In addition, examples with intermediate values have also been provided in these folders.

Notice that one can generate the aforementioned test files using respectively the `kat` and `verbose` modes of our implementation. The procedure to follow in order to do so is detailed in the technical documentation.

5 Security

In this section we prove the security of our encryption scheme viewed as a PKE scheme (IND-CPA). The security of the KEM/DEM version is provided by the transformation described in [23], and the tightness of the reduction provided by this transformation has been discussed at the end of Sec. 2.2.

Theorem 5.1. *The scheme presented above is* IND-CPA *under the assumption that both the* 2-*DQCSD with parity and* 3-*DQCSD with parity and erasures are hard.*

Proof of Theorem 5.1. To prove the security of the scheme, we are going to build a sequence of games transitioning from an adversary receiving an encryption of message $\mathbf{m}_0$ to an adversary receiving an encryption of a message $\mathbf{m}_1$, and show that if the adversary manages to distinguish one from the other, then we can build a simulator breaking the DQCSD assumption with parity and $\ell \geq 1$ erasure(s), for QC codes of index 2 or 3 (codes with parameters $[2n, n]$ or $[3n, n]$), and running in approximately the same time. As for the seed expansion, we assume that all the hash functions used can be modeled by random oracles.

Game G_1: This is the real game, which we can state algorithmically as follows:

```
$\mathbf{Game}^1_{\mathcal{E},\mathcal{A}}(\lambda)$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
2. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ with $\mathsf{pk} = (\mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y})$ and $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$
3. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}(\mathtt{FIND} : \mathsf{pk})$
4. $\mathbf{c}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathbf{m}_0) = (\mathbf{u}, \mathbf{v}) \in \mathbb{F}_2^n \times \mathbb{F}_2^{n_1 n_2}$
5. $b' \leftarrow \mathcal{A}(\mathtt{GUESS} : \mathbf{c}^*)$
6. RETURN $b'$
```

Game G_2: In this game we start by forgetting the decryption key $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$, and taking $\mathbf{s}$ at random of same bit parity $b = w + \mathbf{h}(1) \times w \mod 2$ as $\mathbf{s}' = \mathbf{x} + \mathbf{h} \cdot \mathbf{y}$, and then proceed honestly:

```
$\mathbf{Game}^2_{\mathcal{E},\mathcal{A}}(\lambda)$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
2a. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ with $\mathsf{pk} = (\mathbf{h}, \mathbf{s}' = \mathbf{x} + \mathbf{h} \cdot \mathbf{y})$ and $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$
2b. $\mathbf{s} \overset{\$}{\leftarrow} \mathbb{F}_{2,b}^n$, for $b = \mathbf{s}'(1) \mod 2$
2c. $(\mathsf{pk}, \mathsf{sk}) \leftarrow ((\mathbf{h}, \mathbf{s}), \mathbf{0})$
3. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}(\mathtt{FIND} : \mathsf{pk})$
```

```
4. $\mathbf{c}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathbf{m}_0) = (\mathbf{u}, \mathbf{v}) \in \mathbb{F}_2^n \times \mathbb{F}_2^{n_1 n_2}$
5. $b' \leftarrow \mathcal{A}(\texttt{GUESS} : \mathbf{c}^*)$
6. RETURN $b'$
```

The adversary has access to pk and $\mathbf{c}^*$. As he has access to pk and the $\mathsf{Encrypt}$ function, anything that is computed from pk and $\mathbf{c}^*$ can also be computed from just pk. Moreover, the distribution of $\mathbf{c}^*$ is independent of the game we are in. Indeed, assume that $\mathbf{m}_0$ and $\mathbf{m}_1$ have different bit parities. Without loss of generality, say even for $\mathbf{m}_0$ and odd for $\mathbf{m}_1$ and assume $\mathbf{h}$ has odd parity (a similar reasoning holds for $\mathbf{h}$ of even parity). As the parities of w, $w_{\mathbf{r}}$, and $w_{\mathbf{e}}$ are all known (see Tab. 5), the adversary knows the parity of $\mathbf{m}_b\mathbf{G} \in \mathbb{F}_2^n$, $\mathbf{s}\mathbf{r}_2 \in \mathbb{F}_2^n$, and $\mathbf{e} \in \mathbb{F}_2^n$. As the message is encrypted in $\mathbb{F}_2^{n_1 n_2}$, the last $\ell = n - n_1 n_2$ bits of the vector $\mathbf{v}$ are truncated, yielding a vector $\tilde{\mathbf{v}} \in \mathbb{F}_2^{n_1 n_2}$ of unknown parity. This is illustrated in Fig. 6. Therefore we can suppose the only input of the adversary is pk.

