\pi, \mathcal{T}[\pi])$.

C. Verification

To determine the equality estimate $\hat{v}$, the receiver uses the received noiseless-ID codeword c (that the sender randomly selected). For its own message b, the receiver determines the CW codeword I_b. Then, the receiver checks whether the CW codeword I_b holds a 1 at the received noiseless-ID codeword c. Thereby, the CWC verification rule is

$$\hat{v} = \begin{cases} \hat{v}_{\mathrm{p}} \text{ if } I_b[c] = 1, \\ \hat{v}_{\mathrm{n}} \text{ if } I_b[c] = 0. \end{cases} \tag{64}$$

In the example in Fig. 15, if the message b of the receiver is $b = 0, 1$, or $N-1$ (represented by the column indices), then the equality estimate $\hat{v}$ is a true negative estimate. Because the CW codeword of message $u = N - 2$ coincides with the CW codeword of message $u = 2$, if the receiver has message $b = N - 2$, then the estimate $\hat{v}$ is a false positive estimate.

D. Error Probability Guarantees

The overlap of noiseless-ID codeword subsets $\mathcal{C}$ corresponds to the number of noiseless-ID codewords c in that a pair (I_a, I_b) of CW codewords shares a value of 1. For the example in Fig. 15, the CW codewords of the message pair $(a = 1, b = 2)$ overlap in the noiseless-ID codeword $c = 3$. The upper bound κ for the pairwise overlap K of CW codewords defines the maximum number of noiseless-ID codeword c in that any pair of CW codewords overlaps. The bound κ on the overlap corresponds to the overlap $M - \delta$ in noiseless-ID codes (and tagging codes), cf. Table X. If the noiseless-ID codeword c is uniformly randomly selected from

the M positions set to 1 in the CW codeword I, then the FP error probability bound λ_{fp} of the corresponding CWC is

$$\lambda_{\text{fp}} = \frac{\kappa}{M}. \tag{65}$$

E. Summary

Constant-weight codes (CWCs) are a variation of tagging codes and can thereby be used as noiseless-ID codes. All proposed CWCs utilize the high distance properties of linear block codes to provide guarantees about the overlap between the codeword subsets $\mathcal{C}_u$ of different messages u. CWCs change the representation that tagging codes take in their repurposing of linear block codes. Specifically, each codeword subset $\mathcal{C}_u$ is represented as a binary constant-weight codeword I_u. Each noiseless-ID codeword c that "belongs" to message u is represented as a "1" in the constant-weight codeword I_u and all noiseless-ID codewords that do not "belong" to message u are represented with a "0" entry in their respective position in the CW codeword I_u. Because the constant-weight codewords I_u of different messages u all have the same number of noiseless-ID codewords c, the constant-weight codewords I_u have a "constant weight", i.e., a fixed number of "1"s. The overall CWC guarantees that any pair of constant-weight codewords I_u shares has only a limited number of positions in which both constant-weight codewords have a "1" value. The literature proposed several variations to the way a CWC is constructed from its underlying tagging code. However, the properties of the resulting noiseless-ID code in terms of its efficiency and its reliability are governed by the respective underlying tagging code for all proposed ID-CWCs. Thereby, all existing ID-CWCs can be considered to be a variation in the representation of tagging codes. Generally, using an ID-CWC would entail an additional coding overhead to change the representation from the underlying tagging code without always yielding a benefit. Thereby, in practice, tagging codes are currently the preferred noiseless-ID codes.

IX. Concatenated Noisy-ID Codes

This section explains the interaction of noiseless-ID codes with block codes to form concatenated noisy-ID codes that address the problem of ID via noisy channels.

A. Encoding

When concatenated, a block code and a randomized noiseless-ID code can form a randomized noisy-ID code that Section IV explained. We visualize an example of the encoding and verification steps of a noisy-ID code in Fig. 16. For visual clarity, Fig. 16 only displays the encoding and verifier subsets for the messages $\boldsymbol{a}$ and $\boldsymbol{b}$ of the sender and the receiver instead of displaying the subsets for all messages $\boldsymbol{u} \in \mathcal{U}$.

After the sender selects its message $\boldsymbol{a}$ from the set $\mathcal{U}$ of all messages, the sender determines the noiseless-ID codeword subset $\mathcal{C}_{\boldsymbol{a}}$ associated with the sender's message $\boldsymbol{a}$. In the example in Fig. 16, the sender selects message $\boldsymbol{a} = \boldsymbol{u}_9$ and determines the subset $\mathcal{C}_9$ that is associated with message $\boldsymbol{u}_9$. The receiver also selects a message $\boldsymbol{b}$ from the set $\mathcal{U}$ of all

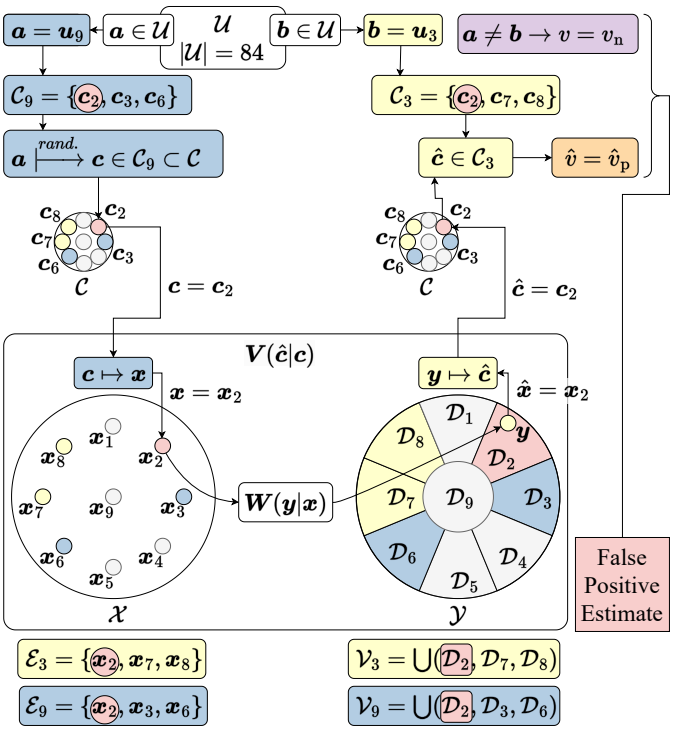


Fig. 16. Example for the encoding and verification of a concatenated noisy-ID code consisting of a block code and a noiseless-ID code. This figure illustrates a practical implementation of the fundamental principles visualized in Fig. 8. Codeword subset indices refer to the associated message $\boldsymbol{u}$, e.g., $\mathcal{C}_9$, $\mathcal{X}_9$, and $\mathcal{Y}_9$ are associated with message $\boldsymbol{u}_9$. All other numerical indices refer to the associated noiseless-ID codeword index, e.g., $\boldsymbol{x_2}$ and $\mathcal{D}_2$ are associated with noiseless-ID codeword $\boldsymbol{c_2}$. Note that the noisy-ID codeword sets $\mathcal{X}$ and $\mathcal{Y}$ have a larger cardinality than the noiseless ID codeword set $\mathcal{C}$, cf. Fig. 2.

messages and determines the noiseless-ID codeword subset $\mathcal{C}_{\boldsymbol{b}}$ associated with the receiver's message $\boldsymbol{b}$.

In the example in Fig. 16, the noiseless-ID codeword subsets that the sender and the receiver determined for their respective messages both hold three noiseless-ID codewords $\boldsymbol{c}$. The subsets $\mathcal{C}_9$ and $\mathcal{C}_3$ both share the noiseless-ID codeword $\boldsymbol{c_2}$. The overlap of noiseless-ID codeword subsets is necessary because the size N of the set $\mathcal{U}$ of messages is much larger than the size $S = |\mathcal{C}|$ of the noiseless-ID codeword set $\mathcal{C}$. In other words, as long as the overlap is not too large it is acceptable.

Every noiseless-ID codeword $\boldsymbol{c}$ in the set $\mathcal{C}$ of noiseless-ID codewords is associated with one or more messages. In other words, the noiseless-ID codebook and the set $\mathcal{C}$ of noiseless-ID codewords are identical in contrast to block codes and noisy-ID codes that both have codebooks that are only a subset of the overall set of codewords.

