AES as Error Correction: Cryptosystems for Reliable Communication

Alejandro Cohen, Member, IEEE, Rafael G. L. D’Oliveira, Member, IEEE, Ken R. Duffy, Member, IEEE, Jongchan Woo, and Muriel Médard, Fellow, IEEE

Abstract—In this letter, we show that the Advanced Encryption Standard (AES) cryptosystem can be utilized as an error-correcting code to obtain reliability over noisy communication and data systems. Specifically, we show that by composing a simple padding followed by encrypting with AES, we can achieve error-correcting performance similar to random codes. We conduct an empirical comparison between our AES-based error correction approach, Random Linear Codes, and CA-Polar codes to assess their performance in practical scenarios. Our results show that the AES-based approach achieves nearly identical performance to the other well-established error-correcting codes.

Index Terms—Communication system security, telecommunication network reliability, cryptography, error correction codes, channel coding.

I. INTRODUCTION

Since Shannon’s groundbreaking work in 1948 [1] to modern day data systems [2], error-correcting codes have been employed to achieve reliable communication at the highest possible data rates [3], [4]. For security purposes, widely-used computational cryptosystems, such as the Advanced Encryption Standard (AES) [5], [6], are typically implemented as a pre-processing step to encrypt information [7], [8]. As a result, the conventional encoding process involves two distinct stages: first, encryption, followed by the application of an error-correcting code. More recently, however, the reverse ordering of encoding and then encrypting has been proposed in a setting where reliability and security can be obtained by partial encryption after error correction encoding [9].

The relationship between coding and cryptography has a rich and extensive history. Indeed, uniformly distributed outputs appear both in the coding scheme of the original work of Shannon and as a desideratum for cryptosystems, albeit for different purposes. In this work, we strengthen this relationship by showing how a cryptosystem, namely AES, can be utilized as an error correcting code. We show that, by utilizing a mere padding in order to manage the code rate, followed by an AES encryption, we can build a high performance error correction mechanism. Consequently, the encoding can be performed in a single stage by the cryptosystem while achieving reliability at high data rates.

To decode the transmitted message, we employ both hard- and soft-detection variants of the recently developed Guessing Additive Noise Decoding (GRAND) [10], [11], [12], [13], [14]. Nearly all error correction decoders are co-designed with specific linear codes and operate by clever exploitation of code-book structure [15]. In contrast, by focusing on identifying the digital corruption of the transmitted codeword and then inferring the codeword, GRAND algorithms can decode all code structures, including the nonlinear one we construct with AES.

GRAND algorithms operate by sequentially creating putative digital noise effects in order from most to least likely according to channel statistics or soft information. These noise effects are removed from the demodulated received signal, and what remains is queried for membership of the codebook. The first codeword element that is identified is a maximum likelihood decoding [10]. To execute this procedure only requires a method to generate putative noise effect sequences in decreasing order of likelihood, which is independent of the codebook, and a function to test codebook membership, which is codebook specific. For linear codes, where codebook membership can be tested by matrix multiplication, efficient in-silicon implementations [16], [17] and syntheses [18], [19], [20] in both hard and soft-detection settings have established the flexibility and energy efficiency of GRAND strategies.

To adapt GRAND algorithms for use with non-linear codes created from cryptographic operations, it suffices to establish a mechanism to encode message data and to test a string for codebook membership. Our joint decryption-decoding scheme is based on a recently proposed decoding scheme [9]. We show that the performance of our AES-based error correcting approach is similar to that of Random Linear Codes (RLC), which are theoretically known to be good codes [21], [22], and CRC-Assisted Polar (CA-Polar) codes, which have recently been introduced into the 5G New Radio communications standard for all control channel communications [23].

This study does not advocate for discarding existing coding schemes and decoding algorithms that are currently deployed. Rather, the goal is to enhance current understanding by establishing that cryptographic functions can expand the available options for system designers in their selection of forward error correction methods.

