IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 1, 2019, accepted November 15, 2019, date of publication December 12, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2959042

Modification of Frodokem Using Gray and Error-Correcting Codes

EUNSANG LEE[1], YOUNG-SIK KIM[2], (Member, IEEE) JONG-SEON NO[1], (Fellow, IEEE), MINKI SONG[3], AND DONG-JOON SHIN[3], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, INMC, Seoul National University, Seoul 08826, South Korea
[2]Department of Information and Communication Engineering, Chosun University, Gwangju 61452, South Korea
[3]Department of Electronics and Computer Engineering, Hanyang University, Seoul 04763, South Korea

Corresponding author: Young-Sik Kim (mypurist@gmail.com)

This work was supported by Samsung Research Funding and Incubation Center of Samsung Electronics under Project SRFC-IT1801-08.

ABSTRACT Lattice-based cryptography is one of the most promising schemes for post-quantum cryptography. Among the many lattice-based cryptosystems, FrodoKEM is a well-known key-encapsulation mechanism (KEM) based on (plain) learning with errors (LWE) problems and is advantageous in that the hardness is based on the problem of unstructured lattices. There are many cryptosystems that adopt error-correcting codes (ECCs) to improve performance, such as LAC, ThreeBears, and Round5. However, for lattice-based cryptosystems that do not use ring structures such as FrodoKEM, it is difficult to use ECCs because the number of transmitted symbols is small. In this study, we propose a method to apply ECCs and Gray codes to FrodoKEM by encoding the bits converted from the encrypted symbols. It is shown that the proposed method improves the security level and/or the bandwidth of FrodoKEM, and 192 message bits, 50% more than the original 128 bits, can be transmitted using one of the modified Frodo-640's.

INDEX TERMS Error-correcting codes (ECCs), FrodoKEM, Gray codes, key-encapsulation mechanism (KEM), lattice-based cryptography, learning with errors (LWE), post-quantum cryptography (PQC).

I. INTRODUCTION

Existing public key cryptosystems such as RSA and elliptic curve cryptography can be broken by future quantum computers because of the rapid development of quantum computers. Therefore, it is very important to develop secure post-quantum cryptography (PQC) algorithms resistant to quantum computing. Currently, NIST is in the process of proposing, evaluating, and standardizing PQC algorithms. In the first-round evaluation of algorithms submitted to NIST PQC Standardization (i.e., NIST PQC Round 1), 26 algorithms have been selected for NIST Round 2 and are under evaluation. Notably, 12 of the 26 algorithms are lattice-based ones. Thus, lattice-based cryptography is clearly the most promising field for PQC [1], [2].

Learning with errors (LWE) is a problem presented by Regev [3] in 2005 and is reduced to worst-case problems on lattices. Ring-LWE (RLWE) is a problem presented in [4], where it is reduced to worst-case problems on ideal lattices. RLWE significantly reduces the key size of cryptosystems based on LWE. Many lattice-based public-key encryption (PKE) and key-encapsulation mechanism (KEM) schemes submitted to NIST are based on the hardness of LWE and RLWE.

Among the lattice-based algorithms selected for NIST Round 2, many use error-correcting codes (ECCs) to improve their performances. NewHope [5] uses a simple error correction technique called additive threshold encoding (ATE). Round5 [6], which is a combined algorithm of Hila5 [7] and Round2 [8], uses an ECC called XE5 which is resistant to side-channel attacks. LAC [9], [10] uses BCH codes of large code lengths. ThreeBears [11] uses BCH codes such that constant-time implementation is possible, but its performance is relatively worse compared with those of the BCH codes used in LAC. A lattice-based KEM scheme called KCL [12], which was submitted to NIST PQC Round 1 but was not selected for Round 2, uses a single-error correcting code, lattice code in $\tilde{D}_4$ [13], or lattice code in E_8 [12]. However, all of these lattice-based algorithms using ECCs are ring-based schemes, and there is no case of using ECCs for non-ring ones such as FrodoKEM [14]–[16]. Here, we want to emphasize that our result is the first-ever one to apply and

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

analyze both ECCs and Gray coding to a non-ring lattice-based KEM, FrodoKEM. In addition, the application of ECCs and Gray coding to non-ring schemes such as FrodoKEM is not straightforward, and hence, various new ideas have been applied to the proposed results as explained here.

FrodoKEM is one of the representative PQC schemes selected for the NIST PQC Standardization Round 2. Therefore, improving the performance of FrodoKEM considerably important, and the proposed schemes in our study are possibly applied to other non-ring schemes. We aim to improve the performance of FrodoKEM as follows:

1) There is a risk that PQC cryptosystems will be broken because of the increasing computing power (era of quantum computers). Therefore, we are motivated to work on how to use ECCs to improve the security level of FrodoKEM so that FrodoKEM can resist enhanced computing power in the coming future.
2) In the IoT era, it is very important for cryptosystems to be able to send multiple keys simultaneously or to reduce the bandwidth. Therefore, we worked on how to use ECCs to increase the message size so that multiple keys can be simultaneously sent and to reduce the bandwidth.

In this study, we investigate how to apply ECCs to FrodoKEM [16] to improve its security level and/or lower its bandwidth. We propose a method to apply ECCs to FrodoKEM by encoding the bits converted from the encrypted symbols. In addition, we reduce the decryption failure rate (DFR) using Gray codes as bit-to-symbol mapping. Our method has the advantage of improving performances without modifying the existing framework of FrodoKEM. Note that the combination of ECCs and Gray coding is widely used in the field of wireless communication systems to lower the bit error probability in higher-order modulation such as pulse amplitude modulation (PAM) and quadrature amplitude modulation (QAM) [17]. However, although such a combination has been widely used, we, for the first time, apply it to a lattice-based PKE or KEM scheme. The symbols of $\mathbb{Z}_q$ considered in our study are similar to those of PAM, but modulo q operations should be performed after adding errors. We call it modulo q PAM. In addition, the environment of wireless communication systems is quite different from that of lattice-based PKE/KEM schemes as follows. In the case of PAM for wireless communications, when an error is added to a symbol with the largest magnitude, it is saturated rather than changed to another symbol. However, for the lattice-based PKE / KEM schemes, all the symbols of $\mathbb{Z}_q$ are computed by modulo q operations. For example, if an error 1 is added to the largest value $q-1$, we will obtain the smallest value 0.

The limited-magnitude error control codes [18] often adopt Gray coding. However, while the errors in the channel model for those codes are asymmetric and limited in their magnitude, the errors in the channel model for FrodoKEM are symmetric and not limited in their magnitude under mod q arithmetic.

In [19], the performance of NewHope was improved by using ECCs. The ATE technique used in NewHope is replaced by BCH codes or concatenated coding schemes of low-density parity check (LDPC) and BCH codes to improve the security level. However, the application of ECCs to FrodoKEM is quite different from [19] for the following reasons:

1) FrodoKEM does not use ECCs, and thus, we must carefully apply ECCs to FrodoKEM. To properly apply ECCs to FrodoKEM, we need to change some parameters of FrodoKEM, and thus, there is no guarantee that the performance will be improved as much as expected even if ECCs are used. We can improve the performance because we carefully select ECCs, change the parameter values, and use Gray coding with modulo q PAM.
2) Unlike the ring-based schemes such as NewHope, ECCs cannot be easily applied to non-ring schemes such as FrodoKEM because the number of symbols to which message bits is mapped to be very small. For example, NewHope and FrodoKEM have 1024 symbols and 64 symbols, respectively. Thus, it is not easy to design effective ECCs for the small number of message symbols in FrodoKEM. Furthermore, because multiple bits are mapped to one symbol in FrodoKEM, an error of one symbol can result in more than one error bit. It is complicated to calculate the DFR by considering more than one error bit. However, we show that the probability that more than one error occurs is relatively negligible when Gray coding is used. Thus, by applying the Gray code in our ECCs, one symbol error can be regarded as one error bit, which makes it possible to calculate the DFR.
3) Let us assume that normal bit-to-symbol mapping is a mapping in which both bit string and symbol size are in an increasing order. The mappings in (1), (2), and (7) in our paper are normal bit-to-symbol mappings. In contrast, the Gray mappings in (3) and (4) are not normal bit-to-symbol mappings. The bit strings are not in increasing order in these Gray mappings. When using normal bit-to-symbol mappings without Gray coding, the DFR becomes very high because more error bits occur without higher probability, and thus, the performance of FrodoKEM is not improved, as given in Tables 4, 6, and 7.