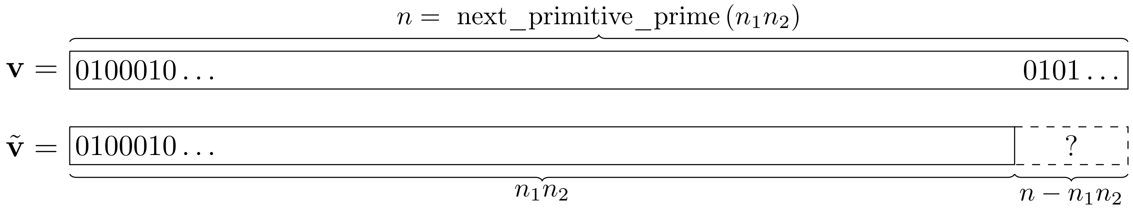

Figure 6: Truncation of vector $\mathbf{v}$ from $\mathbb{F}_2^n$ to $\tilde{\mathbf{v}} \in \mathbb{F}_2^{n_1 n_2}$.

Now suppose the adversary has an algorithm $\mathcal{D}_\lambda$, taking pk as input, that distinguishes with advantage ϵ Game $\boldsymbol{G}_1$ and Game $\boldsymbol{G}_2$, for some security parameter λ. Then he can also build an algorithm $\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}$ which solves the 2-$\mathsf{DQCSD}(n, w, b)$ problem with parity with the same advantage ϵ as the game distinguisher.

```
$\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}((\mathbf{H}, \mathbf{s}))$
1. Set $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
2. $\mathsf{pk} \leftarrow (\mathbf{h}, \mathbf{s})$
3. $b' \leftarrow \mathcal{D}_\lambda(\mathsf{pk})$
4. If $b' == 1$ output QCSD
5. If $b' == 2$ output UNIFORM
```

Note that if we define pk as $(\mathbf{h}, \mathbf{y})$ and $(\mathbf{H}, \mathbf{y}^\top)$ from a 2-$\mathsf{QCSD}(n, w, b)$ distribution with parity, pk follows exactly the same distribution as in Game $\boldsymbol{G}_1$. On the other hand if $(\mathbf{H}, \mathbf{y}^\top)$ comes from a uniform distribution over $\mathbb{F}_{2,b}^{n \times 2n} \times \mathbb{F}_{2,b'}^n$, pk follows exactly the same distribution as in Game $\boldsymbol{G}_2$.

Thus we have:

$$\begin{aligned}\Pr\left[\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}((\mathbf{H},\mathbf{y}^\top)) = \texttt{QCSD} | (\mathbf{H},\mathbf{y}^\top) \leftarrow \text{2-}\mathsf{QCSD}(n,w,b)\right] = \\ \Pr\left[\mathcal{D}_\lambda(\mathsf{pk}) = 1 | \mathsf{pk} \text{ from } \mathbf{Game}^0_{\mathcal{E},\mathcal{A}}(\lambda)\right], \text{ and}\end{aligned} \tag{21}$$

$$\begin{aligned}\Pr\left[\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}((\mathbf{H},\mathbf{y}^\top)) = \texttt{UNIFORM} | (\mathbf{H},\mathbf{y}^\top) \leftarrow \text{2-}\mathsf{QCSD}(n,w,b)\right] = \\ \Pr\left[\mathcal{D}_\lambda(\mathsf{pk}) = 2 | \mathsf{pk} \text{ from } \mathbf{Game}^0_{\mathcal{E},\mathcal{A}}(\lambda)\right]\end{aligned} \tag{22}$$

And similarly when $(\mathbf{H},\mathbf{y}^\top)$ is uniform the probabilities of $\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}$ outputs match those of $\mathcal{D}_\lambda$ when pk is from $\mathbf{Game}^2_{\mathcal{E},\mathcal{A}}(\lambda)$. The advantage of $\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}$ is therefore equal to the advantage of $\mathcal{D}_\lambda$.

