A Comparison of NTRU Variants

John M. Schanck

Department of Combinatorics & Optimization and Institute for Quantum Computing
University of Waterloo, Waterloo, ON, N2L 3G1, Canada
jschanck@uwaterloo.ca [*]

Abstract. We analyze the size vs. security trade-offs that are available when selecting parameters for perfectly correct key encapsulation mechanisms based on NTRU.

1 Introduction.

Three of the submissions to the NIST post-quantum standardization effort are key encapsulation mechanisms based on NTRU. A number of objective measures might be used to compare them, e.g.:

- **Size** – The number of bytes in a public key and/or ciphertext.
- **Security** – The cost of a particular attack.
- **Efficiency** – The cost of using a scheme on a particular CPU.

Each of the NTRU submissions — NTRUEncrypt, NTRU-HRSS-KEM, and NTRU Prime — recommend a small number of concrete parameter sets. One might plot these parameter sets on a size vs. security vs. efficiency graph to see the impact of the decisions each team has made in the design of their KEM. Unfortunately, there would be some difficulties in interpreting such a graph. For example, a direct comparison of Streamlined NTRU Prime and NTRU-HRSS-KEM on a size vs. security vs. efficiency graph would mask a fourth variable:

- **Tightness of the security reduction**.

Worse yet, a comparison of either Streamlined NTRU Prime or NTRU-HRSS-KEM with NTRUEncrypt would mask a fifth variable:

- **Correctness**.

While it is difficult to compare the precise KEMs that were submitted to NIST, there are KEMs in the span of the three submissions that are easy to compare. Here we consider variants of NTRU-HRSS-KEM and NTRUEncrypt that have correctness and tightness properties identical to those of Streamlined NTRU Prime. We plot parameter sets for these variants, and parameter sets for Streamlined NTRU Prime, on size vs. security graphs to see the impact of several design decisions. In particular, our graphs make clear:

1. the impact of using fixed-weight vector sampling routines instead of uniform vector sampling routines;
2. the impact of using a prime modulus instead of a power of two modulus; and
3. the impact of using the ring $\mathbb{Z}[\mathbf{x}]/(\mathbf{x}^n - \mathbf{x} - 1)$ instead of the ring $\mathbb{Z}[\mathbf{x}]/(\mathbf{x}^n - 1)$.

Among the cryptosystems we consider is the variant of NTRU-HRSS-KEM proposed by Saito, Xagawa, and Yamakawa in [10]. We leave the full size vs. security vs. efficiency analysis to future work.

Availability of software and data. The source code used to produce the data used in this paper can be found at https://github.com/ntru-hrss/parameters.

[*] Date: November 30, 2018

Acknowledgements. Thanks to Keito Xagawa for discussions related to Section 5.1. Thanks to Daniel J. Bernstein for providing the motivation to create size vs. security graphs (in [2]).

Outline of this document. General background on NTRU is provided in Section 2. Public key encryption schemes based on NTRU-HRSS are described in Section 3. Public key encryption schemes based on NTRU-HPS are described in Section 4. IND-CCA2 KEMs are described in Section 5. Our size vs. security graphs are described in Section 6. Figures 9 and 10 are size vs. security graphs for the NTRUEncrypt and NTRU-HRSS-KEM variants. The same data is plotted against Streamlined NTRU Prime parameter sets in Figures 11 and 12. We conclude with some remarks on specific parameter sets in Section 7.