We propose and analyze combined schemes of ECCs and Gray coding in a non-ring scheme FrodoKEM for the first time; thus, we can substantially improve the performance of FrodoKEM. Our contributions can be summarized as follows:

1) We improve the security level of FrodoKEM by increasing the standard deviation σ of error distribution. Because the DFR increases as σ increases, the DFR requirement is satisfied by properly using Gray and BCH codes with modulo q PAM.
2) We increase the number of message bits of Frodo-640 from 128 bits to 192 bits while keeping the required

security level. If the number of message bits increases, σ should be reduced to meet the DFR requirement, which leads to degradation of the security level. Such security level degradation can be avoided by properly using Gray and BCH codes with modulo q PAM.

3) We reduce the bandwidth of FrodoKEM by using a smaller q. Because the DFR increases as q decreases, the DFR requirement is satisfied by properly using Gray and BCH codes with modulo q PAM.

The rest of the paper is organized as follows. FrodoKEM, parameters of FrodoKEM, and BCH codes are briefly explained in Section II. How to modify FrodoKEM to use Gray codes and ECCs and how to calculate the DFR are presented in Section III. The main results of improving the security level, increasing the message size, and reducing the bandwidth of FrodoKEM are given in Section IV. Finally, the discussion and future scope for studies are presented in Section V.

II. PRELIMINARIES

A. SUMMARY OF FrodoPKE ALGORITHM

This section reviews the underlying algorithm of FrodoKEM for its relevance to this study. For simplicity, we apply ECCs to FrodoPKE instead of FrodoKEM and calculate the DFR. FrodoKEM is the CCA transformation of FrodoPKE, and the DFR of FrodoKEM is the same as that of FrodoPKE [16]. The algorithms of FrodoPKE are described with the following parameters:

- χ; a probability distribution on $\mathbb{Z}$
- q; a power-of-two integer modulus
- $\bar{m}, \bar{n}, n$; integer matrix dimensions
- B; the number of bits in each symbol, where bits mean the codeword bits if ECC is used and the message bits, otherwise
- len_A; the length of seeds for pseudorandom matrix generation
- len_E; the length of seeds for pseudorandom bit generation for error sampling

Algorithm 1 FrodoPKE.KeyGen [16]

Input: None
Output: Key pair $(pk, sk) \in (\{0,1\}^{len_{\mathbf{A}}} \times \mathbb{Z}_q^{n\times\bar{n}}) \times \mathbb{Z}_q^{n\times\bar{n}}$

1 $seed_{\mathbf{A}} \leftarrow U(\{0,1\}^{len_{\mathbf{A}}})$;
2 Generate pseudorandom matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times n}$ with $seed_{\mathbf{A}}$;
3 $seed_{\mathbf{E}} \leftarrow U(\{0,1\}^{len_{\mathbf{E}}})$;
4 Generate matrices $\mathbf{S}, \mathbf{E} \in \mathbb{Z}_q^{n\times\bar{n}}$ from $seed_{\mathbf{E}}$ according to χ distribution;
5 Compute $\mathbf{B} = \mathbf{AS} + \mathbf{E}$;
6 Return public key $pk \leftarrow (seed_{\mathbf{A}}, \mathbf{B})$ and secret key $sk \leftarrow \mathbf{S}$;

Alice generates a public key and a private key through Algorithm 1 and sends the public key to Bob. Bob generates ciphertexts $\mathbf{C}_1$ and $\mathbf{C}_2$ through Algorithm 2 with the received public key and the message μ and then sends them to Alice. Finally, Alice computes $\mathbf{M} = \mathbf{C_2} - \mathbf{C_1}\,\mathbf{S}$ and restores the message μ sent by Bob.

Algorithm 2 FrodoPKE.Enc [16]

Input: Message $\mu \in \{0,1\}^{\bar{m}\bar{n}B}$ and public key $pk = (seed_{\mathbf{A}}, \mathbf{B}) \in \{0,1\}^{len_{\mathbf{A}}} \times \mathbb{Z}_q^{n\times\bar{n}}$
Output: Ciphertext $c = (\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{\bar{m}\times n} \times \mathbb{Z}_q^{\bar{m}\times\bar{n}}$

1 Generate pseudorandom matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times n}$ with $seed_{\mathbf{A}}$;
2 $seed_{\mathbf{E}} \leftarrow U(\{0,1\}^{len_{\mathbf{E}}})$;
3 Generate error matrices $\mathbf{S}', \mathbf{E}' \in \mathbb{Z}_q^{\bar{m}\times n}$ and $\mathbf{E}'' \in \mathbb{Z}_q^{\bar{m}\times\bar{n}}$ from $seed_{\mathbf{E}}$ according to χ distribution;
4 Compute $\mathbf{B}' = \mathbf{S}'\mathbf{A} + \mathbf{E}'$ and $\mathbf{V} = \mathbf{S}'\mathbf{B} + \mathbf{E}''$;
5 Return ciphertext $c \leftarrow (\mathbf{C_1}, \mathbf{C_2}) = (\mathbf{B}', \mathbf{V} + Frodo.Encode(\mu))$

Algorithm 3 FrodoPKE.Dec [16]

Input: $\mathbf{C_1}, \mathbf{C_2}, \mathbf{S}$
Output: μ '

1 Compute
2 $\mathbf{M} = \mathbf{C_2} - \mathbf{C_1}\,\mathbf{S} = \mathbf{V} + Frodo.Encode(\mu) -$
3 $(\mathbf{S}'\mathbf{A} + \mathbf{E}')\mathbf{S} = Frodo.Encode(\mu) +$
4 $\mathbf{S}'\mathbf{E} + \mathbf{E}'' - \mathbf{E}'\mathbf{S} = Frodo.Encode(\mu) + \mathbf{E}'''$;
5 Return $\mu' \leftarrow Frodo.Decode(\mathbf{M})$;

TABLE 1. Initial error distributions in FrodoKEM [16].

		Probability (in multiples of 2^{-16})											
	σ	0	±1	±2	±3	±4	±5	±6	±7	±8	±9	±10	±11
Frodo-640	2.75	9756	8857	7280	5249	3321	1844	898	384	144	47	13	3
Frodo-976	2.3	11278	10277	7774	4882	2545	1101	396	118	29	6	1	

The *Frodo.Encode* function in Algorithm 2 is defined as follows. *Frodo.Encode* takes a message $\mu \in \mathbb{Z}^{\bar{m}\bar{n}B}$ as input and outputs a matrix in $\mathbb{Z}_q^{\bar{m}\times\bar{n}}$. In Frodo-640 and Frodo-976, the values of B are 2 and 3, respectively, and B-bit messages are encoded into symbols in $\mathbb{Z}_q$ according to the following rules,

$$00 \to 0, 01 \to \frac{q}{4}, \quad 10 \to \frac{2q}{4}, 11 \to \frac{3q}{4} \tag{1}$$

$$000 \to 0, \quad 001 \to \frac{q}{8}, 010 \to \frac{2q}{8}, 011 \to \frac{3q}{8},$$
$$100 \to \frac{4q}{8}, \quad 101 \to \frac{5q}{8}, 110 \to \frac{6q}{8}, 111 \to \frac{7q}{8}. \tag{2}$$

The *Frodo.Decode* function in Algorithm 3 is defined as follows. In Frodo-640 and Frodo-976, Alice rounds each component of the matrix M to the nearest multiples of $q/4$ or $q/8$, respectively. Then, Alice obtains the message μ' by applying the inverse of mapping in (1) or (2) to each rounded symbol in $\mathbb{Z}_q$.