Game $\boldsymbol{G}_3$: Now that we no longer know the decryption key, we can start generating random ciphertexts. So instead of picking correctly weighted $\mathbf{r}_1, \mathbf{r}_2, \mathbf{e}$, the simulator now picks random vectors in $\mathbb{F}^n_{2,w_\mathbf{r}}$ and $\mathbb{F}^n_{2,w_\mathbf{e}}$.

```
$\mathbf{Game}^3_{\mathcal{E},\mathcal{A}}(\lambda)$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
2a. $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ with $\mathsf{pk} = (\mathbf{h},\mathbf{s}' = \mathbf{x} + \mathbf{h}\cdot\mathbf{y})$ and $\mathsf{sk} = (\mathbf{x},\mathbf{y})$
2b. $\mathbf{s} \xleftarrow{\$} \mathbb{F}^n_{2,b}$, for $b = \mathbf{s}'(1) \mod 2$
2c. $(\mathsf{pk},\mathsf{sk}) \leftarrow ((\mathbf{h},\mathbf{s}),\mathbf{0})$
3. $(\mathbf{m}_0,\mathbf{m}_1) \leftarrow \mathcal{A}(\texttt{FIND} : \mathsf{pk})$
4a. $\mathbf{e} \xleftarrow{\$} \mathbb{F}^n_{2,w_\mathbf{e}}, \mathbf{r} = (\mathbf{r}_1,\mathbf{r}_2) \xleftarrow{\$} \mathbb{F}^n_{2,w_\mathbf{r}} \times \mathbb{F}^n_{2,w_\mathbf{r}}$
4b. $\mathbf{u} \leftarrow \mathbf{r}_1 + \mathbf{h}\mathbf{r}_2$ and $\mathbf{v} \leftarrow \mathbf{m}_0\mathbf{G} + \mathbf{s}\cdot\mathbf{r}_2 + \mathbf{e}$
4c. $\mathbf{c}^* \leftarrow (\mathbf{u},\mathbf{v})$, with $\mathbf{v}$ truncated in $\mathbb{F}_2^{n_1 n_2}$
5. $b' \leftarrow \mathcal{A}(\texttt{GUESS} : \mathbf{c}^*)$
6. RETURN $b'$
```

As we have

$$(\mathbf{u},\mathbf{v}-\mathbf{m}_0\mathbf{G})^\top = \begin{pmatrix}\mathbf{I}_n & \mathbf{0} & \mathbf{rot}(\mathbf{h}) \\ \mathbf{0} & \mathbf{I}_n & \mathbf{rot}(\mathbf{s})\end{pmatrix} \cdot (\mathbf{r}_1,\mathbf{e},\mathbf{r}_2)^\top,$$

the difference between Game $\boldsymbol{G}_2$ and Game $\boldsymbol{G}_3$ is that in the former

$$\left(\begin{pmatrix}\mathbf{I}_n & \mathbf{0} & \mathbf{rot}(\mathbf{h}) \\ \mathbf{0} & \mathbf{I}_n & \mathbf{rot}(\mathbf{s})\end{pmatrix}, (\mathbf{u},\mathbf{v}-\mathbf{m}_0\mathbf{G})^\top\right)$$

follows the 3-QCSD distribution with parity, and in the latter it follows a uniform distribution (as $\mathbf{r}_1$ and $\mathbf{e}$ are uniformly distributed over $\mathbb{F}^n_{2,b}$ with b odd) over $\mathbb{F}^{2n\times 3n}_{2,b_1,b_2} \times (\mathbb{F}^n_{2,b'_1} \times \mathbb{F}^n_{2,b'_2})$.

Note that an adversary is not able to obtain $\mathbf{c}^*$ from pk anymore, as depending on which game we are $\mathbf{c}^*$ is generated differently. The input of a game distinguisher will therefore be $(\mathsf{pk},\mathbf{c}^*)$. As it must interact with the challenger as usually we suppose it has two access modes FIND and GUESS to process first pk and later $\mathbf{c}^*$.