2 Preliminaries.

Notation. We denote the n-th cyclotomic polynomial by $\mathbf{\Phi}_n$. Note that $\mathbf{\Phi}_1 = \mathbf{x} - 1$ and if n is prime then $\mathbf{\Phi}_n = \mathbf{x}^{n-1} + \mathbf{x}^{n-2} + \cdots + 1$ and $\mathbf{\Phi}_1\mathbf{\Phi}_n = \mathbf{x}^n - 1$. We write $\mathbf{z} \bmod (\ell, \mathbf{w})$ for the reduced representative of the equivalence class $\mathbf{z} + (\ell, \mathbf{w})$. This is the unique element of $\mathbf{z} + (\ell, \mathbf{w})$ of degree $< \deg \mathbf{w}$ with coefficients between $\lfloor \frac{-\ell+1}{2} \rfloor$ and $\lfloor \frac{\ell-1}{2} \rfloor$. We denote the reduce representative of the multiplicative inverse of $\mathbf{z}$ in $\mathbb{Z}[\mathbf{x}]/(\ell, \mathbf{w})$, when it exists, by $(1/\mathbf{z}) \bmod (\ell, \mathbf{w})$. For a finite set $\mathcal{S}$ and a domain separation string $\texttt{dom}$ we define a random oracle $x \mapsto \mathsf{Sample}\mathcal{S}(x, \texttt{dom})$. In algorithms we use "$\leftarrow$" to denote assignment and "$\leftarrow_\$$" to denote sampling from the uniform distribution.

Parameters for NTRU-HPS. We refer to the original NTRU scheme, as defined by Hoffstein, Pipher, and Silverman [7], as NTRU-HPS. The scheme is parameterized by three coprime integers (n, p, q) and four sample spaces $(\mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_r, \mathcal{L}_m)$. It makes use of a ring $\mathcal{R} = (\mathbb{Z}^n, +, \circledast)$. Elements of this ring are written as polynomials in $\mathbf{x}$, e.g. $\mathbf{a} = a_0 + a_1\mathbf{x} + \cdots + a_{n-1}\mathbf{x}^{n-1}$. The $\circledast$ operation is cyclic convolution:

$$\mathbf{u} \circledast \mathbf{v} = \left(\sum_{i=0}^{n-1} u_i \mathbf{x}^i\right) \circledast \left(\sum_{j=0}^{n-1} v_j \mathbf{x}^j\right) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} u_i v_j \mathbf{x}^{(i+j) \bmod n}.$$

Bold face letters that appear in our description of NTRU will be treated either as elements of $\mathbb{Z}[\mathbf{x}]$ or as elements of $\mathcal{R}$, as convenient. It is not hard to see that $\mathbf{u} \circledast \mathbf{v} = \mathbf{v} \circledast \mathbf{u} = \mathbf{u}\mathbf{v} \bmod (\mathbf{\Phi}_1\mathbf{\Phi}_n)$. In fact, $\mathcal{R} \cong \mathbb{Z}[\mathbf{x}]/(\mathbf{\Phi}_1\mathbf{\Phi}_n)$. The four sample spaces are subsets of $\mathcal{R}$.

The NTRU-HPS *one-way function.* An NTRU-HPS private key is a pair $(\mathbf{f}, \mathbf{f}_p)$ with $\mathbf{f} \in \mathcal{L}_f$ and

$$\mathbf{f}_p \circledast \mathbf{f} \equiv 1 \pmod{p}. \tag{1}$$

A corresponding public key is $\mathbf{h} \in \mathcal{R}$ for which there exists $\mathbf{g} \in \mathcal{L}_g$ with

$$\mathbf{h} \circledast \mathbf{f} \equiv p\mathbf{g} \pmod{q}. \tag{2}$$

The NTRU-HPS one-way function is described in Figure 1.

```
$\mathcal{E}_{\text{HPS}}(\mathbf{h}, \mathbf{r}, \mathbf{m})$
1. $\mathbf{c} \leftarrow (\mathbf{r}\mathbf{h} + \mathbf{m}) \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
2. return $\mathbf{c}$
```

```
$\mathcal{D}_{\text{HPS}}(\mathbf{f}, \mathbf{f}_p, \mathbf{c})$
1. $\mathbf{a} \leftarrow \mathbf{c}\mathbf{f} \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
2. $\mathbf{m}' \leftarrow \mathbf{a}\mathbf{f}_p \bmod (p, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
3. return $\mathbf{m}'$
```

Fig. 1: The NTRU-HPS one-way function.