The initial error distributions χ of FrodoKEM are shown in Table 1 and are derived according to the following procedure. First, we obtain a Gaussian distribution with a given standard deviation σ. Next, a rounded Gaussian distribution

is derived from it. Finally, we obtain the error distribution of the small support that approximates the rounded Gaussian distribution [16].

Let ψ be the product distribution of the two initial error distributions. Let χ' be the error distribution obtained by convolving ψ $2n$ times and then convolving the resulting distribution with χ. Each component of $\mathbf{E}'''$ in Algorithm 3 follows the distribution χ', and the standard deviation of χ' is approximately $\sigma' \approx \sigma\sqrt{2n\sigma^2 + 1}$.

B. PARAMETERS OF FrodoKEM

Important FrodoKEM parameters are given in Table 2. Frodo-640 and Frodo-976 satisfy Categories 1 and 3 for security levels in the NIST PQC Standardization, respectively. How to compute the security level of FrodoKEM is described in [16]. In this study, a FrodoKEM source code is used to calculate the security level of various cases. Actually, the security level of the actual FrodoKEM is derived from a series of reductions, which is 5 or 6 bits smaller than the security level computed by the source code supported by FrodoKEM. Nevertheless, it is still meaningful to use this source code because our goal is not to obtain the accurate security level but to show improvement by using ECCs.

TABLE 2. Parameter sets of FrodoKEM [16].

	n	q	σ	B	$\bar{m} \times \bar{n}$	DFR	c size (bytes)
Frodo-640	640	2^{15}	2.75	2	8×8	$2^{-148.8}$	9736
Frodo-976	976	2^{16}	2.3	3	8×8	$2^{-199.6}$	15768

FrodoKEM has various parameters, $n, q, \sigma, B, \bar{m}$, and $\bar{n}$, which determine the bandwidth, computational complexity, the security level, and the DFR, respectively. To satisfy Categories 1 and 3 for security levels, it is recommended to set the security level higher than 128 and 192 bits, respectively. In addition, because there is an attack method by using decryption failure [20], the DFR should be low. Therefore, it is desirable that the DFRs be less than 2^{-128} and 2^{-192} for Categories 1 or 3 for security levels, respectively.

C. BCH CODES

BCH codes were developed in 1960. These codes can correct multiple errors and exhibit good error correction performance even for small code length. Relatively simple and feasible encoding and decoding techniques are also known, and hence, BCH codes have been widely used.

The code length of the BCH code is $n = q^m - 1$ for some prime q, and $q = 2$ holds for binary codes. In this study, we use binary BCH codes. BCH codes are usually denoted by (l_n, l_k, l_t), where l_n is the length of codeword, l_k is the length of message, and l_t is the error-correction capability, i.e., the maximum number of correctable errors.

The complexity of our scheme is simply the complexity of FrodoKEM plus the complexity of BCH encoding and decoding. The Peterson-Gorenstein-Zierler decoder algorithm has long been known for efficient decoding, and its complexity is $O(l_n l_t)$. BCH decoding algorithms do not usually have a constant computation time. However, using the method given in the recently published paper [21], we can implement constant-time decoding of the BCH code to defend some side-channel attacks.

III. MODIFICATION OF FrodoKEM WITH GRAY CODES AND ECCs

A. VIEWING FrodoPKE AS A DIGITAL COMMUNICATION SYSTEM

To apply ECCs to FrodoPKE and analyze them, it is convenient to understand the FrodoPKE in terms of digital communication systems, where messages are transmitted to the receiver via an encoder, modulator, (noisy) channel, demodulator, and decoder. Fig. 1 shows the description of Frodo-640 as a digital communication system.

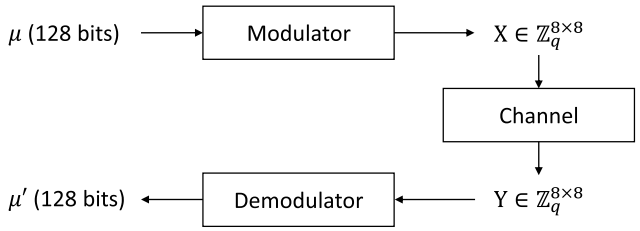


FIGURE 1. Description of Frodo-640 as a digital communication system.

In this model, the sender is Bob and the receiver is Alice. The shared key μ that Bob wants to send corresponds to the message bits. The mapping of binary bits to symbols in $\mathbb{Z}_q$ in FrodoPKE corresponds to modulation. *Frodo.Encode* function uses the term 'encode', but in fact it corresponds to a modulator in digital communication. In this study, we refer to *Frodo.Encode* in FrodoPKE as a modulator.

In FrodoPKE, Bob computes *Frodo.Encode*(μ) to generate two ciphertexts $\mathbf{C_1}$ and $\mathbf{C_2}$ and sends them. Alice computes $\mathbf{M} = \mathbf{C_2} - \mathbf{C_1}\,\mathbf{S}$ with the received $\mathbf{C_1}, \mathbf{C_2}$ and the secret key $\mathbf{S}$. As a result, *Frodo.Encode*(μ) is added with noise $\mathbf{E}'''$. This procedure can be seen as *Frodo.Encode*(μ) passing through a noisy channel in the digital communication. Here, the noise element of $\mathbf{E}'''$ follows χ' described in Section II. These noise elements are not i.i.d. However, because exact analysis is difficult, we assume that they are i.i.d.

The *Frodo.Decode* function works as follows. The *Frodo.Decode* function corresponds to a demodulator in the digital communication. *Frodo.Decode*($\mathbf{M}$) rounds each symbol in $\mathbb{Z}_q$ of the received matrix with errors to the nearest multiple of $q/4$ or $q/8$ for Frodo-640 and Frodo-976, respectively. Then, we apply the inverse of the mapping in (1) or (2) to obtain the estimated bit string μ'.

If we use ECCs in FrodoPKE, encoding is added before modulation, and decoding is added after demodulation. Fig. 2 shows the application of ECC to Frodo-640 as a digital communication system.

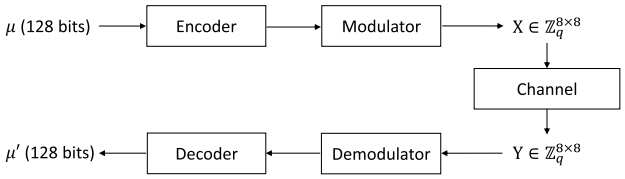


FIGURE 2. Description of Frodo-640 with ECC as a digital communication system.

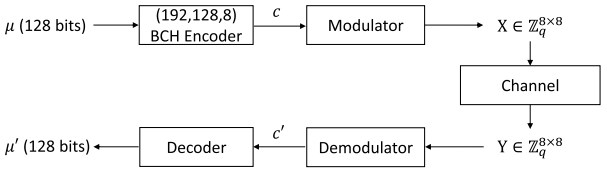


FIGURE 3. Frodo-640 with (192, 128, 8) BCH code.

