INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Padding Techniques for Identifying Decodable Syndrome in Post-Quantum Digital Signature Schemes

Rupali Khurana[1], Ekta Narwal[1]*, Deepika[2]

1 Department of Mathematics, Maharshi Dayanand University, Rohtak, Haryana, India
2 Universitat Rovira i Virgili, Avinguda Catalunya, Tarragona, Spain

OPEN ACCESS

Received: 25-07-2023
Accepted: 25-11-2023
Published: 28-12-2023

Citation: Khurana R, Narwal E, Deepika (2023) Padding Techniques for Identifying Decodable Syndrome in Post-Quantum Digital Signature Schemes. Indian Journal of Science and Technology 16(48): 4638-4647. https://doi.org/10.17485/IJST/v16i48.1871

* **Corresponding author**.

ektanarwal.math@mdurohtak.ac.in

Funding: University Grant Commission (UGC), New Delhi, India

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: The aim of this study is to design a MATLAB algorithm that will identify decodable syndromes easily. **Methods:** The most time-consuming step in developing code-based digital signature schemes is to identify a decodable syndrome. So, to find a decodable syndrome in a short period of time several padding techniques are presented in this paper. Our study uses a polar code with blocklength 16 with a rate of 0.5, and its Successive Cancellation (SC) decoding algorithm is used to decode the syndrome. The techniques suggested in this paper can be used to generate polar code-based digital signatures more quickly. **Findings:** Our study evaluates the effectiveness of padding techniques by simulating random syndromes 10,000 times over Binary Symmetric Channel (BSC) and Additive White Gaussian Noise (AWGN) channel by calculating the success rate, failure rate, and mean failure count for each channel. **Novelty:** In this paper, we present graphs showing the number of syndromes that failed to decode in the BSC and AWGN channels when simulating a predefined number of times. Additionally, we calculate the success rate, failure rate, and mean failure count of decoding for both channels, which demonstrate that the proposed padding techniques are highly effective in decoding syndromes in AWGN.

Keywords: Quantum computing; Post-Quantum digital signature; Decodable syndromes; Padding; Polar codes; Successive cancellation decoding

1 Introduction

As the use of the internet has increased, sharing useful information from one location to another has become an important part of everyone's lives. Therefore, information security becomes a critical concern for users. A digital signature[(1)] plays an important role in data security since it ensures the authenticity, integrity, and non-repudiation of data. Recently, digital signatures have been widely used to maintain data integrity[(2)]. But these signatures are deployed with the evolution of quantum computers[(3)], as they break all the recently used cryptographic digital signatures[(4)]. Thus, researchers are developing post-quantum digital signatures to resist attacks against quantum computers[(5)]. Codes used in cryptography are one of the most promising areas of

Post-Quantum Cryptography (PQC)[6]. A code-based digital signature is based on the syndrome decoding problem[7]. In this problem, we have to choose a binary matrix H of order $(n-k)\times n$, a word $x \in F_2^{n-k}$, an integer $n > 0$. The problem is to find a vector $y \in F_2^n$ having weight $\leq n$ satisfying $Hy^T = x^T$[8]. Following are the steps involved in the creation of a code-based digital signature[7]:

- **Parameters Generation:** The signer first generates the public and private keys, which are then used to generate and verify the signature. Private keys have been kept secret by the signer, while public keys are publicly available for users to verify the signature.
- **Signature Generation:** To generate a signature, the signer first calculates the hash value, and then, using the private keys along with some mathematical algorithms, the signer generates the signature.
- **Signature Verification:** Verification of the signature is performed using the signer's public key as well as the verification algorithm, which generates a specific output. The signature is verified if the output meets certain criteria, otherwise, it is invalid.

The code-based signature schemes[9] offer fast computation speed, making them an excellent choice for post-quantum signatures[10]. In 2019, Sahu et al. proposed a signature scheme based on the CFS scheme using modified QC-LDPC codes[11]. This scheme improves the CFS scheme because it is fast, secure, and has a small public key size. Then, using the modified Reed-Muller codes, Lee et al. proposed a modified pqsigRM signature scheme in 2020[12]. This scheme has various features, like low signing complexity and a small key size. Later on, Forghani et al. proposed the PolarSig scheme using polar codes in 2020 that depend on the CFS scheme[13]. It has low signing complexity, a small public key size, and is secure against forgery and key recovery attacks. With the help of puncturing and random omission of the frozen bits, this scheme reduces the length of the polar code and makes it possible, to decode every random syndrome. In 2023, Hooshmand et al. proposed two identification schemes based on polar codes[14]. These schemes have a cheating probability of $\left(\frac{2}{3}\right)^r$, where r is the number of times the protocol repeats. Additionally, these schemes are resistant to information set decoding attacks. Then, Makoui et al. proposed a signature scheme in 2023 using the McEliece cryptosystem[15]. The signing process of this scheme is simpler, and it requires less computing time to sign a document. The authors also proved that it is secure against public key structural attacks.