In the next step, the sender selects a random noiseless-ID codeword $\boldsymbol{c}$ from its noiseless-ID codeword subset $\mathcal{C}_{\boldsymbol{a}}$. If the noiseless-ID code is a tagging code, then the sender selects the tagging codeword $\boldsymbol{c} = (\pi, t_{\boldsymbol{a}})$. In the example in Fig. 16, the sender selects the noiseless-ID codeword $\boldsymbol{c_2}$ from their subset $\mathcal{C}_9$, whereby the selected noiseless-ID codeword $\boldsymbol{c_2}$ is also an element of the receiver's subset $\mathcal{C}_3$. Thereby, if the receiver in the example correctly receives the selected

noiseless-ID codeword $\boldsymbol{c_2}$, then the receiver will conclude a false positive equality estimate $\hat{v}_{\text{fp}}$. In other words, in the example, the selected noiseless-ID codeword $\boldsymbol{c_2}$ is an FP noiseless-ID codeword $\boldsymbol{c}_{\text{fp}}$, i.e., a noiseless-ID codeword that causes an FP verdict $\hat{v}_{\text{fp}}$ for the given message pair $(\boldsymbol{a}, \boldsymbol{b})$. The sender does not know which message the receiver selected and can neither know nor avoid that the selected noiseless-ID codeword $\boldsymbol{c_2}$ will lead to an FP estimate $\hat{v}_{\text{fp}}$. In the example, the probability of selecting a noiseless-ID codeword that leads to an FP estimate $\hat{v}_{\text{fp}}$ is $1/3$ because the sender selects one noiseless-ID codeword from the subset of size $M = 3$. Practical noiseless-ID codes typically support significantly larger noiseless-ID codeword subsets and smaller FP error probabilities.

Next, the block encoder further encodes the noiseless-ID codeword $\boldsymbol{c}$ (that represents the message $\boldsymbol{a}$ of the sender) into the overall noisy-ID codeword $\boldsymbol{x}$. The block encoder adds redundancy to the noiseless-ID codeword $\boldsymbol{c}$ as Section III-B and Section V-C explained. Note that the block code does not encode the message $\boldsymbol{a}$ but the noiseless-ID codeword $\boldsymbol{c}$. In the example in Fig. 16, the block encoder maps the noiseless-ID codeword $\boldsymbol{c_2}$ to the block codeword $\boldsymbol{x_2}$ (that is also the overall concatenated noisy-ID codeword). The size $|\mathcal{X}|$ of the sender's noisy-ID codeword set $\mathcal{X}$ is larger than the size $S = |\mathcal{C}|$ of the noiseless-ID codeword set $\mathcal{C}$, cf. Fig. 2. This is because the block code adds redundancy to the noiseless-ID codeword $\boldsymbol{c}$ to form the noisy-ID codeword $\boldsymbol{x}$ of the sender, cf. Fig. 11.

Each noiseless-ID codeword $\boldsymbol{c}$ has exactly one associated noisy-ID codeword $\boldsymbol{x_c}$. This corresponds to the one-to-one mapping between a message and its block codeword explained in Section III-B. Therefore, the size of the block codebook $\mathcal{B}$ matches the size S of the set $\mathcal{C}$ of noiseless-ID codewords. In contrast, the block codebook in message transmission matches the size $N > S$ of the set of messages. The smaller codebook is a result of the efficiency gain that the noiseless-ID code provides. The block codebook $\mathcal{B}$ is equal to the concatenated noisy-ID codebook $\mathcal{B}$ that Section IV-B1 explained.

The noisy-ID codeword subset $\mathcal{E}_{\boldsymbol{a}}$ associated with message $\boldsymbol{a}$ consists of the noisy-ID codewords $\boldsymbol{x}$ that are associated with the noiseless-ID codewords $\boldsymbol{c}$ that are an element of the subset $\mathcal{C}_{\boldsymbol{a}}$ of noiseless-ID codewords associated with message $\boldsymbol{a}$. In the example in Fig. 16, the noisy-ID codeword subset $\mathcal{E}_9$ of the sender includes those three noisy-ID codewords $\{\boldsymbol{x_2}, \boldsymbol{x_3}, \boldsymbol{x_6}\}$ that are the block encoded noiseless-ID codewords $\{\boldsymbol{c_2}, \boldsymbol{c_3}, \boldsymbol{c_6}\}$ from the subset $\mathcal{C}_9$. Because the noiseless-ID codeword subsets $\mathcal{C}_{\boldsymbol{u}}$ of the sender and the receiver share the noiseless-ID codeword $\boldsymbol{c_2}$ in the example, the noisy-ID codeword subsets $\mathcal{X}_{\boldsymbol{u}}$ overlap in the noisy-ID codeword $\boldsymbol{x_2}$. The size $M = |\mathcal{C}_{\boldsymbol{u}}|$ of each noiseless-ID codeword subset $\mathcal{C}_{\boldsymbol{u}}$ matches the size $M = |\mathcal{E}_{\boldsymbol{u}}|$ of the noisy-ID encoding subset because of the one-to-one mapping.

B. Verification

The sender transmits the selected noisy-ID codeword $\boldsymbol{x}$ via the DMC $\boldsymbol{W}(\boldsymbol{y}|\boldsymbol{x})$ to the receiver. The block decoder at the receiver determines from the received codeword $\boldsymbol{y}$ a noiseless-ID codeword estimate $\hat{\boldsymbol{c}}$. For that, the block decoder partitions the receiver's noisy-ID codeword set $\mathcal{Y}$ into S block decoding subsets $\mathcal{D}$ that correspond to their respective block codeword. For message $\boldsymbol{u}$, there are M block decoding subsets $\mathcal{D}$ that are associated with message $\boldsymbol{u}$ in correspondence to the M block codewords that form the associated noisy-ID encoding subset $\mathcal{E}_{\boldsymbol{u}}$. The union of all block decoding subsets $\mathcal{D}$ associated with the message $\boldsymbol{u}$ constitutes the overall noisy-ID verification subset $\mathcal{V}_{\boldsymbol{u}}$ of message $\boldsymbol{u}$ [1]. The size $|\mathcal{V}_{\boldsymbol{u}}|$ of the noisy-ID verification subset $\mathcal{V}_{\boldsymbol{u}}$ is significantly larger than the size $M = |\mathcal{E}_{\boldsymbol{u}}|$ of the noisy-ID encoding subset $\mathcal{E}_{\boldsymbol{u}}$ to compensate the channel distortion, cf. Section IV-C.

In the example in Fig. 16, the three block decoding subsets $\{\mathcal{D}_2, \mathcal{D}_7, \mathcal{D}_8\}$ constitute the noisy-ID verification subset $\mathcal{V}_{\boldsymbol{b}}$ associated with message $\boldsymbol{b} = \boldsymbol{u}_3$ of the receiver. The indices of the block decoding subsets align with the elements of the noiseless-ID codeword subset $\mathcal{C}_3 = \{\boldsymbol{c_2}, \boldsymbol{c_7}, \boldsymbol{c_8}\}$, and of the noisy-ID encoding subset $\mathcal{E}_3 = \{\boldsymbol{x_2}, \boldsymbol{x_7}, \boldsymbol{x_8}\}$ of the receiver's message $\boldsymbol{u}_3$. Because the noiseless-ID codeword subsets of the sender and the receiver overlap in noiseless-ID codeword $\boldsymbol{c_2}$, the noisy-ID verification subsets $\mathcal{V}_{\boldsymbol{u}}$ of the sender and the receiver overlap in block decoding subset $\mathcal{D}_2$.

The distortion of the channel does not exceed a level that the block code can mitigate. Hence, the received codeword $\boldsymbol{y}$ is an element of the block decoding subset $\mathcal{D}_2$ and the block decoder correctly maps the received noisy-ID codeword $\boldsymbol{y}$ to the noiseless-ID codeword estimate $\hat{\boldsymbol{c}} = \boldsymbol{c_2}$. If the noiseless-ID code is a tagging code, then the block decoder recovers an estimate $\hat{\boldsymbol{c}} = (\hat{\pi}, \hat{t}_{\boldsymbol{a}})$ of the tagging codeword $\boldsymbol{c}$ from the received noisy-ID codeword $\boldsymbol{y}$.