B. ECC FOR FrodoKEM

To meet Categories 1 and 3 for security in NIST PQC Standardization, the obtained DFR should be less than 2^{-128} and 2^{-192}, respectively. Among various ECCs, we use algebraic codes, especially BCH codes, rather than modern codes such as LDPC codes for the following reasons:

1) LDPC codes have a serious error floor problem to be used for FrodoKEM. Because the error floor is reached quickly, the performance of LDPC codes is much worse than that of the algebraic codes for the region of DFR lower than 2^{-128}.
2) For LDPC codes, it is difficult to algebraically calculate the DFR, and thus, the DFR should be estimated through numerical analysis. However, in FrodoKEM, the DFR should be less than 2^{-128} or 2^{-192}, and numerical analysis in this error range is impossible. Reference [19] also could not calculate the DFR for LDPC code for this low DFR.
3) For the concatenated coding schemes of LDPC and algebraic codes as in [19], we need to know the statistical characteristics of errors remaining after LDPC decoding to algebraically estimate the DFR. Although it was not clearly stated in [19], they seem to assume that the errors remaining after LDPC decoding are statistically independent and uniformly distributed. However, the LDPC decoding errors tend to be bursty, and the analysis of characteristics of LDPC decoding errors is known to be a hard problem in the field of coding theory. In addition, the block error rate around 2^{-128} is the range where numerical analysis is impossible.

We use BCH codes because they provide various parameter values. Specifically, we use binary BCH codes with parameters (192, 128, 8), (256, 128, 18), (256, 192, 8), and (256, 192, 8). These are modified BCH codes obtained by shortening [22] or extending [23] the original BCH codes. For a systematic BCH code, l_t is not reduced from shortening. With extension, l_t is maintained or increased. Using these properties, we found these BCH code parameters suitable for FrodoKEM. Note that shortening and extending do not significantly affect the encoding and decoding algorithms and complexity.

C. GRAY CODING

It is well known that Gray coding should be used to map binary data to symbols from large alphabet for better bit error correction performance in digital communication.

For example, consider the case of applying the (192, 128, 8) BCH code to Frodo-640 as in Fig. 3. Encoding the 128-bit message μ results in a 192-bit codeword c. In the modulation with $B = 3$, each of the three bits in the codeword c is mapped to a symbol in $\mathbb{Z}_q$ according to the following Gray coding, which is different from the mapping in (2),

$$\begin{aligned} &000 \to 0, \quad 001 \to \frac{q}{8}, \; 011 \to \frac{2q}{8}, \; 010 \to \frac{3q}{8}, \\ &110 \to \frac{4q}{8}, \quad 111 \to \frac{5q}{8}, \; 101 \to \frac{6q}{8}, \; 100 \to \frac{7q}{8}. \end{aligned} \tag{3}$$

Gray coding in (3) is depicted in Fig. 4, where dotted lines denote decision boundaries for demodulation. Fig. 4 shows that the bit difference between adjacent symbols is always 1 bit in Gray coding. The reason for using Gray coding is to minimize the number of bit errors and increase the error correction probability of ECCs. Because Gray coding does not improve the symbol error rate (SER), there is no reason to consider Gray coding if an ECC is not used, and thus, the original Frodo-640 does not have to use Gray coding.

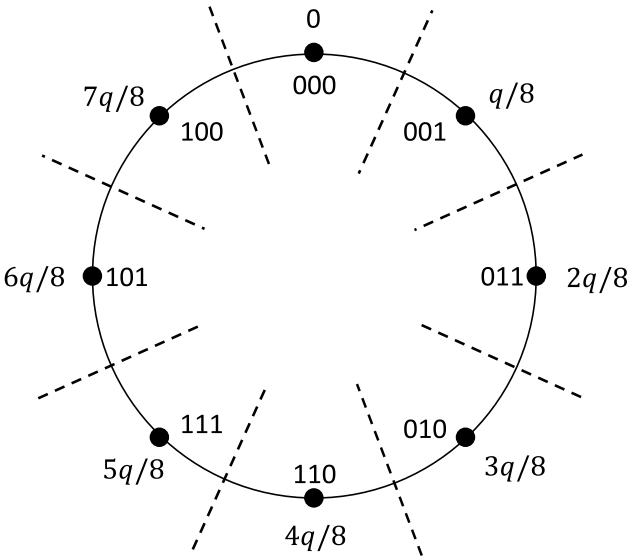


FIGURE 4. Gray coding for $B = 3$.

The 192-bit codeword is mapped to 64 symbols in $\mathbb{Z}_q$, and errors are added to these symbols in $\mathbb{Z}_q$ while passing through the channel. Demodulation rounds each symbol in $\mathbb{Z}_q$ to the nearest multiples of $q/8$ and then applies the inverse of the mapping in (3). Then, we obtain c', which is the codeword c added with errors. Then, BCH decoding is performed to obtain c' for estimating μ'.

Gray coding for $B = 4$ is performed as follows

$$\begin{aligned}
&0000 \to 0, \quad 0001 \to \frac{q}{16}, \ 0011 \to \frac{2q}{16}, \ 0010 \to \frac{3q}{16},\\
&0110 \to \frac{4q}{16}, \quad 0111 \to \frac{5q}{16}, \ 0101 \to \frac{6q}{16}, \ 0100 \to \frac{7q}{16},\\
&1100 \to \frac{8q}{16}, \quad 1101 \to \frac{9q}{16}, \ 1111 \to \frac{10q}{16}, \ 1110 \to \frac{11q}{16},\\
&1010 \to \frac{12q}{16}, \quad 1011 \to \frac{13q}{16}, \ 1001 \to \frac{14q}{16}, \ 1000 \to \frac{15q}{16}.
\end{aligned} \tag{4}$$

D. IND-CCA SECURITY OF MODIFIED FrodoKEM

We apply ECCs to FrodoPKE as shown in Fig. 2. We add BCH encoder *BCH.Encode* before *Frodo.Encode* which corresponds to modulator and BCH decoder *BCH.Decode* after *Frodo.Decode* which corresponds to demodulator. Specific BCH parameters we use are shown in Section III-B. In addition, Gray coding in (3) or (4) is used instead of the existing *Frodo.Encode* function, and *Frodo.Decode* is replaced with another function as shown in Section III-C. Let us call the modified *Frodo.Encode* *Frodo.Encode′* and the modified FrodoPKE FrodoPKE'.

In this study, we describe FrodoPKE' and analyze the performance of our modified FrodoKEM scheme through the analysis of FrodoPKE' without the description of our KEM scheme for simplicity. Our modified FrodoKEM is derived from FrodoPKE' using QFO transformation, similar to that in a previous study [16]. The description of our KEM scheme can be omitted for the following two reasons. First, the QFO transformation method described in [16] can be used almost identically to construct our modified KEM. In addition, it is possible to analyze the performance of our modified FrodoKEM according to the change in parameters only by describing the FrodoPKE'.

Consider the IND-CCA security of our modified KEM scheme. The IND-CCA security proof of FrodoKEM shown in [16] is summarized as follows. They proved that FrodoPKE achieves IND-CPA security and then proved that FrodoKEM modified using QFO transformation achieves IND-CCA security. Similarly, if FrodoPKE' is proved to achieve IND-CPA security, then our modified FrodoKEM can also achieve IND-CCA security. The proposition that FrodoPKE' has IND-CPA security can be proved by the fact that FrodoPKE achieves IND-CPA security as follows.

1) Suppose there is an algorithm $\mathcal{A}$ that attacks FrodoPKE'. Let us design an algorithm $\mathcal{A}'$ that uses $\mathcal{A}$ to attack FrodoPKE.
2) $\mathcal{A}$ claims to be able to distinguish between ciphertexts of μ_0 and μ_1.
3) $\mathcal{A}'$ puts the following values into FrodoPKE as inputs: $Frodo.Encode^{-1}[Frodo.Encode'(BCH.Encode(\mu_i))]$ for $i = 0$ and 1. $\mathcal{A}'$ receives the ciphertexts c_0 and c_1. Then, c_0 and c_1 correspond to ciphertexts of μ_0 and μ_1 in FrodoPKE'.
4) $\mathcal{A}'$ passes the received c_0 and c_1 to $\mathcal{A}$ to distinguish.
5) $\mathcal{A}$ distinguishes c_0 and c_1, and computes $b \in \{0, 1\}$.
6) $\mathcal{A}'$ distinguishes the ciphertexts of the original messages μ_0 and μ_1 by outputting b calculated by $\mathcal{A}$.