Based on the previous study of polar code-based digital signature, we observe that Forghani[13] reduced the length of the polar code by randomly omitting and puncturing of frozen bits, leading to the successful decoding of syndromes. This paper works on the original length of the polar code and proposes several padding techniques that can be used to decode syndromes effectively. A signature scheme can be generated by combining the decoded syndromes with polar codes. This study uses polar codes with a blocklength of 16 and a rate of 0.5.

1.1 Polar Codes

A (N,K) Polar code[16] is a type of error-correcting code which is constructed by computing the Kronecker product of the kernel matrix $\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$. Here, $N = 2^n$;be the length of the polar code, and K be the number of information bits[17].

1.2 Bhattacharyya Parameters

The Bhattacharyya parameter[14] of the B-DMC channel $\mathbf{W} : \mathbf{X} \rightarrow \mathbf{Y}$is denoted by $Z(W)$, and is defined as follows:

$$Z(W) = \sum_{y \in Y} \sqrt{W\left(\frac{y}{0}\right) W\left(\frac{y}{1}\right)}$$

Here,$\left\{W\left(\frac{y}{x}\right), y \in Y, x \in X\right\}$denotes the transfer probabilities of W.

1.3 Good and Bad Bit Channels

The good-bit channels are those channels whose values correspond to the smallest value of the Bhattacharyya parameters, whereas the bad-bit channels are those whose values correspond to the largest values of the Bhattacharyya parameters[18].

This paper presents six different padding techniques for decoding a random syndrome using successive cancellation decoding. As soon as the decodable syndrome is identified, a digital signature based on polar codes can be easily created. The effectiveness of padding techniques is assessed by calculating the success rate, failure rate, and mean failure count. Furthermore, padding techniques are implemented in MATLAB.

2 Methodology

This paper presents six different padding approaches for decoding the syndrome. To determine whether the syndrome can be decoded, padding techniques are applied one by one. We can say that a syndrome is not decodable if all padding techniques fail.

2.1 The Proposed Padding Techniques

This manuscript presents six different padding techniques in a specific way and tries to effectively decode a random syndrome. The six padding techniques are explained below with the help of an example.

For the (16,8) Polar code, the Bhattacharyya parameters of the BSC with a crossover probability of 0.2494 are as follows:

$Z = [1.0000\ 1.0000\ 1.0000\ 0.9987\ 1.0000\ 0.9974\ 0.9950\ 0.8638\ 0.9628\ 0.6513\ 0.5300\ 0.0989]$

Among all Bhattacharyya parameters, the indices [8 10 11 12 13 14 15 16] correspond to the smallest value, known as Good-Bit Channels. The indices [1 2 3 4 5 6 7 9] corresponding to the largest value of the Bhattacharyya parameters are known as Bad-Bit Channels.

Suppose that the word $c = [1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0]$ is transmitted over the BSC and it is received with the error $e = [0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0]$. Now, the received word r is calculated as the sum of the transmitted word c and the error vector e, i.e., $r = c + e = [1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0]$. The parity check matrix of the polar code is:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Now, the syndrome s is calculated as the product of the received word r and transpose of the parity-check matrix H, i.e., $s = r * H^{'} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$

In the next step, padding techniques are performed one by one and examine whether the syndrome is correctly decoded. Here are the different padding techniques:

2.1.1 First Padding Technique

To pad the syndrome into a sequence of N length, we first place the syndrome's bits into the bad bit channels. Then, one by one, we select every bit of the syndrome and place it in the good bit channels. The Figure 1 figure shows the complete working of the first padding technique.