II. PRELIMINARIES

A. Setting

We consider a setting where a transmitter wishes to send a message $m^k \in \mathbb{F}_2^k$ to a legitimate receiver over an additive

noise channel. The sender uses a codebook, which maps each \( m^k \) to a unique codeword \( x^n \in \mathbb{F}_2^n \) where \( n > k \). The rate of the codebook is \( k/n \). The receiver sees the output of the channel which is a demodulated sequence \( y^n = x^n \oplus w^n \) where \( w^n \in \mathbb{F}_2^n \) is termed the noise.

As the channel is noisy, redundancy must be added, where the message is expanded to \( n > k \) bits, to enable error correction at the receiver. Essentially all deployed error correction codes are linear functions of the message data [15]. As a result, they provide no security as they are susceptible to plaintext attacks. The goal in this work is to consider the error correcting performance of a code built from a non-linear cryptographic function that has security and error correction capabilities.

### B. The AES Cryptosystem

AES is a symmetric key encryption algorithm that was established by the U.S. National Institute of Standards and Technology (NIST) in 2001 [5], [6]. It is designed to protect sensitive information ensuring secure communication and data storage. AES operates on fixed-sized blocks of data (128 bits) and offers varying degrees of security depending on the chosen key size between 128, 192, and 256 bits.

The algorithm utilizes a series of transformations, including substitutions, permutations, and mixing of input data, which are executed in multiple rounds to produce an encrypted output. For our purposes, it suffices to consider AES as consisting of two functions, an encryption function \( \text{Crypt}_s : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n \), where \( s \) is the secret key, and a decryption function \( \text{DeCrypt}_s : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n \) such that \( \text{DeCrypt}_s \circ \text{Crypt}_s \) is the identity function. For further details on the inner workings of AES we refer to [24].

By design, the output of AES exhibits a high degree of statistical randomness. Indeed, AES has been analyzed to perform well as a pseudo-random number generator [25]. It is this property of AES which we exploit in our code construction in Section III.

### C. Guessing Random Additive Noise Decoding

Algorithm 1 illustrates the GRAND decoding process [10], [14] executed at the receiver. Given the potentially corrupted demodulated binary sequence \( y^n \in \mathbb{F}_2^n \) and a method to test a string for codebook membership, using channel statistics or soft-information the decoder generates putative noise effect sequences, \( z^n \in \mathbb{F}_2^n \), from most likely to least likely breaking ties arbitrarily (resp. to line 3 in Algo. 1). For example, if the communication is over a binary symmetric channel, which is consistent with a highly interleaved hard detection system, then putative noise effects are generated in increasing order of Hamming weight [10], [12], [16]. If the system has soft input in the form of per-bit reliabilities, then putative noise effects can be sequentially generated in terms of increasing logistic-weight, e.g. [14], [17]. Each putative noise effect \( z^n \) is removed from \( y^n \) and remainder \( y^n \oplus z^n \), where \( \oplus \) denotes addition in \( \mathbb{F}_2^n \), is checked for codebook membership (resp. to line 4 in Algo. 1). The first codebook element that is identified is a maximum likelihood decoding.

GRAND algorithms are universal decoders that only require a mechanism to check codebook membership. Consequently, to generate codes using non-linear cryptographic functions, it suffices to have a mechanism to encode a message and a mechanism to check for code-book membership.

### III. AES AS ERROR CORRECTION

In this section, we use the AES cryptosystem [5], [6] to create an error-correcting code that takes \( k \) message bits and maps them to \( n \geq k \) coded bits. In Figure 1, we illustrate the encoding and decoding operations for the proposed scheme using AES as an error correcting code, and for an existing error correction scheme using a CA-Polar code, as in 5G NR [23]. For the AES cryptosystem, we use a standard FIPS-197 scheme as given in [5].