If $\mathbf{f}$, $\mathbf{f}_p$, and $\mathbf{h}$ are such that Equations (1) and (2) are satisfied and $\mathbf{c} = \mathcal{E}_{\text{HPS}}(\mathbf{h}, \mathbf{r}, \mathbf{m})$, then in Line 1 of $\mathcal{D}_{\text{HPS}}(\mathbf{f}, \mathbf{f}_p, \mathbf{c})$ we have $\mathbf{a} \equiv p\mathbf{r} \circledast \mathbf{g} + \mathbf{m} \circledast \mathbf{f} \pmod{q}$. If

$$|p\mathbf{r} \circledast \mathbf{g} + \mathbf{m} \circledast \mathbf{f}|_\infty < q/2. \tag{3}$$

then this equivalence modulo q can be promoted to an equality in $\mathbb{Z}[\mathbf{x}]$, and it follows that $\mathcal{D}_{\text{HPS}}(\mathbf{f}, \mathbf{f}_p, \mathbf{c}) = \mathbf{m}$.

Parameters for NTRU-HRSS*.* We will refer to the variant of NTRU that was proposed by Hülsing, Rijnveld, Schanck, and Schwabe [9] as NTRU-HRSS. The parameters of NTRU-HRSS differ from the parameters of NTRU-HPS as follows: The parameters n, p, and q are chosen so that $\mathbf{\Phi}_n$ is irreducible modulo p and modulo q; the polynomials in $\mathcal{L}_f$, $\mathcal{L}_g$, $\mathcal{L}_r$, and $\mathcal{L}_m$ are chosen to have degree at most $n-2$; and the elements of $\mathcal{L}_f$ and $\mathcal{L}_g$ are required to satisfy a "non-negative correlation" property, which we will describe momentarily.

The NTRU-HRSS *one-way function.* An NTRU-HRSS private key is $(\mathbf{f}, \mathbf{f}_p)$ with $\mathbf{f} \in \mathcal{L}_f$ and

$$\mathbf{f}_p\mathbf{f} \equiv 1 \pmod{(p, \mathbf{\Phi}_n)}. \tag{4}$$

A corresponding public key is a polynomial $\mathbf{h} \in \mathcal{R}$ for which there exists $\mathbf{g} \in \mathcal{L}_g$ with

$$\mathbf{h} \circledast \mathbf{f} \equiv p\mathbf{\Phi}_1 \circledast \mathbf{g} \pmod{q}. \tag{5}$$

We also define a function $\mathsf{Lift} : \mathcal{R} \to \mathcal{R}$ by

$$\mathsf{Lift}(\mathbf{m}) = \mathbf{\Phi}_1 \left((\mathbf{m}/\mathbf{\Phi}_1) \bmod (p, \mathbf{\Phi}_n) \right). \tag{6}$$

Note that $\mathsf{Lift}(\mathbf{m}) \equiv 0 \pmod{\mathbf{\Phi}_1}$ and that $\mathsf{Lift}(\mathbf{m}) \equiv \mathbf{m} \pmod{(p, \mathbf{\Phi}_n)}$. The NTRU-HRSS one-way function is described in Figure 2.

```
$\mathcal{E}_{\text{HRSS}}(\mathbf{h}, \mathbf{r}, \mathbf{m})$
1. $\mathbf{c} \leftarrow (\mathbf{rh} + \mathsf{Lift}(\mathbf{m})) \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
2. return $\mathbf{c}$

$\mathcal{D}_{\text{HRSS}}(\mathbf{f}, \mathbf{f}_p, \mathbf{c})$
1. $\mathbf{a} \leftarrow \mathbf{cf} \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
2. $\mathbf{m}' \leftarrow \mathbf{af}_p \bmod (p, \mathbf{\Phi}_n)$
3. return $\mathbf{m}'$
```

Fig. 2: The NTRU-HRSS one-way function.