For the verification step, the receiver determines whether the noiseless-ID codeword estimate $\hat{\boldsymbol{c}}$ is an element of the noiseless-ID codeword subset $\mathcal{C}_{\boldsymbol{b}}$ associated with the receiver's message $\boldsymbol{b}$. If the noiseless-ID code is a tagging code, then the receiver extracts the estimated position $\hat{\pi}$ from the received estimated tagging codeword $\hat{\boldsymbol{c}} = (\hat{\pi}, \hat{t}_{\boldsymbol{a}})$. The receiver determines the tag $t_{\boldsymbol{b}}$ via selecting the symbol at position $\hat{\pi}$ of the tag tuple $\mathcal{T}_{\boldsymbol{b}}$. For the verification, the receiver then compares the tag $t_{\boldsymbol{b}} = \mathcal{T}_{\boldsymbol{b}}[\hat{\pi}]$ of the message of the receiver with the received estimated tag $\hat{t}_{\boldsymbol{a}}$ that the receiver extracted from the received estimated tagging codeword $\hat{\boldsymbol{c}} = (\hat{\pi}, \hat{t}_{\boldsymbol{a}})$.

In the example in Fig. 16, the noiseless-ID codeword estimate $\hat{\boldsymbol{c}} = \boldsymbol{c_2}$ is part of the noiseless-ID codeword subset $\mathcal{C}_3$ and the receiver erroneously determines a positive equality estimate $\hat{v}_{\text{p}}$, i.e., an FP estimate $\hat{v}_{\text{fp}}$. Note that the erroneous estimate is not a result of distortion or an unsuited block code, but rather a result of the information loss in the noiseless-ID encoding, i.e., a result of the overlap in the encoding noiseless-ID codeword subsets $\mathcal{C}_{\boldsymbol{u}}$.

C. Discussion

The abstract visualization of randomized noisy-ID coding in Figs. 8 and 9 implies joint randomized noisy-ID coding and illustrates the codeword subsets in a simplified manner. For concatenated noisy-ID codes, the visualization in Fig. 16 elaborates the abstract visualization of the FP case in Fig. 9b and clarifies possible misconceptions by avoiding a significant simplification: neither an encoding subset $\mathcal{E}_{\boldsymbol{a}}$ nor a verifier

subset $\mathcal{V}_{\boldsymbol{b}}$ of noiseless-ID codewords is necessarily a connected subset of neighboring noiseless-ID codewords as visually implied in Fig. 9b. Rather, the subset can also be the union of an arbitrary number of non-adjacent codewords in the codeword sets $\mathcal{X}$ and $\mathcal{Y}$. Furthermore, the overlap between the codeword subsets of two messages does not necessarily consist of adjacent codewords. Rather, the codewords that constitute the overlap of the codeword subsets of two messages can have an arbitrary distance between each other within the set of all codewords.

Figure 16 visualizes how a randomized noisy-ID code increases the number of identifiable messages compared to linear block codes. The block code only provides nine noisy-ID codewords $\boldsymbol{x_1}, \ldots, \boldsymbol{x_9}$ for transmission of information, i.e., $S = 9$. However, the sender and the receiver are able to identify $N = |\mathcal{U}| = 84$ different messages. Since the overall noisy-ID code supports a larger number $N = 84$ of messages than the number $S = 9$ of codewords that the block code provides, it is not possible to assign each message $\boldsymbol{u}$ an unambiguous codeword $\boldsymbol{x}$. Rather, the messages have to share the limited number of codewords. For that purpose, the ID encoder maps each message $\boldsymbol{a}$ to a subset $\mathcal{C}_{\boldsymbol{a}}$, whereby $|\mathcal{C}_{\boldsymbol{a}}| = M = 3$. For a hypothetical subset size $M = 1$, each message $\boldsymbol{u}$ would be associated with a single codeword $\boldsymbol{x}$. However, this codeword $\boldsymbol{x}$ would also be a single codeword of other messages because the supply of codewords is too small to offer each message a unique codeword $\boldsymbol{x}$. This results in guaranteed FP estimates if a pair of messages is selected that shares the same codeword $\boldsymbol{x}$. For larger subset sizes $M > 1$, it is possible to create a randomized noisy-ID code that limits the probability of an FP estimate, because pairs of messages only share a fraction of the subset of their codewords with any other message at a time. For very large subset sizes $M \to \infty$ (which require very large noisy-ID codeword sizes $n \to \infty$ to support the number of noiseless-ID codewords), and for capacity-achieving noiseless-ID codes, the fraction of shared codewords with any other message goes towards zero, thereby approaching error-free identification in the infinite regime.

If the block code in the virtual noiseless channel $\boldsymbol{V}$ is not able to overcome the noisy channel $\boldsymbol{W}$ perfectly, then the receiver could obtain a false estimate $\hat{\boldsymbol{c}}$ of the tagging codeword $\boldsymbol{c}$. In that case, the estimated position $\hat{\pi}$ could be wrong, such that the receiver compares the received tag $\hat{t_{\boldsymbol{a}}}$ with a tag in a position $\hat{\pi} \neq \pi$, which can yield false equality estimates $\hat{v}$. Similarly, if the tag estimate $\hat{t}_{\boldsymbol{a}}$ is incorrect, then the receiver makes a comparison between the receiver's tag $t_{\boldsymbol{b}}$ and a different tag $\hat{t}_{\boldsymbol{a}} \neq t_{\boldsymbol{a}}$. For a detailed, complete breakdown of error types, cf. Section IV-D4 and Fig. 9.

The overall noisy-ID code can at best be as good (in terms of reaching the ID capacity of the noisy channel) as the employed block code rate approaches the transmission capacity of the noisy channel. A noiseless-ID code is optimal for ID if it achieves the ID capacity of the noiseless channel (that the block code "creates"). If the noiseless-ID code is optimal for ID, then the overall noisy-ID code is *exactly* (in contrast to *at best*) as good as the employed block code.

Since the two codes with their respective code rates constitute the overall noisy-ID code, the overall noisy-ID code rate is the noiseless-ID code rate Eq. (41) (with double exponential scaling) times the block code rate (with traditional exponential scaling) that Section V-C explained:

$$R_{\text{id, noisy}} = R_{\text{tx}} R_{\text{id}} = \frac{m}{n} \frac{\log_2(\log_2(N)}{m \log_2(q)} = \frac{\log_2(\log_2(N)}{n \log_2(q)}. \tag{66}$$

As [21] states, if the noiseless-ID code is "optimal for ID", i.e., reaches a noiseless-ID code rate $R = 1$, then the overall noisy-ID code is as good as the block code is (in terms of how close the block code rate approaches the transmission capacity). Note that Eq. (66) matches the noisy-ID code rate from Eq. (17).

D. Summary

By concatenation, a block code and a noiseless-ID code can form a (randomized) noisy-ID code. For efficiency, the noiseless-ID encoder maps the sender's message to a shorter noiseless-ID codeword $\boldsymbol{c}$, cf. Fig. 16. The linear block encoder adds redundancy to the noiseless-ID codeword by mapping the noiseless-ID codeword $\boldsymbol{c}$ to its corresponding linear block codeword $\boldsymbol{x}$ (that acts as the noisy-ID codeword). The linear block decoder recovers an estimate $\hat{\boldsymbol{c}}$ of the noiseless-ID codeword. Finally, the noiseless-ID verifier checks whether the received estimate $\hat{\boldsymbol{c}}$ belongs to the set $\mathcal{C}_{\boldsymbol{b}}$ of noiseless-ID codewords associated with the message $\boldsymbol{b}$ of the receiver. The verification set $\mathcal{V}_{\boldsymbol{b}}$ of the overall noisy-ID code is the union of the decoding sets $\mathcal{D}$ of block codewords that correspond to noiseless-ID codewords in the set $\mathcal{C}_{\boldsymbol{b}}$, cf. Fig. 16.

The current state of the art in concatenated noisy-ID coding lies in concatenating a tagging code, such as an RS2ID code, with a suitable linear block code, such as a Polar code. RS2ID codes can achieve the ID capacity of a noiseless channel and Polar codes can achieve the transmission capacity of a noisy channel. Thereby, their concatenation (a concatenated noisy-ID code) can achieve the ID capacity of a noisy channel.