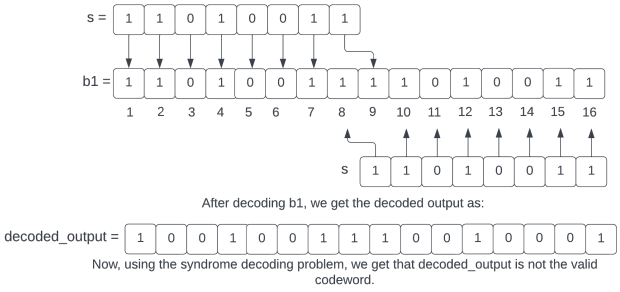

Fig 1. Working of the first padding technique

2.1.2 Second Padding Technique

The second method of padding involves placing the syndrome's bits in the bad bit channels. Then, starting with the last bit of the syndrome, we select each one and place them in the good bit channels. The Figure 2 given below explains the complete working of the second padding technique.

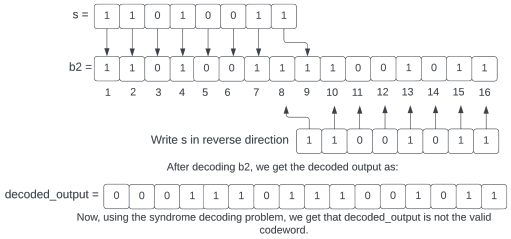

Fig 2. Working of the second padding technique

2.1.3 Third Padding Technique

We start the third padding process by inserting the syndrome's bits into the bad positions. Then, starting from the syndrome's last, we select every bit of the syndrome one by one, complement them, and place them in good positions. The Figure 3 figure explains the complete working of the third padding technique.

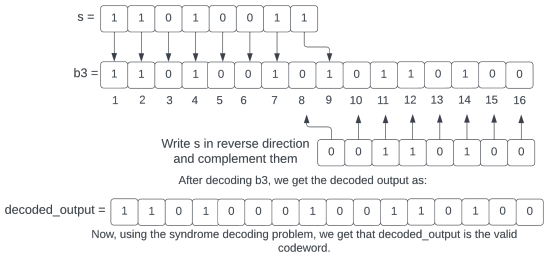

Fig 3. Working of the third padding technique

2.1.4 Fourth Padding Technique

The fourth padding technique begins by positioning the syndrome's bits in bad positions. After that, we select each bit of the syndrome from the first position, complement it, and insert it in the good positions. The Figure 4 figure explains the complete working of the fourth padding technique.

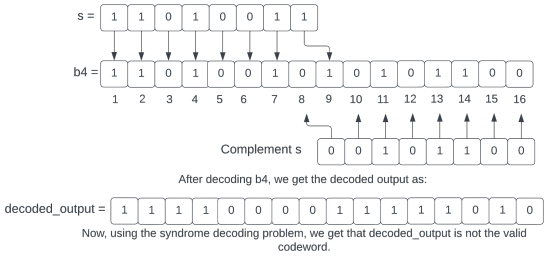

Fig 4. Working of the fourth padding technique

2.1.5 Fifth Padding Technique

In the fifth padding technique, the bits of the syndrome are initially placed in bad positions. Following that, we select each bit of the syndrome one by one and then complement the first and last bits of the syndrome while keeping the rest same, then, place them in good positions. The figure given below explains the complete working of the fifth padding technique. [Figure 5]

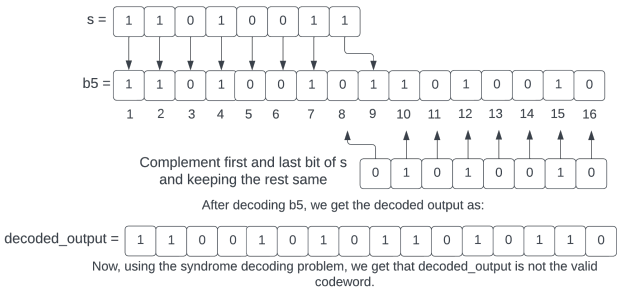

Fig 5. Working of the fifth padding technique

2.1.6 Sixth Padding Technique

The sixth padding technique begins by putting the syndrome's bits in the bad positions. After that, we select each bit of the syndrome one by one, then complement all bits except the first and last bits of the syndrome and insert them in good positions. The Figure 6elow figure explains the complete working of the sixth padding technique.