Suppose the adversary is able to distinguish Game $\boldsymbol{G}_2$ and Game $\boldsymbol{G}_3$, with a distinguisher $\mathcal{D}_\lambda$, which takes as input $(\mathsf{pk}, \mathbf{c}^*)$ and outputs a guess $b' \in \{2, 3\}$ of the game we are in.

Again, we can build a distinguisher $\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}$ that will break the 3-$\mathsf{DQCSD}(n, w, b_1, b_2)$ with parity and $\ell = n - n_1 n_2$ erasures assumption from $\mathsf{Setup}(1^\lambda)$ with the same advantage as the game distinguisher. In the 3-$\mathsf{DQCSD}(n, w, b_1, b_2)$ problem with parity, matrix $\mathbf{H}$ is assumed to be of the form

$$\begin{pmatrix} \mathbf{I}_n & \mathbf{0} & \mathbf{rot}(\mathbf{a}) \\ \mathbf{0} & \mathbf{I}_n & \mathbf{rot}(\mathbf{b}) \end{pmatrix}.$$

In order to use explicitly $\mathbf{a}$ and $\mathbf{b}$ we denote this matrix $\mathbf{H}_{\mathbf{a},\mathbf{b}}$ instead of just $\mathbf{H}$. We will also note $\mathbf{t} = (\mathbf{t}_1, \mathbf{t}_2)$.

```
$\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}\left(\left(\mathbf{H}_{\mathbf{a},\mathbf{b}}, (\mathbf{t}_1, \mathbf{t}_2)^\top\right)\right)$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
2a. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ with $\mathsf{pk} = (\mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y})$ and $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$
2b. $(\mathsf{pk}, \mathsf{sk}) \leftarrow ((\mathbf{a}, \mathbf{b}), \mathbf{0})$
3. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}(\mathtt{FIND} : \mathsf{pk})$
4. $\mathbf{u} \leftarrow \mathbf{t}_1$, $\mathbf{v} \leftarrow \mathbf{m}_0\mathbf{G} + \mathbf{t}_2$ and $\mathbf{c}^* \leftarrow (\mathbf{u}, \mathbf{v})$
5. $b' \leftarrow \mathcal{D}_\lambda(\mathtt{GUESS} : \mathbf{c}^*)$
4. If $b' == 2$ output QCSD
5. If $b' == 3$ output UNIFORM
```

The distribution of pk is unchanged with respect to the games. If $\left(\mathbf{H}_{\mathbf{a},\mathbf{b}}, (\mathbf{t}_1, \mathbf{t}_2)^\top\right)$ follows the 3-$\mathsf{QCSD}(n, w, b_1, b_2)$ distribution with parity, then

$$(\mathbf{t}_1, \mathbf{t}_2)^\top = \begin{pmatrix} \mathbf{I}_n & \mathbf{0} & \mathbf{rot}(\mathbf{a}) \\ \mathbf{0} & \mathbf{I}_n & \mathbf{rot}(\mathbf{b}) \end{pmatrix} \cdot (\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3)^\top$$

with $\omega(\mathbf{z}_1) = \omega(\mathbf{z}_2) = \omega(\mathbf{z}_3) = w$. Thus, $\mathbf{c}^*$ follows the same distribution as in Game $\boldsymbol{G}_2$. If $\left(\mathbf{H}_{\mathbf{a},\mathbf{b}}, (\mathbf{t}_1, \mathbf{t}_2)^\top\right)$ follows a uniform distribution with $\mathbf{a}$ of parity b_1 and $\mathbf{b}$ of parity b_2, then $\mathbf{c}^*$ follows the same distribution as in Game $\boldsymbol{G}_3$. We obtain therefore the same equalities for the output probabilities of $\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}$ and $\mathcal{D}_\lambda$ as with the previous games and therefore the advantages of both distinguishers are equal.