If $\mathbf{f}$, $\mathbf{f}_p$, and $\mathbf{h}$ are such that Equations 4 and 5 are satisfied and $\mathbf{c} = \mathcal{E}_{\text{HRSS}}(\mathbf{h}, \mathbf{r}, \mathbf{m})$, then in Line 1 of $\mathcal{D}_{\text{HRSS}}(\mathbf{f}, \mathbf{f}_p, \mathbf{c})$ we have $\mathbf{a} \equiv p\mathbf{\Phi}_1 \circledast \mathbf{r} \circledast \mathbf{g} + \mathbf{\Phi}_1 \circledast \mathbf{s} \circledast \mathbf{f} \pmod{q}$ where $\mathbf{s} = (\mathbf{m}/\mathbf{\Phi}_1) \bmod (p, \mathbf{\Phi}_n)$. If

$$|p\mathbf{\Phi}_1 \circledast \mathbf{r} \circledast \mathbf{g} + \mathbf{\Phi}_1 \circledast \mathbf{s} \circledast \mathbf{f}|_\infty < q/2. \tag{7}$$

then this equivalence modulo q can be promoted to an equality in $\mathbb{Z}[\mathbf{x}]$, and it follows that $\mathcal{D}_{\text{HRSS}}(\mathbf{f}, \mathbf{f}_p, \mathbf{c}) \equiv \mathsf{Lift}(\mathbf{m}) \bmod (p, \mathbf{\Phi}_n)$.

The $\mathbf{x} \mapsto 1$ *homomorphism.* The "evaluate at 1" map, $\mathbf{x} \mapsto 1$, is a ring homomorphism from $\mathbb{Z}[\mathbf{x}]$ to $\mathbb{Z}$ with kernel $(\mathbf{\Phi}_1)$. Since $\mathcal{R} \cong \mathbb{Z}[\mathbf{x}]/(\mathbf{\Phi}_1) \times \mathbb{Z}[\mathbf{x}]/(\mathbf{\Phi}_n)$, the map that we obtain by treating an element of $\mathcal{R}$ as a polynomial in $\mathbf{x}$ and evaluating it at 1 is a ring homomorphism from $\mathcal{R}$ to $\mathbb{Z}$. This has some unfortunate consequences for NTRU-HPS. Public keys reveal potentially useful information about private keys: $\mathbf{h}(1) = p\mathbf{g}(1)/\mathbf{f}(1) \bmod q$. Ciphertexts reveal potentially useful information about messages: $\mathbf{c}(1) = \mathbf{r}(1)\mathbf{h}(1) + \mathbf{m}(1) \bmod q$. The amount of information revealed depends on the choice of the sample spaces $\mathcal{L}_f$, $\mathcal{L}_g$, $\mathcal{L}_r$, and $\mathcal{L}_m$. The main difference between NTRU-HPS and NTRU-HRSS is in how the $\mathbf{x} \mapsto 1$ homomorphism is treated. The extra multiplications by $\mathbf{\Phi}_1$ in the definition of NTRU-HRSS are there to ensure that $\mathbf{h}(1) = 0$ and $\mathbf{c}(1) = 0$ regardless of the choice of sample spaces.

Geometry and $\circledast$*-multiplication.* We will need some basic geometric facts in our discussion of perfect correctness. Let $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{R}$. The inner product of $\mathbf{u}$ and $\mathbf{v}$ is $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=0}^{n-1} u_i v_i$. The conjugate of $\mathbf{u}$ is $\overline{\mathbf{u}} = u_0 + \sum_{i=1}^{n-1} u_{n-i}\mathbf{x}^i$. Note that $\overline{\mathbf{u} + \mathbf{v}} = \overline{\mathbf{u}} + \overline{\mathbf{v}}$ and $\overline{\mathbf{u} \circledast \mathbf{v}} = \overline{\mathbf{u}} \circledast \overline{\mathbf{v}}$. Moreover, for $0 \leq k < n$, $\langle \mathbf{x}^k, \mathbf{u} \rangle = u_k = \langle 1, \overline{\mathbf{x}}^k \circledast \mathbf{u} \rangle$ and by bilinearity of the inner product $\langle \mathbf{u} \circledast \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{u}, \overline{\mathbf{v}} \circledast \mathbf{w} \rangle$. The 2-norm of $\mathbf{u}$ is $|\mathbf{u}|_2 = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$. The max-norm of $\mathbf{u}$ is $|\mathbf{u}|_\infty = \max_i |\langle \mathbf{x}^i, \mathbf{u} \rangle|$. The max-norm of a $\circledast$-product satisfies

$$|\mathbf{u} \circledast \mathbf{v}|_\infty \leq |\mathbf{u}|_2 |\mathbf{v}|_2. \tag{8}$$