X. Summary and Outlook

A. Summary

This tutorial explained in detail codes that address identification via channels efficiently and reliably. In ID, the receiver determines whether its own message $\boldsymbol{b}$ equals the message $\boldsymbol{a}$ of the sender, i.e., a binary yes-no answer. As this communication problem is easier than transmitting a message (that encompasses decoding *which* of N messages the sender selected), coding schemes that are up to exponentially more efficient in terms of the achievable code rates can be constructed. Noisy-ID codes are special-purpose codes that are tailored to the ID problem. To construct noisy-ID codes, it is possible to concatenate a linear block code that addresses the noise in the channel with a noiseless-ID code that "hashes" the message $\boldsymbol{a}$ of the sender into a noiseless-ID codeword. Despite the "hashing", the receiver is able to determine whether $\boldsymbol{a} = \boldsymbol{b}$ with high probability, i.e., reliably. Specifically, a noiseless-ID code guarantees a small upper bound λ_{fp} on the collision probability of the "hashes" of differing messages. Tagging

codes are a family of noiseless-ID codes that repurpose error correction or erasure codes, such as Reed-Solomon, Reed-Muller, or Random Linear codes, to provide reliability guarantees. While tagging codes are related to other hash functions, they are generally not cryptographic. State-of-the-art implementations of tagging codes achieve GB/s encoding speeds. Since the receiver only verifies equality, there is no need for a decoder.

To create state-of-the-art noisy-ID codes, a tagging code such as an RS2ID code is concatenated with a capacity-achieving channel code such as a Polar code. The result is a concatenated randomized noisy-ID code that can be up to exponentially more efficient than a capacity-achieving channel code on its own in addressing the ID problem via noisy channels. The specific parameters of the tagging and Polar code depend on the use case in order to fine-tune the metrics of the code to the number of supported messages, the required reliability guarantees, and the coding complexity, among other performance metrics.

The up-to-exponential efficiency gains that are possible in ID (compared to message transmission) increase even further when the sender and the receiver have access to a common source of randomness. This can be a random number generator.

B. Outlook

While the state-of-the-art theory behind ID coding promises significant advantages over existing coding techniques when addressing the ID problem, there remains a need for further verification of the expected gains in practice. The study [24] proposed a deterministic noisy-ID coding scheme that enables reliable ID via Gaussian channels with very low Signal-to-Noise Ratios (SNRs). Except for an initial investigation in [25], concatenated randomized noisy-ID codes have not been evaluated in practice, partly due to the former high computational cost of tagging encoding and verification that has since been reduced by orders of magnitude [132]. Randomized noisy-ID codes should aim to surpass the theoretically less powerful deterministic noisy-ID codes in practical investigations. Specifically, when addressing the ID problem via Gaussian channels with very low SNRs, a comparison with deterministic noisy-ID codes could highlight the suitability of randomized noisy-ID codes to enable reliable ID even under severe noise conditions. Reducing the signal power may save energy at the cost of operating at lower SNRs. This can be beneficial for battery-powered devices in particular but also to lower the overall power consumption of wireless communication [136], [137] when addressing the ID problem. This can benefit an envisioned use case that verifies the data integrity between sensors, robots, and their digital twins in complex extended reality settings [138]–[140] with minimal communication overhead. Investigations should not be limited to the low SNR regime but extend up to a very high SNR to investigate the potential trade-off between using energy to increase the signal power and using energy to use more reliable (but possibly computationally more complex) ID coding schemes that can mitigate the effects of a low SNR.

To date, tagging codes have mainly been investigated in a standalone manner without considering their interaction with a linear block code when forming concatenated (randomized) noisy-ID codes. Closer integration of tagging codes and linear block codes may show additional benefits, such as lower computational complexity. Such interaction between the two concatenated codes may be dynamic, and parameters may need to be adapted to changing channel properties, such as SNR. Investigating more complex channel models such as fading channels may be closer to the conditions that noisy-ID codes will need to address in real-world over-the-air use cases. For example, industrial scenarios with moving robots and multiple reflective metallic surfaces may create complex transmission conditions. As the complexity of the channel and the complexity of the channel code increase and thereby the interaction of the channel and the channel code with a tagging code grows more complex, employing machine learning methods may become necessary to find good trade-offs in a growing parameter space.

Current ID coding schemes do not take into account the statistical properties of the data source. Investigating ID source coding promises even higher efficiency gains compared to source-agnostic ID coding. Ultimately, this may lead to a noisy-ID code that is aware of the source, mitigates channel distortion, and optimizes the overall code for ID. One possible avenue of obtaining such a code may lie in using machine learning methods to learn patterns in the source data. The training could be performed offline on historical source data or the training could be part of a protocol that continuously adapts the ID code to newly observed source properties.

The broadening of the paradigm of ID to messages that differ at most by a prescribed number of bits (or distortion level) is also an interesting direction for future research. In the digital twin use case, this can make the receiver more "tolerant" as it verifies positively not only when the messages match exactly, but also if the difference between the two messages is below a certain threshold. This may be achieved by extending existing ID codes towards K-ID, whereby K is the number of receiver messages that the receiver is "comparing" to the sender's message. For pairwise ID which is the focus of this tutorial, $K = 1$. While K-ID is envisioned to allow for arbitrary sets of receiver messages, differing by a prescribed number of bits from a "central" message adds more structure to the communication goal that may be exploited to allow for more efficient coding schemes. In general, extending tagging codes to communication goals that are related to ID may benefit from machine learning methods that can automate the prototyping of codes. This could make it possible to scale the finding of a large number of codes that are each tailored to a specific communication goal.

This tutorial described the ID problem in point-to-point communication, i.e., for one sender-receiver pair. In addition to extending the number of identified messages from $K = 1$ to arbitrary values of K, following the principles of multicast communication [141], [142] and broadcast communication [143], the number of receivers could increase, whereby each receiver aims to identify one or more messages. Furthermore, increasing the number of senders could "distribute" the message $\boldsymbol{a}$ that the receiver(s) compare to their own message(s) $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_K$ among several sender nodes that each

only have one part of the overall message $\boldsymbol{a}$. Also, following the paradigm of in-network recoding [144], [145], intermediate nodes can compute ID codewords based on one or more ID codewords they received. Random Linear Network Coding (RLNC) benefits such generalized network setups for message transmission [146]–[150]. Random Linear ID (RLID) codes [130] may address the problem "Identification via Networks" [151] similar to RLNC addressing transmission via networks.

ID coding can benefit significantly from Common Randomness (CR) whereby CR generation is a communication goal. Methods for efficiently generating secure CR via measuring, for example, phase noise in the transmission [113], can be combined with noisy-ID coding schemes to enable physical-layer secure ID. This is of particular interest because this type of security is information-theoretically proven and comes "for free", i.e., a randomized noisy-ID code does not need to spend additional bits to become secure from eavesdropping.

In addition to their role in forming noisy-ID codes, standalone tagging codes offer high reliability and high efficiency for ID problems over communication networks with virtually lossless links. Therefore, tagging codes can also be considered within the application layer. Tagging codes can offer strong reliability guarantees for applications that currently use hash functions or checksums that provide only mean collision probability guarantees. While the mean collision probability is often practically sufficient for hashes of 256 bit or larger, tagging codes are expected to enable reliable ID even for small hash sizes. One possible future direction lies in expanding on tagging codes with respect to adversarial attacks and security aspects towards cryptographic use cases, such as verifying data integrity, which is a critical component of a wide range of information security systems [66], [152]–[154].

ACKNOWLEDGMENTS

The authors thank Johannes Rosenberger for many helpful discussions about the definition of communication goals, reliability, and efficiency.

APPENDIX

Proof for Eq. (44): We begin with Eq. (42) by replacing the terms for the rate R_{id} and the error exponent E_{fp} with their definitions from Eq. (41) and Eq. (43), respectively; thus,

$$\frac{\log_2(\log_2(N)) - 2\log_2(\lambda_{\text{fp}})}{m'} \leq 1. \tag{67}$$

The noiseless-ID codeword size m' can be shifted to the other side of the inequality. With the logarithm laws, the factor 2 can be moved into the logarithm as an exponent, and the resulting squared collision probability bound λ_{fp}^2 can be moved into the denominator, i.e.,

$$\log_2\left(\frac{\log_2(N)}{\lambda_{\text{fp}}^2}\right) \leq m'. \tag{68}$$

Next, both sides can be used as the exponent for a power of 2, and the squared collision probability bound λ_{fp}^2 can be shifted to the other side of the inequality, i.e.,

$$\log_2(N) \leq 2^{m'}\lambda_{\text{fp}}^2. \tag{69}$$

Finally, with the definition of the binary message size k', we can reformulate to Eq. (44).