Game $\boldsymbol{G}_4$: We now encrypt the other plaintext. We chose $\mathbf{r}'_1, \mathbf{r}'_2, \mathbf{e}'$ uniformly at random in $\mathbb{F}^n_{2,w_\mathbf{r}}$ and $\mathbb{F}^n_{2,w_\mathbf{e}}$ and set $\mathbf{u} = \mathbf{r}'_1 + \mathbf{h}\mathbf{r}'_2$ and $\mathbf{v} = \mathbf{m}_1\mathbf{G} + \mathbf{s} \cdot \mathbf{r}'_2 + \mathbf{e}'$. This is the last game we describe explicitly since, even if it is a mirror of Game $\boldsymbol{G}_3$, it involves a new proof.

```
$\mathbf{Game}^4_{\mathcal{E},\mathcal{A}}(\lambda)$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
```

```
2a. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ with $\mathsf{pk} = (\mathbf{h}, \mathbf{s}' = \mathbf{x} + \mathbf{h} \cdot \mathbf{y})$ and $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$
2b. $\mathbf{s} \overset{\$}{\leftarrow} \mathbb{F}^n_{2,b}$, with $b = \mathbf{s}'(1) \mod 2$
2c. $(\mathsf{pk}, \mathsf{sk}) \leftarrow ((\mathbf{h}, \mathbf{s}), \mathbf{0})$
3. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}(\texttt{FIND} : \mathsf{pk})$
4a. $\mathbf{e}' \overset{\$}{\leftarrow} \mathbb{F}^n_{2,w_\mathbf{e}}, \mathbf{r}' = (\mathbf{r}'_1, \mathbf{r}'_2) \overset{\$}{\leftarrow} \mathbb{F}^n_{2,w_\mathbf{r}} \times \mathbb{F}^n_{2,w_\mathbf{r}}$
4b. $\mathbf{u} \leftarrow \mathbf{r}'_1 + \mathbf{h}\mathbf{r}'_2$ and $\mathbf{v} \leftarrow \mathbf{m}_1\mathbf{G} + \mathbf{s} \cdot \mathbf{r}'_2 + \mathbf{e}'$
4c. $\mathbf{c}^* \leftarrow (\mathbf{u}, \mathbf{v})$
5. $b' \leftarrow \mathcal{A}(\texttt{GUESS} : \mathbf{c}^*)$
6. RETURN $b'$
```

The outputs from Game $\boldsymbol{G}_3$ and Game $\boldsymbol{G}_4$ follow the exact same distribution, and therefore the two games are indistinguishable from an information-theoretic point of view. Indeed, for each tuple $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{e})$ of Game $\boldsymbol{G}_3$, resulting in a given $(\mathbf{u}, \mathbf{v})$, there is a one to one mapping to a couple $(\mathbf{r}'_1, \mathbf{r}'_2, \mathbf{e}')$ resulting in Game $\boldsymbol{G}_4$ in the *same* $(\mathbf{u}, \mathbf{v})$, namely $\mathbf{r}'_1 = \mathbf{r}_1$, $\mathbf{r}'_2 = \mathbf{r}_2$ and $\mathbf{e}' = \mathbf{m}_0\mathbf{G} + \mathbf{m}_1\mathbf{G}$. This implies that choosing uniformly $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{e})$ in Game $\boldsymbol{G}_3$ and choosing uniformly $(\mathbf{r}'_1, \mathbf{r}'_2, \mathbf{e}')$ in Game $\boldsymbol{G}_4$ leads to the same output distribution for $(\mathbf{u}, \mathbf{v})$.

Game $\boldsymbol{G}_5$: In this game, we now pick $\mathbf{r}'_1, \mathbf{r}'_2, \mathbf{e}'$ with the correct weight.

Game $\boldsymbol{G}_6$: We now conclude by switching the public key to an honestly generated one.

We do not explicit these last two games as Game $\boldsymbol{G}_4$ and Game $\boldsymbol{G}_5$ are the equivalents of Game $\boldsymbol{G}_3$ and Game $\boldsymbol{G}_2$ except that $\mathbf{m}_1$ is used instead of $\mathbf{m}_0$. A distinguisher between these two games breaks therefore the 3-DQCSD with parity and $\ell = n - n_1 n_2$ erasures assumption too. Similarly Game $\boldsymbol{G}_5$ and Game $\boldsymbol{G}_6$ are the equivalents of Game $\boldsymbol{G}_2$ and Game $\boldsymbol{G}_1$ and a distinguisher between these two games breaks the 2-DQCSD with parity assumption.