Since our scheme simply adds BCH encoding and BCH decoding algorithms to FrodoKEM, there is no danger of being particularly vulnerable to primal and dual attacks, which are simply LWE attacks.

TABLE 3. Comparison between the probability of crossing the decision boundary once and that of crossing the decision boundary twice.

	σ	B	the probability of crossing decision boundary once	the probability of crossing decision boundary twice
Case 1)	3.38	3	$2^{-20.42}$	$2^{-72.73}$
Case 2)	2.87	4	$2^{-10.72}$	$2^{-36.73}$
Case 3)	2.84	4	$2^{-26.06}$	$2^{-94.52}$
Case 4)	1.93	3	$2^{-155.62}$	$2^{-528.52}$
Case 5)	1.78	3	$2^{-203.53}$	.
Case 6)	2.38	4	$2^{-20.45}$	$2^{-72.87}$
Case 7)	2.22	4	$2^{-26.15}$	$2^{-93.92}$
Case 8)	2.3	2	$2^{-82.42}$	$2^{-291.49}$
Case 9)	2.3	3	$2^{-23.04}$	$2^{-82.41}$
Case 10)	2.3	3	$2^{-7.00}$	$2^{-23.04}$

E. EVALUATING THE DFR

In FrodoKEM, 64 symbols in $\mathbb{Z}_q$ are transmitted. Given the parameters n, q, σ, B, and the maximum number of correctable errors t using an ECC ($t = 0$ if ECC is not used), the DFR is computed through the following procedures:

1) Find an optimal discrete noise distribution χ that approximates the rounded continuous Gaussian for a given σ. Find the optimal distribution considering attack time for the LWE problem.
2) Obtain the product distribution ψ of two optimal distributions χ. Convolve ψ $2n$ times and then convolve the resulting distribution with χ. Then, we obtain a distribution χ'.
3) Compute the SER p with χ' as follows

$$\text{SER} = \sum_{x \in [q/2^{B+1}, q-q/2^{B+1})} \chi'(x). \tag{5}$$

4) Compute the DFR of FrodoKEM using an ECC as follows. When an ECC with the error correction capability t is used, Gray coding should be used. The 64 B-bit messages are encoded into 64 symbols in $\mathbb{Z}_q$, with the probability p that an error occurs for each symbol. When Gray coding is used, most symbol errors only generate one-bit errors. For example, Fig. 4 shows that only a one-bit error occurs when the decision boundary is crossed once. For various cases that will be described in Section IV, Table 3 shows the comparison between the probability of crossing the decision boundary once and that of crossing the decision boundary twice. The probability of crossing the decision boundary twice is relatively negligible, as shown in Table 3, and in Case 5), it is too small; thus, we cannot obtain the value. The DFR is the probability that more than t symbol

TABLE 4. Cases for improving the security level of FrodoKEM scheme.

	message size [bits]	σ	security level [bits]	SER with Gray coding	DFR with Gray coding	SER without Gray coding	DFR without Gray coding
Frodo-640	128	2.75	149.30	$2^{-154.82}$	$2^{-148.82}$	.	.
Frodo-976	192	2.3	215.66	$2^{-205.56}$	$2^{-199.56}$	.	.
Case 1)	128	3.38	156.98	$2^{-20.42}$	$2^{-149.05}$	$2^{-20.42}$	$2^{-51.90}$
Case 2)	128	2.87	152.25	$2^{-10.72}$	$2^{-150.71}$	$2^{-10.72}$	$2^{-50.77}$
Case 3)	192	2.84	225.97	$2^{-26.06}$	$2^{-199.88}$	$2^{-26.06}$	$2^{-68.52}$

errors out of 64 symbols occur, which can be obtained using the following equation

$$\mathrm{DFR} = \sum_{i=t+1}^{64} \binom{64}{i} p^i (1-p)^{64-i}. \tag{6}$$

When using an ECC, if you do not use Gray coding, you can use the mapping in (2) for $B = 3$ and the following mapping for $B = 4$,

$$\begin{aligned}
&0000 \to 0, \quad 0001 \to \frac{q}{16}, \ 0010 \to \frac{2q}{16}, \ 0011 \to \frac{3q}{16},\\
&0100 \to \frac{4q}{16}, \quad 0101 \to \frac{5q}{16}, \ 0110 \to \frac{6q}{16}, \ 0111 \to \frac{7q}{16},\\
&1000 \to \frac{8q}{16}, \quad 1001 \to \frac{9q}{16}, \ 1010 \to \frac{10q}{16}, \ 1011 \to \frac{11q}{16},\\
&1100 \to \frac{12q}{16}, \quad 1101 \to \frac{13q}{16}, \ 1110 \to \frac{14q}{16}, \ 1111 \to \frac{15q}{16}.
\end{aligned} \tag{7}$$

Mappings in (1) and (2) are given in FrodoKEM. However, there is no bit-to-symbol mapping for $B = 4$ in FrodoKEM; thus, we made the bit-to-symbol mapping in (7) for $B = 4$ in the same way as the mappings in (1) and (2).

In this paper, we compute and compare the DFR of FrodoKEM using an ECC with and without Gray coding. The following procedure depicts how to compute the DFR of FrodoKEM using an ECC without Gray coding. Let p_1, p_2, p_3, and p_4 be the probability values that the number of bit errors in one symbol in $\mathbb{Z}_q$ after demodulation is 1, 2, 3, and 4, respectively. Let n_i, $i \in 1, 2, 3, 4$, be the number of symbols in $\mathbb{Z}_q$ such that the number of bit errors in a symbol is i. Suppose the codeword bits are uniform at random. Then, p_1, p_2, and p_3 are approximately $p/2$, $p/4$, and $p/4$ when $B = 3$, respectively. In addition, p_1, p_2, p_3, and p_4 are approximately $p/2$, $p/4$, $p/8$, and $p/8$ when $B = 4$, respectively. Then, the DFRs of FrodoKEM using ECCs without Gray coding are obtained for $B = 3$ and $B = 4$ as follows, respectively,

$$\sum_{n_1+2n_2+3n_3>t} \binom{64}{n_1, n_2, n_3, 64-n_1-n_2-n_3} \times {p_1}^{n_1} {p_2}^{n_2} {p_3}^{n_3} (1-p)^{64-n_1-n_2-n_3} \tag{8}$$

$$\sum_{n_1+2n_2+3n_3+4n_4>t} \binom{64}{n_1, n_2, n_3, n_4, 64-n_1-n_2-n_3-n_4} \times {p_1}^{n_1} {p_2}^{n_2} {p_3}^{n_3} {p_4}^{n_4} (1-p)^{64-n_1-n_2-n_3-n_4}. \tag{9}$$

F. ERROR DEPENDENCY

Because it is very difficult to derive the DFR considering error dependency, the DFR is usually computed assuming that errors are statistically independent. Recent studies have reported that the DFR was underestimated when using ECC [24] because error dependency is not considered, and they proposed a DFR calculation method for LAC considering error dependency. However, their DFR calculation method cannot be applied to other schemes including FrodoKEM. We compute the DFR based on the assumption that the error coefficients of $\mathbf{S}'\mathbf{E} + \mathbf{E}'' - \mathbf{E}'\mathbf{S}$ in Algorithm 3 are independent. Considering the DFR deviation due to the independence assumption, we try to set a margin that is enough in the DFRs of the proposed improvements. The exact analysis considering error dependency will be performed in future work.