REFERENCES

[1] R. Ahlswede and G. Dueck, "Identification via channels," *IEEE Transactions on Information Theory*, vol. 35, no. 1, pp. 15–29, 1989.
[2] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.
[3] W. Weaver, "Recent contributions to the mathematical theory of communication," *ETC: A Review of General Semantics*, vol. 10, no. 4, pp. 261–281, Summer 1953.
[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proceedings IEEE ICC*, vol. 2, 1993, pp. 1064–1070.
[5] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.
[6] 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.
[7] K. Arora, J. Singh, and Y. S. Randhawa, "A survey on channel coding techniques for 5G wireless networks," *Telecommunication Systems*, vol. 73, no. 4, pp. 637–663, 2020.
[8] V. Bioglio, C. Condo, and I. Land, "Design of polar codes in 5G new radio," *IEEE Commun, Surv. & Tut.*, vol. 23, no. 1, pp. 29–40, 2021.
[9] N. Bonello, S. Chen, and L. Hanzo, "Low-Density Parity-Check codes and their rateless relatives," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 1, pp. 3–26, 2011.
[10] M. F. Brejza, L. Li, R. G. Maunder, B. M. Al-Hashimi, C. Berrou, and L. Hanzo, "20 years of Turbo coding and energy-aware design guidelines for energy-constrained wireless applications," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 8–28, 2016.
[11] Z. B. Kaykac Egilmez, L. Xiang, R. G. Maunder, and L. Hanzo, "The development, operation and performance of the 5G Polar codes," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 96–122, 2020.
[12] H. Mukhtar, A. Al-Dweik, and A. Shami, "Turbo product codes: Applications, challenges, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 3052–3069, 2016.
[13] S. Shao *et al.*, "Survey of Turbo, LDPC, and Polar decoder ASIC implementations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2309–2333, 2019.
[14] M. Vaezi *et al.*, "Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road towards 6G," *IEEE Commun. Surv. & Tut.*, vol. 24, no. 2, pp. 1117–1174, 2022.
[15] O. Goldreich, B. Juba, and M. Sudan, "A theory of goal-oriented communication," *Journal of the ACM*, vol. 59, no. 2, pp. 1–65, 2012.
[16] I. Csiszár and P. Narayan, "Common randomness and secret key generation with a helper," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 344–366, 2000.
[17] V. Doshi, D. Shah, M. Médard, and M. Effros, "Functional compression through graph coloring," *IEEE Transactions on Information Theory*, vol. 56, no. 8, pp. 3901–3917, 2010.
[18] H. Boche and C. Deppe, "Secure identification for wiretap channels; robustness, super-additivity and continuity," *IEEE Trans. on Information Forensics and Security*, vol. 13, no. 7, pp. 1641–1655, Jul. 2018.
[19] H. Boche, C. Deppe, and A. Winter, "Secure and robust identification via classical-quantum channels," *IEEE Transactions on Information Theory*, vol. 65, no. 10, pp. 6734–6749, Oct. 2019.
[20] H. Boche and C. Deppe, "Secure identification under passive eavesdroppers and active jamming attacks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 472–485, Feb. 2019.
[21] S. Verdú and V. K. Wei, "Explicit construction of optimal constant-weight codes for identification via channels," *IEEE Transactions on Information Theory*, vol. 39, no. 1, pp. 30–36, 1993.
[22] R. Roth, *Introduction to coding theory*. Cambridge University Press, 2006.
[23] E. R. Berlekamp, *Algebraic Coding Theory (Revised Edition)*. World Scientific, Singapore, 2015.
[24] I. Vorobyev, C. Deppe, L. Torres-Figueroa, and H. Boche, "Deterministic identification: From theoretical analysis to practical identification codes," in *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, 2024, pp. 303–308.
[25] C. von Lengerke, J. A. Cabrera, and F. H. Fitzek, "Identification codes for increased reliability in digital twin applications over noisy channels," in *Proc. IEEE Int. Conf. on Metaverse Comp., Netw. and Appl. (MetaCom)*, 2023, pp. 550–557.

[26] S. Derebeyoğlu, C. Deppe, and R. Ferrara, "Performance analysis of identification codes," *Entropy*, vol. 22, no. 10, p. Art. no. 1067, 2020.

[27] J. Ja Ja, "Identification is easier than decoding," in *Proc. Ann. Symp. on Foundations of Computer Science (SFCS)*, 1985, pp. 43–50.

[28] R. Ahlswede and Z. Zhang, "New directions in the theory of identification via channels," *IEEE Transactions on Information Theory*, vol. 41, no. 4, pp. 1040–1050, 1995.

[29] R. Ahlswede, "Identification via channels," in *Identification and Other Probabilistic Models*. Springer, Cham, Switzerland, 2021, pp. 3–43.

[30] J. L. Carter and M. N. Wegman, "Universal classes of hash functions," in *Proceedings of the ninth annual ACM symposium on Theory of computing*, 1977, pp. 106–112.

[31] C. Chaccour, C. Kurisummoottil Thomas, W. Saad, and M. Debbah, "Introduction to semantic communications," in *Foundations of Semantic Communication Networks*. John Wiley & Sons, Ltd., 2025, pp. 1–18.

[32] C. Chaccour, W. Saad, M. Debbah, Z. Han, and H. V. Poor, "Less data, more knowledge: Building next generation semantic communication networks," *IEEE Commun. Surv. & Tut.*, vol. 27, no. 1, pp. 37–76, 2025.

[33] T. M. Getu, G. Kaddoum, and M. Bennis, "Semantic communication: A survey on research landscape, challenges, and future directions," *Proceedings of the IEEE*, vol. 112, no. 11, pp. 1649–1685, 2024.

[34] D. Gündüz *et al.*, "Beyond transmitting bits: Context, semantics, and task-oriented communications," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 5–41, 2022.

[35] H. Lee, H. Ahn, and Y. D. Park, "Performance analysis of coexistence of traditional communication system and emerging semantic communication system," *ICT Express*, vol. 9, no. 3, pp. 420–426, 2023.

[36] Y. Liu, X. Wang, Z. Ning, M. Zhou, L. Guo, and B. Jedari, "A survey on semantic communications: Technologies, solutions, applications and challenges," *Digital Communications and Networks*, vol. 10, no. 3, pp. 528–545, 2024.

[37] Z. Lu *et al.*, "Semantics-empowered communications: A tutorial-cum-survey," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 1, pp. 41–79, 2024.

[38] A. Maatouk, M. Assaad, and A. Ephremides, "The age of incorrect information: An enabler of semantics-empowered communication," *IEEE Trans. on Wireless Commun.*, vol. 22, no. 4, pp. 2621–2635, 2023.

[39] S. Raj Pandey, V. Phuc Bui, and P. Popovski, "Semantic and goal-oriented communication: A data valuation perspective," in *Foundations of Semantic Communication Networks*. John Wiley & Sons, Ltd, 2025, pp. 291–306.

[40] H. Seo, J. Park, M. Bennis, and M. Debbah, "Semantics-native communication via contextual reasoning," *IEEE Trans. on Cognitive Commun. and Netw.*, vol. 9, no. 3, pp. 604–617, 2023.

[41] S. Seo, J. Park, S.-W. Ko, J. Choi, M. Bennis, and S.-L. Kim, "Toward semantic communication protocols: A probabilistic logic perspective," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 8, pp. 2670–2686, 2023.

[42] Y. Shao, Q. Cao, and D. Gündüz, "A theory of semantic communication," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 12 211–12 228, 2024.

[43] G. Shi, Y. Xiao, Y. Li, and X. Xie, "From semantic communication to semantic-aware networking: Model, architecture, and open problems," *IEEE Communications Magazine*, vol. 59, no. 8, pp. 44–50, 2021.