To see this note that $|\mathbf{u} \circledast \mathbf{v}|_\infty = \max_i |\langle \mathbf{x}^i \circledast \overline{\mathbf{u}}, \mathbf{v}\rangle|$, and $|\langle \mathbf{x}^i \circledast \overline{\mathbf{u}}, \mathbf{v}\rangle| \leq |\mathbf{x}^i \circledast \overline{\mathbf{u}}|_2 |\mathbf{v}|_2$ by Cauchy–Schwarz. Equation (8) follows because both $\circledast$-multiplication with $\mathbf{x}$ and conjugation are 2-norm preserving. Equality holds when $\mathbf{u} = \mathbf{x}^i \circledast \overline{\mathbf{v}}$ for some i.

For NTRU-HRSS we also need to bound the max-norm of three term products of the form $\mathbf{\Phi}_1 \circledast \mathbf{u} \circledast \mathbf{v}$. The proof of [9, Lemma 1] shows that if $\langle \mathbf{x} \circledast \mathbf{v}, \mathbf{v}\rangle \geq 0$ then

$$|\mathbf{\Phi}_1 \circledast \mathbf{u} \circledast \mathbf{v}|_\infty \leq \sqrt{2} |\mathbf{u}|_2 |\mathbf{v}|_2. \tag{9}$$

When $\mathbf{v}$ is such that $\langle \mathbf{x} \circledast \mathbf{v}, \mathbf{v}\rangle \geq 0$ we say that $\mathbf{v}$ has the **non-negative correlation** property.

Sample spaces. Typical NTRU-HPS instantiations take the four sample spaces to be sets of polynomials of degree $\leq n-1$ with coefficients in between $\lfloor \frac{-p+1}{2} \rfloor$ and $\lfloor \frac{p-1}{2} \rfloor$, i.e. sets of reduced representatives of $\mathbb{Z}[\mathbf{x}]/(p, \mathbf{\Phi}_1\mathbf{\Phi}_n)$. The instantiations we consider below use polynomials of degree $\leq n-2$, i.e. sets of reduced representatives of $\mathbb{Z}[\mathbf{x}]/(p, \mathbf{\Phi}_n)$. We define

$$\mathcal{T} = \{\mathbf{z} \in \mathcal{R} \; : \; \mathbf{z} = \mathbf{z} \bmod (p, \mathbf{\Phi}_n)\}.$$

The subset of $\mathcal{T}$ with non-negative correlation, needed for NTRU-HRSS, is

$$\mathcal{T}_+ = \{\mathbf{z} \in \mathcal{T} \; : \; \langle \mathbf{x} \circledast \mathbf{z}, \mathbf{z}\rangle \geq 0\}.$$

The subset of $\mathcal{T}$ with coefficients that sum to zero, needed for our NTRU-HPS instantiation, is

$$\mathcal{T}_0 = \{\mathbf{z} \in \mathcal{T} \; : \; \mathbf{z}(1) = 0\}.$$

We will also consider sets of vectors of fixed 2-norm:

$$\mathcal{T}(d) = \{\mathbf{z} \in \mathcal{T} \; : \; |\mathbf{z}|_2^2 = d\}, \quad \mathcal{T}_+(d) = \mathcal{T}(d) \cap \mathcal{T}_+, \quad \text{and} \quad \mathcal{T}_0(d) = \mathcal{T}(d) \cap \mathcal{T}_0.$$

When $p = 3$ we will refer to $\mathcal{T}(d)$ and $\mathcal{T}_+(d)$ as sets of **fixed-weight** vectors. The fixed-weight sets contain vectors of hamming weight d. When $p = 3$ we will refer to $\mathcal{T}_0(d)$ as a set of **fixed-type** vectors. We will assume that d is even in this case. The fixed-type sets contain vectors that have exactly $d/2$ coefficients equal to $+1$ and $d/2$ coefficients equal to -1.