IV. IMPROVING FrodoKEM USING GRAY AND ERROR-CORRECTING CODES

A. IMPROVING THE SECURITY LEVEL OF FrodoKEM

We use BCH codes to improve the security levels of Frodo-640 and Frodo-976, and B should be increased to use ECCs. Because of the increase in B, the DFR also increases. However, using BCH codes, the DFR can be effectively lowered. We improve the security level by maximizing σ while satisfying DFR $< 2^{-148.8}$ for Frodo-640 and DFR $< 2^{-199.6}$ for Frodo-976. The parameters n, q, $\bar{n}$, and $\bar{m}$ are maintained, but only σ and B are adjusted. Then, the security level can be improved while maintaining the bandwidth and satisfying the DFR requirement. Table 4 summarizes the performances of various cases with and without Gray coding to improve the security level as explained below.

Case 1) FRODO-640 WITH (192, 128, 8) BCH CODE

Encode the 128-bit message with (192, 128, 8) BCH code to obtain a 192-bit codeword c. Modulates c using Gray coding in (3), and then, c passes through the channel. B, σ, and the security level are changed as follows:

- B; $2 \to 3$
- σ; $2.75 \to 3.38$
- security level; $149.30 \to 156.98$
- SER $= 2^{-20.42}$
- DFR = error probability after BCH decoding = probability of more than 8 errors $= 2^{-149.05}$.

If we use the mapping in (2) instead of Gray coding in (3), SER and DFR are calculated as follows

TABLE 5. Comparison of Frodo-640 with (192, 128, 8) BCH code and Frodo-640 with increased *n*.

	security level [bits]	DFR	public key size [bytes]	ciphertext size [bytes]
Frodo-640 with (192, 128, 8) BCH code	$149 \rightarrow 157$	$2^{-149.05}$	9616	9736
Frodo-640 $n : 640 \rightarrow 700$	$149 \rightarrow 166$	$2^{-137.31}$	10516	10636

- SER $= 2^{-20.42}$
- DFR $= 2^{-51.90}$.

From these results, we can see that the security level is increased by applying a BCH code to Frodo-640. In addition, Gray coding is essential because when Gray coding is not used, the DFR is much higher than the DFR when Gray coding is used.

Note that the security level can also be improved by simply increasing n without using the BCH code. However, the bandwidth also increases as n increases. Table 5 compares Frodo-640 with (192, 128, 8) BCH code and Frodo-640 with increased n while all parameters other than B and n are unchanged. In Frodo-640 with increased n, the security level is improved, but the bandwidth also increases. Case 1) improves the security level while maintaining the bandwidth.

Case 2) FRODO-640 WITH (256, 128, 18) BCH CODE

Encode the 128-bit message with the (256, 128, 18) BCH code to obtain a 256-bit codeword c. Modulate c using Gray coding in (4), and then, c passes through the channel. B, σ, and the security level are changed as follows:

- B; $2 \rightarrow 4$
- σ; $2.75 \rightarrow 2.87$
- security level; $149.30 \rightarrow 152.25$
- SER $= 2^{-10.72}$
- DFR = error probability after BCH decoding = probability of more than 18 errors $= 2^{-150.71}$.

If we use the mapping in (7) instead of Gray coding in (4), SER and DFR can be calculated as follows:

- SER $= 2^{-10.72}$
- DFR $= 2^{-50.77}$.

From these results, we can see that the security level is increased by applying a BCH code to Frodo-640. However, the security level of Case 2) is less than the security level of Case 1). In addition, Gray coding is essential similar to Case 1).

Case 3) FRODO-976 WITH (256, 192, 8) BCH CODE

Encode the 192-bit message with the (256, 192, 8) BCH code to obtain a 256-bit codeword c. Modulate c using Gray coding in (4), and then, c passes through the channel. B, σ, and the security level are changed as follows:

- B; $3 \rightarrow 4$
- σ; $2.3 \rightarrow 2.84$
- security level; $215.66 \rightarrow 225.97$
- SER $= 2^{-26.06}$
- DFR = error probability after BCH decoding = probability of more than 8 errors $= 2^{-199.88}$.

If we use the mapping in (7) instead of Gray coding in (4), SER and DFR are calculated as follows:

- SER $= 2^{-26.06}$
- DFR $= 2^{-68.52}$.

From these results, we can see that the security level is increased by applying a BCH code to Frodo-976. In addition, Gray coding is essentially similar to the previous cases.

B. INCREASING MESSAGE SIZE IN FRODO-640

In this section, we use the 192-bit message for Frodo-640 instead of the 128-bit message because increasing the message bits from 128 bits to 192 bits has several advantages. The 128-bit key for symmetric key encryption and 64-bit key for authentication can be sent at the same time via the 192-bit key. In addition, if we use the 80-bit lightweight cryptographic keys for IoT systems, we can send two 80-bit keys at once.

In the following Cases 5) and 6), 256-bit codewords are mapped to the transmitted matrix in $\mathbb{Z}_q^{8\times 8}$, and thus, B is increased from 2 to 4. Because of the increase in B, the DFR also increases, and we should decrease σ to satisfy the DFR requirement. Then, the security level significantly decreases. However, it is possible to prevent degradation of the security level by using BCH codes. Case 4) uses the 192-bit message without using the BCH code. However, the security level significantly decreases. Cases 5) and 6) using BCH codes can satisfy DFR $< 2^{-148.82}$ and DFR $< 2^{-199.56}$, respectively. Table 6 summarizes the performances of various cases with and without Gray coding to increase the message as explained below.

Case 4) FRODO-640, MESSAGE; 128 $\rightarrow$192 BITS, DFR $< 2^{-148.82}$

The 192-bit message μ is modulated using the mapping in (2), and then, it passes through the channel. Then, B, σ, and the security level are changed as follows:

- B; $2 \rightarrow 3$
- σ; $2.3 \rightarrow 1.93$
- security level; $149.30 \rightarrow 137.18$
- SER $= 2^{-155.62}$
- DFR $= 2^{-149.62}$.

We can see that increasing the message size decreases the security level significantly.

Case 5) FRODO-640, MESSAGE; 128 $\rightarrow$192 BITS, DFR $< 2^{-199.56}$

The 192-bit message μ is modulated using the mapping in (2), and then, it passes through the channel. Then, B, σ, and the security level are changed as follows:

- B; $2 \rightarrow 3$
- σ; $2.3 \rightarrow 1.78$
- security level; $149.30 \rightarrow 134.52$
- SER $= 2^{-207.62}$
- DFR $= 2^{-201.62}$.

TABLE 6. Cases for increasing message size of the FrodoKEM scheme.

	message size [bits]	σ	security level [bits]	SER with Gray coding	DFR with Gray coding	SER without Gray coding	DFR without Gray coding
Frodo-640	128	2.75	149.30	$2^{-154.82}$	$2^{-148.82}$	.	.
Frodo-976	192	2.3	215.66	$2^{-205.56}$	$2^{-199.56}$	.	.
Case 4)	192	1.93	137.18	$2^{-155.62}$	$2^{-149.62}$	.	.
Case 5)	192	1.78	134.52	$2^{-207.62}$	$2^{-201.62}$	.	.
Case 6)	192	2.38	144.27	$2^{-20.45}$	$2^{-149.35}$	$2^{-20.45}$	$2^{-51.68}$
Case 7)	192	2.22	141.91	$2^{-26.15}$	$2^{-200.70}$	$2^{-26.15}$	$2^{-68.79}$

We can see that increasing the message size decreases the security level significantly.

Case 6) FRODO-640 WITH (256, 192, 8) BCH CODE, MESSAGE; 128 $\rightarrow$192 BITS, DFR $< 2^{-148.82}$

Encode the 192-bit message with the (256, 192, 8) BCH code to obtain the 256-bit codeword c. c is modulated using Gray coding in (4), and it passes through the channel. B, σ, and the security level are changed as follows:

- B; $2 \rightarrow 4$
- σ; $2.75 \rightarrow 2.38$
- security level; $149.30 \rightarrow 144.27$
- SER $= 2^{-20.45}$
- DFR = error probability after BCH decoding = probability of more than 8 errors $= 2^{-149.35}$.