[44] E. Uysal *et al.*, "Semantic communications in networked systems: A data significance perspective," *IEEE Network*, vol. 36, no. 4, p. 233–240, Jul. 2022.

[45] D. Wheeler and B. Natarajan, "Engineering semantic communication: A survey," *IEEE Access*, vol. 11, pp. 13 965–13 995, 2023.

[46] L. Xia, Y. Sun, C. Liang, L. Zhang, M. A. Imran, and D. Niyato, "Generative AI for semantic communication: Architecture, challenges, and outlook," *IEEE Wireless Communications*, vol. 32, no. 1, pp. 132–140, 2025.

[47] G. Xin, P. Fan, and K. B. Letaief, "Semantic communication: A survey of its theoretical development," *Entropy*, vol. 26, no. 2, p. Art. no. 102, 2024.

[48] Q. Zhao, H. Zou, M. Bennis, and M. Debbah, "Semantic communications," in *Artificial Intelligence for Future Networks*. John Wiley & Sons, Ltd, 2025, pp. 131–149.

[49] A. Alexiou, M. Debbah, M. Di Renzo, E. C. Strinati, and H. Viswanathan, "Guest editorial beyond shannon communications—A paradigm shift to catalyze 6G," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 8, pp. 2299–2305, 2023.

[50] M. Chafii, L. Bariah, S. Muhaidat, and M. Debbah, "Twelve scientific challenges for 6G: Rethinking the foundations of communications theory," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 868–904, 2023.

[51] G. Fernandes, H. Fontes, and R. Campos, "Semantic communications: The new paradigm behind beyond 5G technologies," *Preprint arXiv:2406.00754*, 2024.

[52] Y. Li, F. Zhou, L. Yuan, Q. Wu, and N. Al-Dhahir, "Cognitive semantic communication: A new communication paradigm for 6G," *IEEE Communications Magazine, in print*, pp. 1–8, 2025.

[53] Y. Lin, Z. Gao, H. Du, J. Wang, and J. Zheng, "Semantic communication in the metaverse," in *Wireless Semantic Communications*. John Wiley & Sons, Ltd, 2025, pp. 133–161.

[54] K. Lu *et al.*, "Rethinking modern communication from semantic coding to semantic communication," *IEEE Wireless Communications*, vol. 30, no. 1, pp. 158–164, 2023.

[55] S. Iyer *et al.*, "A survey on semantic communications for intelligent wireless networks," *Wireless Personal Communications*, vol. 129, no. 1, pp. 569–611, 2023.

[56] Y. E. Sagduyu, T. Erpek, A. Yener, and S. Ulukus, "Will 6G be semantic communications? opportunities and challenges from task oriented and secure communications to integrated sensing," *IEEE Network*, vol. 38, no. 6, pp. 72–80, 2024.

[57] M. Shokrnezhad, H. Mazandarani, T. Taleb, J. Song, and R. Li, "Semantic revolution from communications to orchestration for 6G: Challenges, enablers, and research directions," *IEEE Network*, vol. 38, no. 6, pp. 63–71, 2024.

[58] E. C. Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, p. Art. no. 107930, 2021.

[59] W. Tong and G. Y. Li, "Nine challenges in artificial intelligence and wireless communications for 6G," *IEEE Wireless Communications*, vol. 29, no. 4, pp. 140–145, 2022.

[60] Y. Wang, H. Han, Y. Feng, J. Zheng, and B. Zhang, "Semantic communication empowered 6G networks: Techniques, applications, and challenges," *IEEE Access*, vol. 13, pp. 28 293–28 314, 2025.

[61] Z. Wang, S. Leng, H. Zhang, and C. Yuen, "Deep semantic communication for knowledge sharing in internet of vehicles," *IEEE Internet of Things Journal, in print*, pp. 1–1, 2025.

[62] W. Xu, Z. Yang, D. W. K. Ng, O. A. Dobre, L.-C. Wang, and R. Schober, "Guest editorial: Task-oriented communications for future wireless networks," *IEEE Wireless Communications*, vol. 30, no. 3, pp. 16–17, 2023.

[63] P. Zhang *et al.*, "Intellicise wireless networks from semantic communications: A survey, research issues, and challenges," *IEEE Communications Surveys & Tutorials, in print*, pp. 1–1, 2025.

[64] H. Zhou, Y. Deng, X. Liu, N. Pappas, and A. Nallanathan, "Goal-oriented semantic communications for 6G networks," *IEEE Internet of Things Magazine*, vol. 7, no. 5, pp. 104–110, 2024.

[65] G. P. Fettweis and H. Boche, "On 6G and trustworthiness," *Communications of the ACM*, vol. 65, no. 4, pp. 48–49, 2022.

[66] S. Guo *et al.*, "A survey on semantic communication networks: Architecture, security, and privacy," *IEEE Communications Surveys & Tutorials, in print*, pp. 1–1, 2024.

[67] ——, "A survey on semantic communication networks: Architecture, security, and privacy," *IEEE Communications Surveys & Tutorials, in print*, pp. 1–1, 2025.

[68] R. Meng *et al.*, "A survey of secure semantic communications," *arXiv preprint arXiv:2501.00842*, 2025.

[69] D. Won *et al.*, "Resource management, security, and privacy issues in semantic communications: A survey," *IEEE Communications Surveys & Tutorials, in print*, pp. 1–1, 2025.

[70] Z. Yang, M. Chen, G. Li, Y. Yang, and Z. Zhang, "Secure semantic communications: Fundamentals and challenges," *IEEE Network*, vol. 38, no. 6, pp. 513–520, 2024.

[71] M. M. Hafidi, M. Djezzar, M. Hemam, F. Z. Amara, and M. Maimour, "Semantic web and machine learning techniques addressing semantic interoperability in Industry 4.0," *Int. Journal of Web Information Systems*, vol. 19, no. 3/4, pp. 157–172, 2023.

[72] S. Meng, S. Wu, J. Zhang, J. Cheng, H. Zhou, and Q. Zhang, "Semantics-empowered space-air-ground-sea integrated network: New paradigm, frameworks, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 1, pp. 140–183, 2025.

[73] Z. Kaleem, F. A. Orakzai, W. Ishaq, K. Latif, J. Zhao, and A. Jamalipour, "Emerging trends in UAVs: From placement, semantic communications to generative AI for mission-critical networks," *IEEE Transactions on Consumer Electronics, in print*, pp. 1–1, 2025.

[74] W. Yang *et al.*, "Semantic communications for future internet: Fundamentals, applications, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 213–250, 2023.

[75] ——, "Semantic communication meets edge intelligence," *IEEE Wireless Communications*, vol. 29, no. 5, pp. 28–35, 2022.

[76] M. Zhang, M. Abdi, V. R. Dasari, and F. Restuccia, "Semantic edge computing and semantic communications in 6G networks: A unifying survey and research challenges," *Preprint arXiv:2411.18199*, 2024.

[77] M. Bouazizi and T. Ohtsuki, "Multi-dimensional representation for semantic communication: A new horizon for customized visualization of shared knowledge," in *Proc. IEEE 100th Vehicular Technology Conf. (VTC-Fall)*, 2024, pp. 1–6.

[78] M. Kalfa, M. Gok, A. Atalik, B. Tegin, T. M. Duman, and O. Arikan, "Towards goal-oriented semantic signal processing: Applications and future challenges," *Dig. Sig. Proc.*, vol. 119, p. Art. no. 103134, 2021.

[79] J. Chen, C. Yi, S. D. Okegbile, J. Cai, and X. Shen, "Networking architecture and key supporting technologies for human digital twin in personalized healthcare: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 1, pp. 706–746, 2024.

[80] S. K. Jagatheesaperumal *et al.*, "Semantic-aware digital twin for metaverse: A comprehensive review," *IEEE Wireless Communications*, vol. 30, no. 4, pp. 38–46, 2023.

[81] K. Li, B. P. L. Lau, X. Yuan, W. Ni, M. Guizani, and C. Yuen, "Toward ubiquitous semantic metaverse: Challenges, approaches, and opportunities," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21 855–21 872, 2023.

[82] U. Khalid, M. S. Ulum, A. Farooq, T. Q. Duong, O. A. Dobre, and H. Shin, "Quantum semantic communications for metaverse: Principles and challenges," *IEEE Wireless Communications*, vol. 30, no. 4, pp. 26–36, 2023.