Clean parameters. We say that a parameter set $(n, p, q, \mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_r, \mathcal{L}_m)$ is **clean** if 1) p is prime, 2) $\mathbf{\Phi}_n$ is irreducible modulo p and modulo q, 3) $\mathcal{L}_f$, $\mathcal{L}_g$, $\mathcal{L}_r$, and $\mathcal{L}_m$ are all subsets of $\mathcal{T}$, and 4) all $d > 1$ that divide q satisfy $d > \lfloor p/2 \rfloor$. The fourth condition ensures that the content (g.c.d. of coefficients) of an element of $\mathcal{T}$ is coprime with q, and allows us to eliminate some invertibility tests.

3 OW-CPA public key encryption schemes from NTRU-HRSS.

NTRU-HRSS was originally presented as a probabilistic public key encryption (PPKE) scheme [9]. This scheme is reproduced in Figure 3. The deterministic public key encryption (DPKE) scheme in Figure 4 was presented by Saito, Xagawa, and Yamakawa in [10]. Here we consider more general parameters than have been considered in previous work and we state the general versions of the correctness theorems.

Remark 1. The polynomial $p\mathbf{\Phi}_1\mathbf{g}\mathbf{f}_q$ has coefficients that sum to zero, so $\mathbf{h}$ has coefficients that sum to a multiple of q. A small space-saving optimization is to drop the coefficient h_{n-1} before transmission; the recipient can recover it as $h_{n-1} = -\sum_{i=0}^{n-2} h_i \bmod q$. The same can be done for $\mathbf{c}$. We assume that this optimization is used when we calculate the size of NTRU-HRSS public keys and ciphertexts.

```
$\mathsf{KeyGen}_1(seed)$
1. $\mathbf{g} \leftarrow \mathsf{Sample}\mathcal{L}_g(seed, \mathtt{domg})$
2. $\mathbf{f} \leftarrow \mathsf{Sample}\mathcal{L}_f(seed, \mathtt{domf})$
3. $\mathbf{f}_q \leftarrow (1/\mathbf{f}) \bmod (q, \mathbf{\Phi}_n)$
4. $\mathbf{f}_p \leftarrow (1/\mathbf{f}) \bmod (p, \mathbf{\Phi}_n)$
5. $\mathbf{h} \leftarrow (p\mathbf{\Phi}_1\mathbf{g}\mathbf{f}_q) \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
6. return $dk = (\mathbf{f}, \mathbf{f}_p),\ ek = \mathbf{h}$
```

```
$\mathsf{Enc}_1(\mathbf{h}, coins, \mathbf{m})$ with $\mathbf{m} \in \mathcal{L}_m$
1. $\mathbf{r} \leftarrow \mathsf{Sample}\mathcal{L}_r(coins, \mathtt{domr})$
2. $\mathbf{c} \leftarrow (\mathbf{r}\mathbf{h} + \mathsf{Lift}(\mathbf{m})) \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
3. return $\mathbf{c}$
```

```
$\mathsf{Dec}_1((\mathbf{f}, \mathbf{f}_p), \mathbf{c})$
1. $\mathbf{a} \leftarrow (\mathbf{c}\mathbf{f}) \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
2. $\mathbf{m}' \leftarrow (\mathbf{a}\mathbf{f}_p) \bmod (p, \mathbf{\Phi}_n)$
3. if $\mathbf{m}' \notin \mathcal{L}_m$ return $\perp$
4. else return $\mathbf{m}'$
```