We start by presenting the encoding process at the transmitter. Figure 2 illustrate the encoding process using the considered AES-based error correction approach. A message \( m^k \in \mathbb{F}_2^k \) of size \( k \) is first padded with a function Pad : \( \mathbb{F}_2^k \rightarrow \mathbb{F}_2^n \) that concatenates \( n-k \) zeroes to the end of \( m^k \) and then encrypted with AES using a secret key \( s \), shared by the transmitter and receiver. Thus the encoder is given by \( \text{Crypt}_s \circ \text{Pad} : \mathbb{F}_2^k \rightarrow \mathbb{F}_2^n \) and the rate in the proposed scheme is \( R = k/n \).

Equipped with the secret key \( s \), the receiver can execute a hard- or soft-detection variant of GRAND, as in Algo. 1 so long as it can specify a codebook membership checking function. For a binary string of length \( n \) bits, \( y^n \oplus z^n \in \mathbb{F}_2^n \), this consists of a two-part procedure. First, the string is decrypted using the secret key \( s \), \( \text{DeCrypt}(y^n \oplus z^n, s) \). If the final \( n-k \) entries of what results are all 0s, corresponding to a padded string, then \( y^n \oplus z^n \) is in the codebook.

### IV. EMPIRICAL STUDY

In this section, we evaluate the performance of AES as an error-correcting code when compared with existing binary linear codes. The results demonstrate that a cryptosystem such as AES, whose outcome is considered pseudo-random, is capable of being used to generate error-correcting performance that is similar to random linear codes (RLCs).

For the performance evaluation, binary messages of length \( k \) bits, \( m^k \), are chosen uniformly at random from \( \mathbb{F}_2^k \) and transmitted using Binary Phase Shift Keying (BPSK) over a Complex Additive White Gaussian (CAWGN) channel. In the

---

**Algorithm 1** Guessing Random Additive Noise Decoding

| Input: code-book membership test; demodulated sequence \( y^n \); (optional) statistical channel or soft-information \( \phi^n \) |
| Output: \( c^{s,n} \), maximum likelihood decoding |

1. \( d \leftarrow 0 \)
2. \( \text{while } d = 0 \text{ do} \)
3. \( z^n \leftarrow \text{next most likely noise effect} \)
4. \( \text{if } y^n \oplus z^n \text{ is in the codebook then} \)
5. \( c^{s,n} \leftarrow y^n \oplus z^n \)
6. \( d \leftarrow 1 \)
7. \( \text{return } c^{s,n} \)
8. \( \text{end if} \)
9. \( \text{end while} \)
Fig. 1. Proposed cryptosystem with a uniform outcome as error correction, e.g., AES (upper figure) and RLC, CA-Polar code, or a CRC decoded with GRAND (bottom figure).

For decoding, we use GRAND [10], [31] as presented in Algo. 1. For hard detection, the noise effects are generated in increasing order of Hamming weight. For soft-detection, they are created in increasing order of logistic weight, as defined in the Ordered Reliability Bits GRAND (ORBGRAND) [14], [32] variant. Performance evaluation results for GRAND and ORBGRAND when used with linear code-structures beyond those shown here and, in particular, when compared to CA-Polar codes decoded with successive cancellation list decoding, appear in the published literature, e.g. [14], [16], [17], [19], [33], [34]. Implementations used for GRAND, ORBGRAND and AES are available at [35] and [36], respectively.