[83] Z. Wang, Y. Deng, and A. Hamid Aghvami, "Goal-oriented semantic communications for avatar-centric augmented reality," *IEEE Transactions on Communications*, vol. 72, no. 12, pp. 7982–7995, 2024.

[84] M.-D. Nguyen, Q.-V. Do, Z. Yang, Q.-V. Pham, and W.-J. Hwang, "Joint communication and computation framework for goal-oriented semantic communication with distortion rate resilience," *Preprint arXiv:2309.14587*, 2023.

[85] J. Pei, C. Feng, P. Wang, H. Tabassum, and D. Shi, "Latent diffusion model-enabled low-latency semantic communication in the presence of semantic ambiguities and wireless channel noises," *IEEE Transactions on Wireless Communications, in print*, pp. 1–1, 2025.

[86] E. Grassucci, Y. Mitsufuji, P. Zhang, and D. Comminiello, "Enhancing semantic communication with deep generative models: An overview," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 13 021–13 025.

[87] P. Jiang, C.-K. Wen, X. Yi, X. Li, S. Jin, and J. Zhang, "Semantic communications using foundation models: Design approaches and open issues," *IEEE Wireless Commun*, vol. 31, no. 3, pp. 76–84, 2024.

[88] C. Liang *et al.*, "Generative AI-driven semantic communication networks: Architecture, technologies and applications," *IEEE Transactions on Cognitive Communications and Networking*, vol. 11, no. 1, pp. 27–47, 2025.

[89] J. Ren *et al.*, "Generative semantic communication: Architectures, technologies, and applications," *arXiv preprint arXiv:2412.08642*, 2024.

[90] H. Zhou *et al.*, "Large language models (LLMs) for wireless networks: An overview from the prompt engineering perspective," *Preprint arXiv:2411.04136*, 2024.

[91] K. Eswaran, "Identification via channels and constant-weight codes," 2005, available from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.92.7552.

[92] J. A. Cabrera, H. Boche, C. Deppe, R. F. Schaefer, C. Scheunert, and F. H. Fitzek, "6G and the post-shannon theory," *Shaping Future 6G Networks: Needs, Impacts, and Technologies*, pp. 271–294, 2021.

[93] M. J. Salariseddigh, U. Pereg, H. Boche, and C. Deppe, "Deterministic identification over channels with power constraints," *IEEE Transactions on Information Theory*, vol. 68, no. 1, pp. 1–24, 2021.

[94] C. von Lengerke, A. Hefele, J. A. Cabrera, O. Kosut, M. Reisslein, and F. H. P. Fitzek, "Identification codes: A topical review with design guidelines for practical systems," *IEEE Access*, vol. 11, pp. 14 961–14 982, 2023.

[95] C. von Lengerke, A. Hefele, J. A. Cabrera, M. Reisslein, and F. H. Fitzek, "Beyond the bound: A new performance perspective for identification via channels," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 8, pp. 2687–2706, 2023.

[96] R. Ahlswede, "Identification entropy," in *Identification and Other Probabilistic Models: Rudolf Ahlswede's Lectures on Information Theory 6*. Springer, Cham, Switzerland, 2021, pp. 375–397.

[97] ——, "L-identification for sources," in *Identification and Other Probabilistic Models: Rudolf Ahlswede's Lectures on Information Theory 6*. Springer, Cham, Switzerland, 2021, pp. 429–511.

[98] ——, "K-identification," in *Identification and Other Probabilistic Models: Rudolf Ahlswede's Lectures on Information Theory 6*. Springer, Cham, Switzerland, 2021, pp. 142–153.

[99] M. Mitzenmacher, "Digital fountains: A survey and look forward," in *Proc. IEEE Information Theory Workshop*, 2004, pp. 271–276.

[100] K. R. Duffy, J. Li, and M. Médard, "Capacity-achieving guessing random additive noise decoding," *IEEE Transactions on Information Theory*, vol. 65, no. 7, pp. 4023–4040, 2019.

[101] G. Tzimpragos, C. Kachris, I. B. Djordjevic, M. Cvijetic, D. Soudris, and I. Tomkos, "A survey on FEC codes for 100 G and beyond optical networks," *IEEE COMST*, vol. 18, no. 1, pp. 209–221, 2016.

[102] R. Ahlswede, "Elimination of correlation in random codes for arbitrarily varying channels," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 44, no. 2, p. 159–175, 1978.

[103] H. Boche and C. Deppe, "Robust and secure identification," in *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, 2017, pp. 1539–1543.

[104] ——, "Secure identification for wiretap channels; robustness, superadditivity and continuity," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1641–1655, 2018.

[105] H. Boche, C. Deppe, and A. Winter, "Secure and robust identification via classical-quantum channels," *IEEE Transactions on Information Theory*, vol. 65, no. 10, pp. 6734–6749, 2019.

[106] W. Labidi, C. Deppe, and H. Boche, "Secure identification for gaussian channels," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2872–2876.

[107] W. Labidi, H. Boche, C. Deppe, and M. Wiese, "Identification over the Gaussian channel in the presence of feedback," in *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, 2021, pp. 278–283.

[108] P. Colomer, C. Deppe, H. Boche, and A. Winter, "Deterministic identification over channels with finite output: A dimensional perspective on superlinear rates," in *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, 2024, pp. 297–302.

[109] I. Vorobyev, C. Deppe, and H. Boche, "Deterministic identification codes for fading channels," *Preprint arXiv:2404.02723*, 2024.

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

[111] ——, "Common randomness in information theory and cryptography. II. CR capacity," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 225–240, 1998.

[112] R. Ezzine, M. Wiese, C. Deppe, and H. Boche, "A general formula for uniform common randomness capacity," in *Proc. IEEE Information Theory Workshop (ITW)*, 2022, pp. 762–767.

[113] P. Kumar Herooru Sheshagiri, M. Reisslein, J. A. Cabrera, and F. H. P. Fitzek, "CFO-CR: Carrier frequency offset methodology for high-rate common randomness generation," *IEEE Access*, vol. 13, pp. 15 469–15 488, 2025.

[114] R. Ahlswede and G. Dueck, "Identification in the presence of feedback-a discovery of new capacity formulas," *IEEE Transactions on Information Theory*, vol. 35, no. 1, pp. 30–36, 1989.

[115] C. Shannon, "The zero error capacity of a noisy channel," *IRE Transactions on Information Theory*, vol. 2, no. 3, pp. 8–19, 1956.

[116] K. Kurosawa and T. Yoshida, "Strongly universal hashing and identification codes via channels," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 2091–2095, 1999.

[117] B. den Boer, "A simple and key-economical unconditional authentication scheme," *J. Comput. Secur.*, vol. 2, no. 1, pp. 65–71, 1993.

[118] J. Bierbrauer, "Universal hashing and geometric codes," *Designs, Codes and Cryptography*, vol. 11, pp. 207–221, 1997.

[119] R. Taylor, "An integrity check value algorithm for stream ciphers," in *Advances in Cryptology—CRYPTO'93: Proc. 13th Ann. Int. Cryptology Conf.* Springer, Berlin, Heidelberg, 1994, pp. 40–48.

[120] J.-P. Aumasson and D. J. Bernstein, "SipHash: A fast short-input PRF," in *Proc. Int. Conf. on Cryptology in India, Lecture Notes in Computer Science (LNSC), Vol. 7668*. Springer, Berlin, Heidelberg, 2012, pp. 489–508.

[121] O. Peters, "PolymurHash," Dec. 2024. [Online]. Available: https://github.com/orlp/polymur-hash

[122] H. Boche, R. F. Schaefer, and H. Vincent Poor, "Identification capacity of correlation-assisted discrete memoryless channels: Analytical properties and representations," in *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, 2019, pp. 470–474.