If we use the mapping in (7) instead of Gray coding in (4), SER and DFR are calculated as follows:

- SER $= 2^{-20.45}$
- DFR $= 2^{-51.68}$.

Even though a BCH code is used, increasing the message size while maintaining the DFR reduces the security level. However, in Case 6), the security level does not deteriorate that much as compared to Case 4). In addition, it is clear that Gray coding is essential.

Case 7) FRODO-640 WITH (256, 192, 8) BCH CODE, MESSAGE; 128 $\rightarrow$192 BITS, DFR $< 2^{-199.56}$

Encode the 192-bit message with the (256, 192, 8) BCH code to obtain the 256-bit codeword c. c is modulated using Gray coding in (4), and then, it passes through the channel. B, σ, and the security level are changed as follows:

- B; $2 \rightarrow 4$
- σ; $2.75 \rightarrow 2.22$
- security level; $149.30 \rightarrow 141.91$
- SER $= 2^{-26.15}$
- DFR = error probability after BCH decoding = probability of more than 8 errors $= 2^{-200.70}$.

If we use the mapping in (7) instead of Gray coding in (4), SER and DFR are calculated as follows:

- SER $= 2^{-26.15}$
- DFR $= 2^{-68.79}$.

Even though a BCH code is used, increasing the message size while satisfying DFR $< 2^{-199.56}$ reduces the security level. However, in Case 7), the security level does not reduce much compared to Case 5). In addition, it is clear that Gray coding is essential.

TABLE 7. Cases for reducing the bandwidth of FrodoKEM scheme.

	q	B	σ	public key [bytes]	ciphertext [bytes]	security level [bits]	DFR with Gray coding	DFR without Gray coding
Frodo-640	32768	2	2.75	9616	9736	149.30	$2^{-148.82}$	.
Case 8)	16384	2	2.3	8976	9088	156.39	$2^{-76.41}$	.
Case 9)	16384	3	2.3	8976	9088	156.39	$2^{-172.63}$	$2^{-59.76}$
Case 10)	8192	3	2.3	8336	8440	172.33	$2^{-28.84}$	$2^{-10.98}$

C. REDUCING THE BANDWIDTH

The bandwidth of Frodo-640 can be reduced by reducing q. Then, σ should be reduced to keep the DFR low because reducing q will increase the DFR. However, there are limits to reducing σ. To meet the bounded distance decoding with the discrete Gaussian sampling (BDDwDGS) reduction requirement, σ should be larger than 2.3 [16].

To reduce the bandwidth of Frodo-640, we can reduce q by half and improve the security level while satisfying the condition $\sigma \geq 2.3$ using BCH codes, where the DFR still meets the requirement. Table 7 summarizes the performances of the following cases for reducing the bandwidth of FrodoKEM schemes.

Case 8) FRODO-640, q; 32768 $\rightarrow$ 16384, σ; 2.75 $\rightarrow$ 2.3

We reduce q into half without using the BCH code and reduce σ as much as possible to decrease the DFR such as $\sigma = 2.3$. Then, SER and DFR are calculated as follows:

- SER is $2^{-82.42}$
- DFR is $2^{-76.41}$.

From these results, we can see that the DFR requirement cannot be satisfied by simply reducing q without using the BCH code.

Case 9) FRODO-640 WITH (192, 128, 8) BCH CODE, q; 32768 $\rightarrow$ 16384, σ; 2.75 $\rightarrow$ 2.3

In this case, $q = 16384$, $\sigma = 2.3$, $B = 3$, and the (192, 128, 8) BCH code are used. We encode the 128-bit message with the (192, 128, 8) BCH code to obtain the 192-bit codeword c. Then, the codeword c is modulated using Gray coding in (3), and it passes through the channel. Then, SER and DFR are calculated as follows:

- SER $= 2^{-23.04}$
- DFR $= 2^{-172.63}$.

In this case, the DFR is lower than the DFR requirement $2^{-148.82}$, and the bandwidth of Frodo-640 can be decreased while satisfying the DFR requirement.

- public key; 9616 bytes → 8976 bytes
- ciphertext; 9736 bytes → 9088 bytes

At this point, the security level is improved.

- security level; 149.30 → 156.39

If we use the mapping in (2) instead of Gray coding in (3), SER and DFR are calculated as follows:

- SER $= 2^{-23.04}$
- DFR $= 2^{-59.76}$.

From these results, we can see that the security level can be improved and the bandwidth can be reduced while satisfying the DFR requirement. In addition, it is clear that Gray coding is essential.

Case 10) FRODO-640 WITH (192, 128, 8) BCH CODE, q; 32768 → 8192, σ; 2.75 → 2.3

In this case, $q = 8192$, $\sigma = 2.3$, $B = 3$, and the (192, 128, 8) BCH code are used. We encode the 128-bit message with the (192, 128, 8) BCH code to obtain a 192-bit codeword c. Then, the codeword c is modulated using Gray coding in (3), and it passes through the channel. Then, SER and DFR are calculated as follows:

- SER $= 2^{-7.00}$
- DFR $= 2^{-28.84}$.

In this case, the DFR is higher than the requirement $2^{-148.82}$. If we use the mapping in (2) instead of Gray coding in (3), SER and DFR are calculated as follows:

- SER $= 2^{-7.00}$
- DFR $= 2^{-10.98}$.

From these results, we can see that if q is reduced to 8192, then the DFR requirement cannot be satisfied even if the BCH code is used.

V. DISCUSSION AND FUTURE WORKS

Previously, ECCs have been used only for ring-based PKE/KEM schemes. In this study, however, we applied ECCs and Gray codes to FrodoKEM, which is a non-ring scheme. Especially, we improved FrodoKEM by carefully combining BCH codes and Gray coding with modulo q PAM as follows:

1) The security level is improved by about 7 bits and 10 bits in Frodo-640 and Frodo-976, respectively.
2) Instead of 128-bit message, a 192-bit message is loaded on Frodo-640 while maintaining the required security level.
3) The bandwidth is reduced by 1288 bytes because of the decreasing q, while DFR and BDDwDGS reduction requirements are satisfied.

Because we conducted the analysis assuming that the errors are independent, it is necessary to study how to accurately calculate the DFR. Reference [24] provided a method to calculate the DFR considering error dependency. However, the method is applicable only to LAC, and further research is needed to accurately calculate the DFR of other schemes by considering error dependency including FrodoKEM. In addition, we mentioned in Section III-B that we did not use LDPC codes because it is difficult to analyze the exact statistical characteristics of errors remaining after LDPC decoding. Therefore, it may be a good future work to analyze the accurate DFR for cases using LDPC codes.

The modified technique proposed here, which uses the combination of ECCs and Gray coding, is applicable not only to FrodoKEM but also to other ring/non-ring lattice-based KEM/PKE schemes such as Kyber [25], [26], Round5, and LAC.

ACKNOWLEDGMENT

We would like to thank anonymous reviewers and the associate editor for their valuable suggestions and comments that helped improve the quality of this paper.

REFERENCES

[1] D. Micciancio, "Lattice-based cryptography," in *Post-Quantum Cryptography* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2011, pp. 147–191.

[2] C. Peikert, "A decade of lattice cryptography," *Found. Trends Theor. Comput. Sci.*, vol. 10, no. 4, pp. 283–424, 2016.

[3] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, pp. 1–37, 2009.

[4] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 6110. Berlin, Germany: Springer, 2010, pp. 1–23.