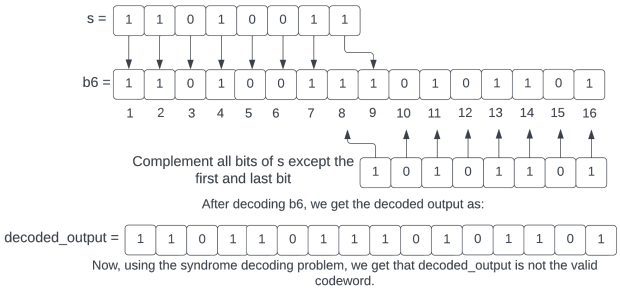

Fig 6. Working of the sixth padding technique

2.2 Flowchart of the Padding Techniques

The main aim of creating a code-based digital signature is to find a decodable syndrome. Padding is used in the polar code-based digital signature for decoding a syndrome. This manuscript uses padding in a specific way to effectively decode any random syndrome using successive cancellation decoding[18]. The Figure 7 given below illustrates the complete working mechanism of the padding techniques. A padding technique is initially applied to the syndrome, and then successive cancellation decoding is used to decode it. If the syndrome is successfully decoded, then it is said to be decodable; otherwise, a different padding approach is used, and repeat the same procedure until the syndrome is decodable.

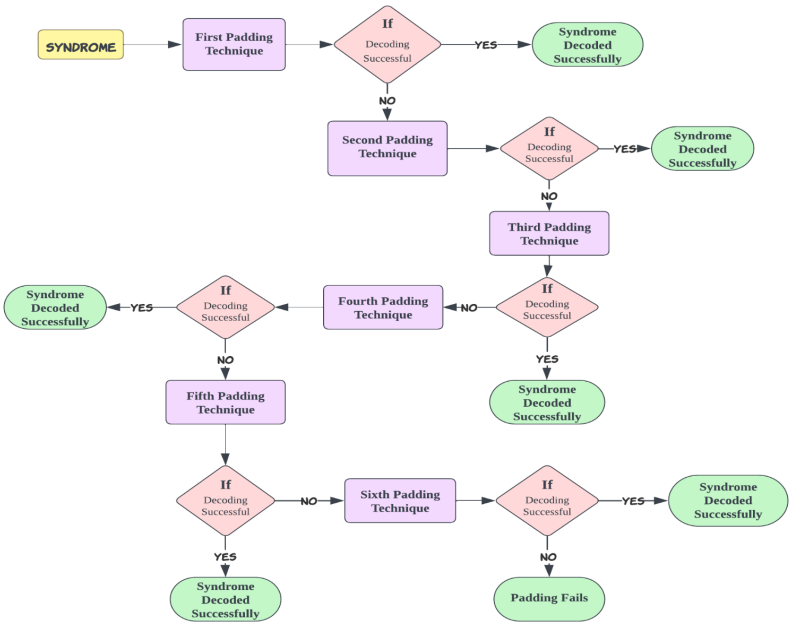

Fig 7. Working mechanism of the padding techniques

2.3 Implementation of the Padding Techniques

This section provides a step-by-step explanation of padding techniques in MATLAB. The Bhattacharyya parameters of a Polar code with blocklength 16 are first generated, and then these parameters are used to determine the good and bad bit channels. After that, the keys are generated using the Kronecker product of the matrix F. Later, each of the six padding approaches is applied to the syndrome one by one and attempts to decode it. MATLAB implementation of different padding approaches is explained below:

Padding Techniques MATLAB Implementation

Blocklength = 16;
Rate = 0.5;
K = Rate*Blocklength;
n = log2(Blocklength);
p = 0.2494
%Input a Random Message p1 of 1×K
p1 = randi(2,1,K)-1;
%Syndrome
s = [1 1 0 1 0 0 1 1]
%Calculate the Bhattacharyya Parameters
Z = compute_bhattacharyya_parameters(p, Blocklength);
%Find Good and Bad Bit Channels
[Good, Bad] = good_bad_channels(Z, K);
%Key Generation
F = $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$
G_n = Kronecker_product(F, n);
H = G_n(:,Bad); **%H is the parity-check matrix**
%First Padding
b1 = first_padding(s, Good);
x = b1;

```
y = bsc_channel(x, p);
decoded_output = polar_decoder(y, Blocklength, K);
v1 = mod(decoded_output*H', 2);
if v1 == s
fprintf('Padding is successful for b1');
else
%Second Padding
b2 = second_padding(s, Good);
x = b2;
y = bsc_channel(x, p);
decoded_output = polar_decoder(y, Blocklength, K);
v2 = mod(decoded_output*H', 2);
 if v2 == s
fprintf('Padding is successful for b2');
else
%Third Padding
b3 = third_padding(s, Good);
x = b3;
y = bsc_channel(x, p);
decoded_output = polar_decoder(y, Blocklength, K);
v3 = mod(decoded_output*H', 2);
 if v3 == s
fprintf('Padding is successful for b3');
else
%Fourth Padding
b4 = fourth_padding(s, Good);
x = b4;
y = bsc_channel(x, p);
decoded_output = polar_decoder(y, Blocklength, K);
v4 = mod(decoded_output*H', 2);
 if v4 == s
fprintf('Padding is successful for b4');
else
%Fifth Padding
b5 = fifth_padding(s, Good);
x = b5;
y = bsc_channel(x, p);
decoded_output = polar_decoder(y, Blocklength, K);
v5 = mod(decoded_output*H',2);
 if v5 == s
fprintf('Padding is successful for b5');
else
%Sixth Padding
b6 = sixth_padding(s, Good);
x = b6;
y = bsc_channel(x, p);
decoded_output = polar_decoder(y, Blocklength, K);
v6 = mod(decoded_output*H',2);
 if v6 == s
fprintf('Padding is successful for b6');
else
 fprintf('Padding fails');
end
```

end
end
end
end
end

3 Results and Discussion

This paper investigates the realm of polar codes, taking inspiration from the PolarSig[13] approach in order to do so. In contrast to PolarSig, in which frozen bits were omitted and punctured for the purpose of optimizing polar codes for digital signatures, the study in this article focuses exclusively on the original, unaltered polar codes used in PolarSig. It is for this reason that the authors introduced six distinct padding techniques in order to enhance the syndrome's decoding process via the SC decoding algorithm. As a result of the study, a meticulously crafted error vector is presented that aligns with the core objectives of the study. The paper adds innovative insights into the field of digital signatures based on polar codes and presents an innovative approach. It is hoped that this work will contribute to a broader understanding of polar codes and their possible applications. As a result, it gives us valuable insight into how digital signatures are likely to develop in the future while keeping them both efficient and secure.

Here, in this section, the efficiency of padding techniques in BSC and AWGN channels is evaluated by calculating the success rate, failure rate, and mean failure count of decoding by simulating random messages of K length up to 10,000 times. For this, the number of different syndromes that cannot be decoded after being simulated a predetermined number of times in BSC and AWGN is noted. The software we used is MATLAB, running on a 64-bit Windows 11 Pro operating system, and the hardware parameters are Intel Core i7-10700T, 2.00GHz, and 16GB RAM for implementation of the padding techniques. Figure 8 illustrates the number of random syndromes that are not decoded in the AWGN channel. From this figure, we observe that after simulating 10,000 random syndromes of length 8, we get 9698 decodable syndromes.

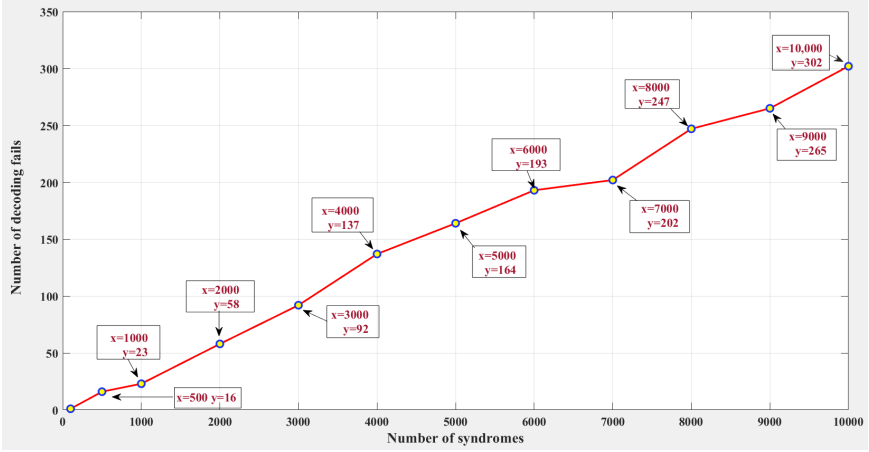

Fig 8. Number of syndromes simulated in AWGN vs number of decoding fails

Figure 9 shows the simulation results of a number of random syndromes that failed to decode in the BSC. From the figure, we can see that 270 decodable syndromes are obtained after the simulation of 10,000 random syndromes.