[123] R. Ezzine, W. Labidi, H. Boche, and C. Deppe, "Common randomness generation and identification over Gaussian channels," in *Proc. IEEE GLOBECOM*, 2020, pp. 1–6.
[124] P. K. Sheshagiri, J. A. Cabrera, and F. H. Fitzek, "Improving common randomness rate using software defined radios," in *Proc. IEEE INFOCOM WKSHPS*, 2024, pp. 1–2.
[125] P. K. Sheshagiri, S. Das, J. A. Cabrera, R. Bassoli, and F. H. Fitzek, "Common randomness generation for information theoretic security in post-quantum internet of things," in *Proc. IEEE 9th World Forum on Internet of Things (WF-IoT)*, 2023, pp. 1–6.
[126] M. Adil, H. Ullah Khan, M. Arif, M. Shah Nawaz, and F. Khan, "New dimensions for physical layer secret key generation: Excursion lengths-based key generation," *IEEE Access*, vol. 12, pp. 82 972–82 983, 2024.
[127] G. Li, C. Sun, J. Zhang, E. Jorswieck, B. Xiao, and A. Hu, "Physical layer key generation in 5G and beyond wireless communications: Challenges and opportunities," *Entropy*, vol. 21, no. 5, p. Art. no. 497, 2019.
[128] A. Ibrahim, R. Ferrara, and C. Deppe, "Identification under effective secrecy," in *Proc. IEEE Information Theory Workshop (ITW)*, 2021, pp. 1–6.
[129] M. Spandri, R. Ferrara, and C. Deppe, "Reed-Muller identification," in *Proc. Int. Zurich Seminar on Information and Communication (IZS)*, Mar. 2022, pp. 74–78.
[130] V. R. Sidorenko and C. Deppe, "Identification based on random coding," *Preprint arXiv:2207.03413*, 2022.
[131] R. Ferrara *et al.*, "Implementation and experimental evaluation of reed-solomon identification," in *Proc. VDE 27th European Wireless Conf.*, 2022, pp. 1–6.
[132] P. Kuthe, E. Schwarz Danni, and A. Kaplan, "Identification System Evaluation," Technical University of Dresden, Dresden, Seminar Report, Jul. 2023.
[133] M. Wanzenböck, "WanzenBug/g2p," Dec. 2024. [Online]. Available: https://github.com/WanzenBug/g2p
[134] V. Gopal *et al.*, "Fast CRC Computation for Generic Polynomials Using PCLMULQDQ Instruction," Intel Corp., Tech. Rep., 2009.
[135] R. Ferrara and C. von Lengerke, "Polynomial Universal Hashing," Dec. 2024. [Online]. Available: https://github.com/dragomang87/PolynomialUniversalHashing
[136] O. O. Erunkulu, A. M. Zungeru, C. K. Lebekwe, M. Mosalaosi, and J. M. Chuma, "5G mobile communication applications: A survey and comparison of use cases," *IEEE Access*, vol. 9, pp. 97 251–97 295, 2021.
[137] T. Hoeschele, C. Dietzel, D. Kopp, F. H. Fitzek, and M. Reisslein, "Importance of internet exchange point (IXP) infrastructure for 5G: Estimating the impact of 5G use cases," *Telecommunications Policy*, vol. 45, no. 3, p. Art. No. 102091, 2021.
[138] H. M. Kamdjou, D. Baudry, V. Havard, and S. Ouchani, "Resource-constrained extended reality operated with digital twin in industrial internet of things," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 928–950, 2024.
[139] S. Rezwan, H. Wu, J. A. Cabrera, G. T. Nguyen, M. Reisslein, and F. H. P. Fitzek, "cXR+ voxel-based semantic compression for networked immersion," *IEEE Access*, vol. 11, pp. 52 763–52 777, 2023.
[140] H. Sami *et al.*, "The metaverse: Survey, trends, novel pipeline ecosystem & future directions," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 4, pp. 2914–2960, 2024.
[141] S. Islam, N. Muslim, and J. W. Atwood, "A survey on multicasting in software-defined networking," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 355–387, 2018.
[142] Y. R. Julio, I. G. García, and J. M. Dí az, "Fulcrum rateless multicast distributed coding design," *IEEE Access*, vol. 11, pp. 73 839–73 849, 2023.
[143] E. Garro *et al.*, "5G mixed mode: NR multicast-broadcast services," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 390–403, 2020.
[144] P. Garrido, A. Fernandez, and R. Agueero, "To recode or not to recode: Optimizing RLNC recoding and performance evaluation over a COTS platform," in *Proc. 24th IEEE European Wireless Conf.*, 2018, pp. 1–7.
[145] E. Tasdemir *et al.*, "SpaRec: Sparse systematic RLNC recoding in multi-hop networks," *IEEE Access*, vol. 9, pp. 168 567–168 586, 2021.
[146] M. Z. Farooqi, S. M. Tabassum, M. H. Rehmani, and Y. Saleem, "A survey on network coding: From traditional wireless networks to emerging cognitive radio networks," *J. Network and Computer Appl.*, vol. 46, pp. 166–181, 2014.
[147] F. Gabriel, S. Wunderlich, S. Pandi, F. H. Fitzek, and M. Reisslein, "Caterpillar RLNC with feedback (CRLNC-FB): Reducing delay in selective repeat ARQ through coding," *IEEE Access*, vol. 6, pp. 44 787–44 802, 2018.
[148] F. Jamil, A. Javaid, T. Umer, and M. H. Rehmani, "A comprehensive survey of network coding in vehicular ad-hoc networks," *Wireless Networks*, vol. 23, no. 8, pp. 2395–2414, 2017.
[149] S. Kafaie, Y. Chen, O. A. Dobre, and M. H. Ahmed, "Joint inter-flow network coding and opportunistic routing in multi-hop wireless mesh networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1014–1035, 2018.
[150] D. E. Lucani *et al.*, "Fulcrum: Flexible network coding for heterogeneous devices," *IEEE Access*, vol. 6, pp. 77 890–77 910, 2018.
[151] J. Rosenberger, H. Boche, J. A. Cabrera, and F. H. Fitzek, "Consensus testing via relay networks by physical-layer network coding," in *Proc. IEEE Global Commun. Conf.*, 2024, pp. 1359–1364.
[152] A. Siddiqui, B. P. Rimal, M. Reisslein, and Y. Wang, "Survey on Unified Threat Management (UTM) systems for home networks," *IEEE Commun. Surveys & Tutorials*, vol. 26, no. 4, pp. 2459–2509, 2024.
[153] V. Sundaravarathan *et al.*, "Cross-Domain Solutions (CDS): A comprehensive survey," *IEEE Access*, vol. 12, pp. 163 551–163 620, 2024.
[154] X. Wang, B. Wang, Y. Wu, Z. Ning, S. Guo, and F. R. Yu, "A survey on trustworthy edge intelligence: From security and reliability to transparency and sustainability," *IEEE Communications Surveys & Tutorials, in print*, pp. 1–1, 2025.

Caspar von Lengerke studied Electrical Engineering at RWTH Aachen University, Germany, and received his Bachelor's and Master's degrees in 2017 and 2019 respectively. He joined the Deutsche Telekom Chair of Communication Networks at TU Dresden in 2021. His research interests lie in identification codes, as well as in cryptographic and non-cryptographic hash functions.

Juan A. Cabrera received the Dr.-Ing. degree from the TU Dresden, Germany in 2022. He works at the Deutsche Telekom Chair of Communication Networks at TU Dresden where he leads the research group on semantic and goal-oriented communications. His research interests are semantic and goal-oriented communications, functional compression, message identification, common randomness generation, network coding, and in-network distributed storage and computing.

Martin Reisslein (S'96-M'98-SM'03-F'14) received his Ph.D. in systems engineering from the University of Pennsylvania, Philadelphia, PA, USA in 1998. He is currently a Professor with the School of Electrical, Computer, and Energy Engineering, Arizona State University (ASU), Tempe, AZ, USA, and also Program Chair of the Computer Engineering (CEN) program at ASU. He is currently an Associate Editor for *IEEE Access* and *IEEE Transactions on Network and Service Management*.

Frank H. P. Fitzek is a Professor and head of the "Deutsche Telekom Chair of Communication Networks" at TU Dresden. He is the spokesman of the DFG Cluster of Excellence CeTI. He received his Ph.D. (Dr.-Ing.) in Electrical Engineering from the Technical University Berlin, Germany in 2002 and became Adjunct Professor at the University of Ferrara, Italy in the same year. In 2003 he joined Aalborg University as Associate Professor and later became Professor.