We managed to build a sequence of games allowing a simulator to transform a ciphertext of a message $\mathbf{m}_0$ to a ciphertext of a message $\mathbf{m}_1$. Hence, the advantage of an adversary against the IND-CPA experiment is bounded as:

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda) \leq 2\left(\mathsf{Adv}^{\text{2-DQCSD}}(\lambda) + \mathsf{Adv}^{\text{3-DQCSD}}(\lambda)\right). \tag{23}$$

□

6 Known Attacks

The practical complexity of the SD problem for the Hamming metric has been widely studied for more than 50 years. Most efficient attacks are based on Information Set Decoding, a technique first introduced by Prange in 1962 [36] and improved later by Stern [40], then Dumer [13]. Recent works [32, 4, 33] suggest a complexity of order $2^{cw(1+\mathsf{negl}(1))}$, for some constant c. A particular work focusing on the regime $w = \mathsf{negl}(n)$ confirms this formula, with a close dependence between c and the rate k/n of the code being used [11].

Specific structural attacks. Quasi-cyclic codes have a special structure which may potentially open the door to specific structural attacks. A first generic attack is the DOOM attack [38] which because of cyclicity implies a gain of $\mathcal{O}(\sqrt{n})$ (when the gain is in $\mathcal{O}(n)$ for MDPC codes, since the code is generated by a small weight vector basis). It is also possible to consider attacks on the form of the polynomial generating the cyclic structure. Such attacks have been studied in [22, 30, 38], and are especially efficient when the polynomial $x^n - 1$ has many low degree factors. These attacks become inefficient as soon as $x^n - 1$ has only two irreducible factors of the form $(x - 1)$ and $x^{n-1} + x^{n-2} + ... + x + 1$, which is the case when n is prime and q generates the multiplicative group $(\mathbb{Z}/n\mathbb{Z})^*$. Such numbers are known up to very large values. We consider such primitive n for our parameters.

Parameters and tightness of the reduction. We proposed different sets of parameters in Sec. 2.7 that provide 128 (category 1), 192 (category 3), and 256 (category 5) bits of classical (*i.e.* pre-quantum) security. The quantum-safe security is obtained by dividing the security bits by two (taking the square root of the complexity) [7]. Best known attacks include the works from [10, 8, 14, 32, 4, 33] and for quantum attacks, the work of [7]. In the setting $w = \mathcal{O}\left(\sqrt{n}\right)$, best known attacks have a complexity in $2^{-t \ln(1-R)(1+o(1))}$ where $t = \mathcal{O}(w)$ and R is the rate of the code [11]. In our configuration, we have $t = 2w$ and $R = 1/2$ for the reduction to the 2-DQCSD problem, and $t = 3w_{\mathbf{r}}$ and $R = 1/3$ for the 3-DQCSD problem. By taking into account the DOOM attack [38], and also the fact that we consider balanced vectors $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{r}_1, \mathbf{e}, \mathbf{r}_2)$ for the attack (which costs only a very small factor, since random words have a good probability to be balanced on each block), we need to divide this complexity by approximately $\sqrt{n}$ (up to polylog factor). The term $o(1)$ is respectively $\log\left(\binom{n}{w}^2 / \binom{2n}{2w}\right)$ and $\log\left(\binom{n}{w_{\mathbf{r}}}^3 / \binom{3n}{3w_{\mathbf{r}}}\right)$ for the 2-DQCSD and 3-DQCSD problems. Overall our security reduction is tight corresponding to generic instances of the classical 2-DQCSD and 3-DQCSD problems according to the best attacks of [11].

7 Advantages and Limitations

7.1 Advantages

The main advantages of HQC over existing code-based cryptosystems are:

- its IND-CPA reduction to a well-understood problem on coding theory: the Quasi-Cyclic Syndrome Decoding problem,
- its immunity against attacks aiming at recovering the hidden structure of the code being used,
- small public key size