[5] T. Poppelmann, E. Alkim, R. Avanzi, J. Bos, L. Ducas, A. Piedra, P. Schwabe, and D. Stebila. *NewHope*. Accessed: Dec. 13, 2019. [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/Round-1-Submissions

[6] S. Bhattacharya, O. Garcia-Morchon, T. Laarhoven, R. Rietman, M. O. Saarinen, L. Tolhuizen, and Z. Zhang, "Round5: Compact and fast post-quantum public-key encryption," IACR, Lyon, France, Tech. Rep. 2019/090, 2019. Accessed: Dec. 13, 2019. [Online]. Available: https://eprint.iacr.org/2019/090

[7] M. O. Saarinen, "HILA5: On reliability, reconciliation, and error correction for Ring-LWE encryption," in *Proc. Int. Conf. Sel. Areas Cryptogr.*, in Lecture Notes in Computer Science, vol. 10719. Berlin, Germany: Springer, 2017, pp. 192–212.

[8] O. Garcia-Morchon, Z. Zhang, S. Bhattacharya, R. Rietman, L. Tolhuizen, J. Torre-Arce, and H. Baan. *Round2*. Accessed: Dec. 13, 2019. [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/Round-1-Submissions

[9] X. Lu, Y. Liu, D. Jia, H. Xue, J. He, and Z. Zhang. *LAC*. Accessed: Dec. 13, 2019. [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/Round-1-Submissions

[10] X. Lu, Y. Liu, Z. Zhang, D. Jia, H. Xue, J. He, and B. Li, "LAC: Practical ring-LWE based public-key encryption with byte-level modulus," IACR, Lyon, France, Tech. Rep. 2018/1009, 2018. Accessed: Dec. 13, 2019. [Online]. Available: https://eprint.iacr.org/2018/1009

[11] M. Hamburg. *Three Bears*. Accessed: Dec. 13, 2019. [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/Round-1-Submissions

[12] Y. Zhao, Z. Jin, B. Gong, and G. Sui. *KCL*. Accessed: Dec. 13, 2019. [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/Round-1-Submissions

[13] E. Alkim, L. Ducas, T. Poppelmann, and P. Schwabe, "Post-quantum key exchange-a new hope," in *Proc. 25th USENIX Secur. Secur.*, Santa Clara, CA, USA, 2016, pp. 327–343.

[14] J. Bos, C. Costello, L. Ducas, L. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila, "Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1006–1018.

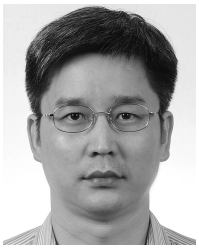
[15] R. Lindner and C. Peikert, "Better key sizes (and attacks) for LWE-based encryption," in *Proc. Cryptographers' Track RSA Conf.*, in Lecture Notes in Computer Science, vol. 6558, 2011, pp. 319–339.
[16] M. Naehrig, E. Alkim, J. Bos, L. Ducas, K. Easterbrook, B. LaMacchia, P. Longa, I. Mironov, V. Nikolaenko, C. Peikert, A. Raghunathan, and D. Stebila. *FrodoKEM*. Accessed: Dec. 13, 2019. [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/Round-1-Submissions
[17] J. G. Proakis and M. Salehi, *Communication Systems Engineering*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1994.
[18] Y. Cassuto, M. Schwartz, V. Bohossian, and J. Bruck, "Codes for multi-level flash memories: Correcting asymmetric limited-magnitude errors," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Nice, France, Jun. 2007, pp. 1176–1180.
[19] T. Fritzmann, T. Poppelmann, and J. Sepulveda, "Analysis of error-correcting codes for lattice-based key exchange," in *Proc. Int. Conf. Sel. Areas Cryptogr.*, in Lecture Notes in Computer Science. Berlin, Germany: Springer, 2018, pp. 369–390.
[20] J. P. D'Anvers, F. Vercauteren, and I. Verbauwhede, "On the impact of decryption failures on the security of LWE/LWR based schemes," IACR, Lyon, France, Tech. Rep. 2018/1089, 2018. Accessed: Dec. 13, 2019. [Online]. Available: https://eprint.iacr.org/2018/1089
[21] M. Walters and S. S. Roy, "Constant-time BCH error-correcting code," IACR, Lyon, France, Tech. Rep. 2019/155, 2019. Accessed: Dec. 13, 2019. [Online]. Available: https://eprint.iacr.org/2019/155
[22] H. Helgert and R. Stinaff, "Shortened BCH codes," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 6, pp. 818–820, Nov. 1973.
[23] R. E. Blahut, *Algebraic Codes for Data Transmission*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[24] J. P. D'Anvers, F. Vercauteren, and I. Verbauwhede, "The impact of error dependencies on Ring/Mod-LWE/LWR based schemes," in *Proc. 2019 Int. Conf. Post-Quantum Cryptogr.*, in Lecture Notes in Computer Science. Berlin, Germany: Springer, 2019, pp. 225–246.
[25] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle, "CRYSTALS—Kyber: A CCA-secure module-lattice-based KEM," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS P)*, Apr. 2018, pp. 353–367.
[26] P. Schwabe, R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, G. Seiler, and D. Stehle. *CRYSTALS-KYBER*. Accessed: Dec. 13, 2019. [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/Round-1-Submissions

EUNSANG LEE received the B.S. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2014, where he is currently pursuing the Ph.D. degree in electrical engineering and computer science. His current research interests include cryptography and error-correcting codes.

YOUNG-SIK KIM received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from Seoul National University, in 2001, 2003, and 2007, respectively. He joined the Semiconductor Division, Samsung Electronics, where he performed research and development of security hardware IPs for various embedded systems, including modular exponentiation hardware accelerator (*Tornado 2MX2*) for RSA and elliptic curve cryptography in smart card products and mobile application processors of Samsung Electronics, until 2010. He has been with Chosun University, Gwangju, South Korea, as a Professor. He is currently a Submitter for two candidate algorithms (McNie and pqsigRM) in the first round for the NIST Post Quantum Cryptography Standardization. His research interests include post-quantum cryptography, the IoT security, physical layer security, data hiding, channel coding, and signal design. He is selected as one of 2025's 100 Best Technology Leaders (for Crypto-Systems) by the National Academy of Engineering, South Korea.

JONG-SEON NO received the B.S. and M.S.E.E. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1981 and 1984, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1988. He was a Senior MTS with Hughes Network Systems, from 1988 to 1990. He was an Associate Professor with the Department of Electronic Engineering, Konkuk University, Seoul, from 1990 to 1999. He joined the Faculty of the Department of Electrical and Computer Engineering, Seoul National University, in 1999, where he is currently a Professor. His current research interests include error-correcting codes, sequences, cryptography, LDPC codes, interference alignment, and wireless communication systems. He became the IEEE Fellow through the IEEE Information Theory Society, in 2012. He was a recipient of the IEEE Information Theory Society Chapter of the Year Award, in 2007. From 1996 to 2008, he served as the Founding Chair for the Seoul Chapter of the IEEE Information Theory Society. He was the General Chair of *Sequence and Their Applications 2004*, Seoul, and the General Co-Chair for the International Symposium on Information Theory and Its Applications 2006 and the International Symposium on Information Theory 2009, Seoul.

MINKI SONG received the B.S. degree in electronics and computer engineering from Hanyang University, Seoul, South Korea, in 2013, and the M.S. degree in electronics and communications engineering from Hanyang University, Seoul, in 2015, where he is currently pursuing the Ph.D. degree in electronics and computer engineering. His research interests include signal processing, error-correcting codes, and cryptography.



DONG-JOON SHIN received the B.S. degree in electronics engineering from Seoul National University, Seoul, South Korea, the M.S. degree in electrical engineering from Northwestern University, Evanston, USA, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, USA. From 1999 to 2000, he was a Member of Technical Staff with the Wireless Network Division and Satellite Network Division, Hughes Network Systems, Maryland, USA. Since September 2000, he has been with the Department of Electronic Engineering, Hanyang University, Seoul, South Korea. His current research interests include signal processing, error-correcting codes, sequences, discrete mathematics, and cryptography.