Now, the success and failure rates of decoding are calculated by using the following formula:

$$Failure\ Rate = \frac{Number\ of\ failures}{Total\ number\ of\ simulations} * 100$$

$$Success\ Rate = 100 - Failure\ Rate$$

Based on the calculation of the success rate and failure rate, we find that more than 96% of the syndromes are successfully decoded in the AWGN channel, whereas in the BSC, less than 3% of the syndromes are successfully decoded. To analyze the

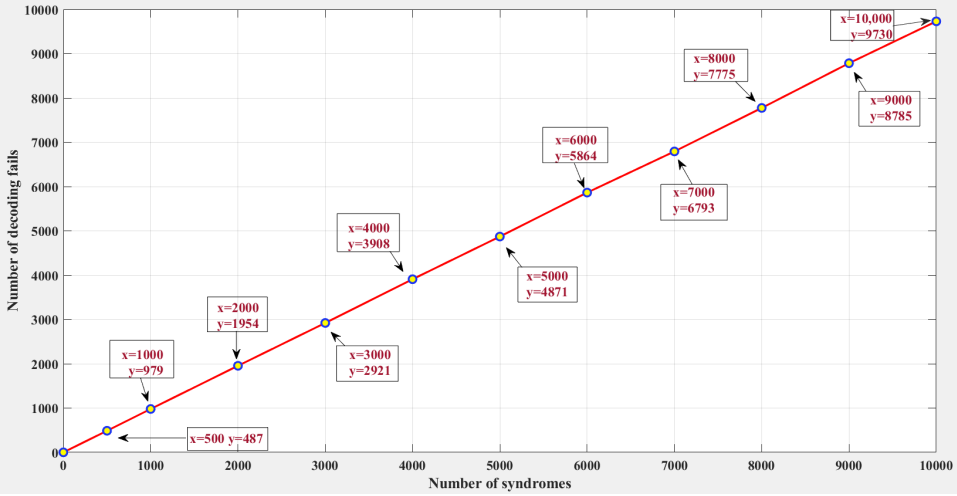

Fig 9. Number of syndromes simulated in BSC vs number of decoding fails

average number of times the code failed in decoding a syndrome across multiple simulators we calculate the mean failure count whose formula is given as:

$$Mean\ Failure\ Count = \frac{Total\ number\ of\ failures}{Total\ number\ of\ simulations}$$

In the AWGN channel, the mean failure count is 0.03, whereas in the BSC, it is 0.97, indicating that the proposed padding schemes are more effective at finding decodable syndromes in the AWGN channel as compared to the BSC.

4 Conclusion

This study presents six distinct padding methodologies to enhance the decoding of arbitrary syndromes through successive cancellation decoding. Unlike PolarSig, where the shortening of Polar codes was performed before use, this work uses the actual polar code parameters (N, K) and pads the syndrome to achieve successful syndrome decoding. In order to achieve high accuracy, these padding techniques have been implemented within the MATLAB environment. In addition, a comparative analysis of decoding performance in the Binary Symmetric Channel and Additive White Gaussian Noise channel is performed, which includes the success rate, failure rate, and mean failure count, and involves simulating about 10,000 random syndromes. Based on the experimental results, it has been demonstrated that the proposed padding techniques perform better in the AWGN channel than in the BSC, with a mean failure count of 0.03 and a decoding success rate exceeding 96%. By utilizing these decodable syndromes, polar codes may be utilized for the generation of code-based digital signatures.

5 Acknowledgment

The first author gratefully acknowledges the financial support from the “University Grant Commission (UGC), New Delhi, India”.

References

1) Aggarwal S, Kumar N. Chapter Four - Digital signatures. In: Advances in Computers;vol. 121. Academic Press Inc. 2021;p. 95–107. Available from: https://doi.org/10.1016/bs.adcom.2020.08.004.
2) Narwal E, Gill S. Simulating Manual Signature using Elman Back Propagation Model to Create Pseudo Digital Signature. *International Journal of Innovative Technology and Exploring Engineering*. 2019;9(2):3548–3551. Available from: https://www.ijitee.org/wp-content/uploads/papers/v9i2/B6452129219.pdf.
3) Rietsche R, Dremel C, Bosch S, Steinacker L, Meckel M, Leimeister JM. Quantum computing. *Electron Markets*. 2022;32:2525–2536. Available from: https://doi.org/10.1007/s12525-022-00570-y.
4) Sihotang HT, Efendi S, Zamzami EM, Mawengkang H. Design and Implementation of Rivest Shamir Adleman's (RSA) Cryptography Algorithm in Text File Data Security. In: International Conference on Advanced Information Scientific Development (ICAISD) 2020, 6-7 August 2020, West Java, Indonesia;vol. 1641 of Journal of Physics: Conference Series. IOP Publishing. 2020;p. 1–8. Available from: https://doi.org/10.1088/1742-6596/1641/1/012042.