In Figure 3 we show the BER and BLER vs Eb/N0 for an RLC [128,116], a CA-Polar [128,116] code, and an AES as error-correcting code with $k=116$ and with a zero padding of $n-k=12$. The rate for all the tested codes presented in this figure is $R=0.91$. The black solid and dashed lines show the performance of RLC codes with GRAND and ORBGRAND, respectively. The red and blue lines in the same present the performance of RLC and CA-Polar codes with CRC, respectively. The yellow lines show the performance of AES as an error-correcting code. All the schemes tested obtain almost the same performance for decoding with a soft-information decoder as proposed by ORBGRAND. Using GRAND for hard-decision decoding, both the RLC code and AES as an error-correcting code achieve the same performance. However, we note that the coding schemes with CRC and using GRAND decoder can obtain slightly better performance in SNR higher than $\sim 8$dB. This is settled with the known results in the literature, which show that structured codes can slightly outperform the performance of random codes in the tested rate and code-size regime when one utilizes a hard-decision decoder. In Figures 4 and 5, with rates $R=0.88$ and $R=0.94$, respectively, the same performance comparison is presented as in Figure 3. Similar performance trends to those for $R=0.91$ are observed.
Fig. 3. BER on the left and BLER on the right vs. Eb/N0 for codes of length $n = 128$ and rate $R = 0.91$ encoded with AES as error correcting code and with RLC and CA-Polar code. The joint decryption-decoding is performed with GRAND and ORBGRAND with soft-information.

Fig. 4. BER on the left and BLER on the right vs. Eb/N0 for codes of length $n = 128$ and rate $R = 0.88$ encoded with AES as error correcting code and with RLC and CA-Polar code.

Fig. 5. BER vs. Eb/N0 for codes of length $n = 128$ and rate $R = 0.94$ encoded with AES as error correcting and with RLC.

V. DISCUSSION AND FUTURE WORK

In this letter, we consider a novel approach of using common cryptosystems, which behave like random functions, to create error correcting codes. We assessed the proposed performance, finding that adapting AES to generate an error-correcting code results, perhaps surprisingly, in similar bit- and block-error performance as Random Linear Codes, which are provably good [22], and CA-Polar codes that are found in the 5G NR standard [23].

The complexity of all GRAND algorithms consists of two components [12], [14]: the generation of putative noise effects; and testing codebook membership. For hard- and soft-detection decoding, efficient circuits have been designed and taped out that create noise effect sequences, which can be reused for our purposes.

For each sequence created, a codebook membership test is executed. For linear codes, codebook membership consists of a single matrix multiplication that can be achieved with in-memory computation in silicon, resulting in sub pJ per bit soft detection decoding [17]. While the true noise effect will be identified earlier in a well-operating code, for any code of any structure consisting of $2^k$ binary strings of length $n$, it has been established that any GRAND algorithm will identify an erroneous decoding after approximately $2^{n-k}$ codebook queries, which thus serves as an upper-bound on its decoding complexity [10]. As a result, GRAND algorithms are suitable for decoding any moderate redundancy code, typically reported up to 25 or 30 redundant bits, of any length.

To check a sequence’s codebook membership with the AES scheme, we must decrypt, which is more involved than matrix
multiplication and slower in software. Due to the importance of AES in cryptographic applications, however, dedicated hardware circuits has been created that do so efficiently, e.g. [37], making the approach viable in practice.

This approach opens a new area of exploration. The AES as an error-correcting scheme we present is a good candidate for the investigation of possible security properties. Future work includes considering how to incorporate other cryptosystems in the literature, in order to jointly obtain security and error correction capabilities. Furthermore, practical considerations that treat the computational complexity of the AES as an error-correcting scheme, its memory requirements, running time, and parallelization are interesting questions for future work.