5) Gueron S, Persichetti E, Santini P. Designing a Practical Code-Based Signature Scheme from Zero-Knowledge Proofs with Trusted Setup. *Cryptography*. 2022;6(1):1–17. Available from: https://doi.org/10.3390/cryptography6010005.
6) Balamurugan C, Singh K, Ganesan G, Rajarajan M. Post-quantum and code-based cryptography-Some Prospective Research Directions. *Cryptography*. 2021;5(4):1–30. Available from: https://doi.org/10.3390/cryptography5040038.
7) Liu X, Yang X, Han Y, Wang XA. A Secure and Efficient Code-Based Signature Scheme. *International Journal of Foundations of Computer Science*. 2019;30(04):635–645. Available from: https://doi.org/10.1142/S0129054119400173.
8) Perera MNS, Nakamura T, Hashimoto M, Yokoyama H, Cheng CMM, Sakurai K. A Survey on Group Signatures and Ring Signatures: Traceability vs. Anonymity. *Cryptography*. 2022;6(1):1–22. Available from: https://doi.org/10.3390/cryptography6010003.
9) Khurana R, Narwal E. Analysis of code-based digital signature schemes. *International Journal of Electrical and Computer Engineering (IJECE)*. 2023;13(5):5534–5541. Available from: http://doi.org/10.11591/ijece.v13i5.pp5534-5541.
10) Alahmadi A, Çalkavur S, Solé P, Khan AN, Raza MA, Aggarwal V. A New Code Based Signature Scheme for Blockchain Technology. *Mathematics*. 2023;11(5):1–12. Available from: https://doi.org/10.3390/math11051177.
11) Sahu R, Tripathi BP. A Code-Based Digital Signature Scheme Using Modified Quasi-Cyclic Low-Density Parity-Check Codes (QC-LDPC). *International Journal of Engineering and Advanced Technology*. 2019;8(6):2759–2763. Available from: https://www.ijeat.org/wp-content/uploads/papers/v8i6/F8822088619.pdf.
12) Lee Y, Lee W, Kim YS, No JS. Modified pqsigRM: RM Code-Based Signature Scheme. *IEEE Access*. 2020;8:177506–177518. Available from: https://ieeexplore.ieee.org/document/9206580.
13) Forghani P, Shooshtari MK, Aref MR. PolarSig: An efficient digital signature based on polar codes. *IET Communications*. 2020;14(17):2889–2897. Available from: https://digital-library.theiet.org/content/journals/10.1049/iet-com.2019.0578.
14) Hooshmand R, Jafari A, Karamali G. Id-PC: An Identification Scheme based on Polar Codes. *Information Security Journal: A Global Perspective*. 2023;32(4):283–296. Available from: https://doi.org/10.1080/19393555.2021.2023239.
15) Makoui FH, Gulliver TA, Dakhilalian M. A new code-based digital signature based on the McEliece cryptosystem. *IET Communications*. 2023;17(10):1199–1207. Available from: https://doi.org/10.1049/cmu2.12607.
16) Redhu R, Narwal E. Polar code-based cryptosystem: comparative study and analysis of efficiency. *Indonesian Journal of Electrical Engineering and Computer Science*. 2023;32(2):804–810. Available from: http://doi.org/10.11591/ijeecs.v32.i2.pp804-810.
17) Kaime IE, Madi AA, Erguig H. A Survey of Polar Codes. In: 2019 7th Mediterranean Congress of Telecommunications (CMT). IEEE. 2019. Available from: https://doi.org/10.1109/CMT.2019.8931392.
18) Hooshmand R, Shooshtari MK, Aref MR. PKC-PC: A variant of the McEliece public-key cryptosystem based on polar codes. *IET Communications*. 2020;14(12):1883–1893. Available from: https://doi.org/10.1049/iet-com.2019.0689.