REFERENCES

[1] C. E. Shannon, “A mathematical theory of communication,” Bell Syst. Tech. J., vol. 27, no. 3, pp. 379–423, Jul. 1948.
[2] X. Lin and N. Lee, 5G and Beyond. Springer, 2021.
[3] W. C. Huffman, J.-L. Kim, and P. Solé, Concise Encyclopedia of Coding Theory. London, U.K.: Chapman & Hall/CRC, 2021.
[4] T. M. Cover and J. A. Thomas, Elements of Information Theory. Hoboken, NJ, USA: Wiley, 2012.
[5] M. J. Dworkin et al., “Advanced encryption standard (AES),” Federal Inf. Process. Standards Publication, vol. 197, no. 441, pp. 1–47, 2001.
[6] J. Nechvatal et al., “Report on the development of the advanced encryption standard (AES),” J. Res. Nat. Inst. Standards Technol., vol. 106, no. 3, p. 511, 2001.
[7] J. Hoffstein, J. Pipher, J. H. Silverman, and J. H. Silverman, An Introduction to Mathematical Cryptography, 1st ed. New York, NY, USA: Springer, 2008.
[8] W. Stallings, Cryptography and Network Security, 4/E. India: Pearson Education, 2006.
[9] A. Cohen, R. G. L. D’Oliveira, K. R. Duffy, and M. Méard, “Partial encryption after encoding for security and reliability in data systems,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jun. 2022, pp. 1779–1784.
[10] K. R. Duffy, J. Li, and M. Méard, “Capacity-achieving guessing random additive noise decoding,” IEEE Trans. Inf. Theory, vol. 65, no. 7, pp. 4023–4040, Jul. 2019.
[11] A. Solomon, K. R. Duffy, and M. Méard, “Soft maximum likelihood decoding using GRAND,” in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2020, pp. 1–6.
[12] W. An, M. Méard, and K. R. Duffy, “Keep the bursts and ditch the interleavers,” IEEE Trans. Commun., vol. 70, no. 6, pp. 3655–3667, Jun. 2022.
[13] K. R. Duffy, M. Méard, and W. An, “Guessing random additive noise decoding with symbol reliability information (SRGRAND),” IEEE Trans. Commun., vol. 70, no. 1, pp. 3–18, Jan. 2022.
[14] K. R. Duffy, W. An, and M. Méard, “Ordered reliability bits guessing random additive noise decoding,” IEEE Trans. Signal Process., vol. 70, pp. 4528–4542, 2022.
[15] S. Lin and D. J. Costello, Error Control Coding: Fundamentals and Applications. Upper Saddle River, NJ, USA: Prentice-Hall, 2004.
[16] A. Riaz et al., “Multi-code multi-rate universal maximum likelihood decoder using GRAND,” in Proc. IEEE 47th Eur. Solid State Circuits Conf. (ESSCIRC), Sep. 2021, pp. 239–246.
[17] A. Riaz et al., “A sub-0.8pJ/b 16.3Gbps/mm² universal soft-detection decoder using ORBGRAND in 40 nm CMOS,” in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2023, pp. 432–434.
[18] S. M. Abbas, T. Tonnellier, F. Ercan, and W. J. Gross, “High-throughput VLSI architecture for GRAND,” in Proc. IEEE Workshop Signal Process. Syst. (SiPS), Oct. 2020, pp. 1–5.
[19] S. M. Abbas, T. Tonnellier, F. Ercan, M. Jalaleddine, and W. J. Gross, “High-throughput and energy-efficient VLSI architecture for ordered reliability bits GRAND,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 30, no. 6, pp. 681–693, Jun. 2022.
[20] C. Condo, “A fixed latency ORBGRAND decoder architecture with LUT-aided error-pattern scheduling,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 69, no. 5, pp. 2203–2211, May 2022.
[21] R. Gallager, “The random coding bound is tight for the average code (corresp.),” IEEE Trans. Inf. Theory, vol. IT-19, no. 2, pp. 244–246, Mar. 1973.
[22] J. T. Coffey and R. M. Goodman, “Any code of which we cannot think is good,” IEEE Trans. Inf. Theory, vol. 36, no. 6, pp. 1453–1461, Nov. 1990.
[23] E. Arikan, “Channel polarization: A method for constructing capacity-achieving codes,” in Proc. IEEE Int. Symp. Inf. Theory Proc., Jul. 2011, pp. 1–5.
[24] I. Tal and A. Vardy, “List decoding of polar codes,” in Proc. IEEE Trans. Inf. Theory, vol. 61, no. 5, pp. 2213–2226, May 2015.
[25] A. Riaz et al., “Batteryless, wireless, and secure SoC for implantable strain sensing,” IEEE Open J. Solid-State Circuits Soc., vol. 3, pp. 41–51, 2023.