Fig. 3: A probabilistic public key encryption scheme using the NTRU-HRSS one-way function.

```
$\mathsf{KeyGen}_2(seed)$
1. $\mathbf{g} \leftarrow \mathsf{Sample}\mathcal{L}_g(seed, \mathtt{domg})$
2. $\mathbf{f} \leftarrow \mathsf{Sample}\mathcal{L}_f(seed, \mathtt{domf})$
3. $\mathbf{f}_q \leftarrow (1/\mathbf{f}) \bmod (q, \mathbf{\Phi}_n)$
4. $\mathbf{f}_p \leftarrow (1/\mathbf{f}) \bmod (p, \mathbf{\Phi}_n)$
5. $\mathbf{h} \leftarrow (p\mathbf{\Phi}_1\mathbf{g}\mathbf{f}_q) \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
6. $\mathbf{h}_q \leftarrow (1/\mathbf{h}) \bmod (q, \mathbf{\Phi}_n)$
7. return $dk = (\mathbf{f}, \mathbf{f}_p, \mathbf{h}_q),\ ek = \mathbf{h}$
```

```
$\mathsf{Enc}_2(\mathbf{h}, (\mathbf{r}, \mathbf{m}))$ with $\mathbf{r} \in \mathcal{L}_r, \mathbf{m} \in \mathcal{L}_m$
1. $\mathbf{c} \leftarrow (\mathbf{r}\mathbf{h} + \mathsf{Lift}(\mathbf{m})) \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
2. return $\mathbf{c}$
```

```
$\mathsf{Dec}_2((\mathbf{f}, \mathbf{f}_p, \mathbf{h}_q), \mathbf{c})$
1. $\mathbf{a} \leftarrow (\mathbf{c}\mathbf{f}) \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
2. $\mathbf{m}' \leftarrow (\mathbf{a}\mathbf{f}_p) \bmod (p, \mathbf{\Phi}_n)$
3. $\mathbf{b} \leftarrow (\mathbf{c} - \mathsf{Lift}(\mathbf{m}')) \bmod (q, \mathbf{\Phi}_n)$
4. $\mathbf{r}' \leftarrow (\mathbf{b}\mathbf{h}_q) \bmod (q, \mathbf{\Phi}_n)$
5. if $(\mathbf{r}', \mathbf{m}') \notin \mathcal{L}_r \times \mathcal{L}_m$ return $\perp$
6. else return $(\mathbf{r}', \mathbf{m}')$
```

Fig. 4: A deterministic public key encryption scheme using the NTRU-HRSS one-way function.

3.1 Correctness for the NTRU-HRSS PPKE and DPKE schemes.

Line 2 of $\mathsf{Enc}_1(\mathbf{h}, coins, \mathbf{m})$ computes $\mathcal{E}_{\text{HRSS}}(\mathbf{h}, \mathbf{r}, \mathbf{m})$, for some $\mathbf{r}$ determined by $coins$, and Lines 1-2 of $\mathsf{Dec}_1((\mathbf{f}, \mathbf{f}_p), \mathbf{c})$ compute $\mathcal{D}_{\text{HRSS}}(\mathbf{f}, \mathbf{f}_p, \mathbf{c})$. A proof of correctness largely follows the reasoning around Equation (7). Parameters must be chosen such that the following three conditions are satisfied:

1. Equations (4) and (5) hold for all $((\mathbf{f}, \mathbf{f}_p), \mathbf{h})$ output by KeyGen_1;
2. The inequality (7) holds for all $\mathbf{f} \in \mathcal{L}_f$, $\mathbf{g} \in \mathcal{L}_g$, $\mathbf{r} \in \mathcal{L}_r$, and $\mathbf{s} \in \mathcal{T}$; and
3. For all $\mathbf{m} \in \mathcal{L}_m$ it is the case that $\mathbf{m} = \mathsf{Lift}(\mathbf{m}) \bmod (p, \mathbf{\Phi}_n)$.

The inverses in Lines 3 and 4 of KeyGen_1 exist when the parameters are clean. The first condition is satisfied when these inverses exist. The second and third conditions imply that $\mathcal{D}_{\text{HRSS}}(\mathbf{f}, \mathbf{f}_p, \mathcal{E}_{\text{HRSS}}(\mathbf{h}, \mathbf{r}, \mathbf{m})) = \mathbf{m}$ for all key pairs output by KeyGen_1, all $\mathbf{r} \in \mathcal{L}_r$, and all $\mathbf{m} \in \mathcal{L}_m$. The second condition must be enforced through the choice of q and the sample spaces; we will provide examples in the following section. The third condition holds so long as $\mathcal{L}_m \subseteq \mathcal{T}$, hence it holds for clean parameters. With these conditions the proof of Theorem 1 is a routine calculation.

Theorem 1. *Suppose that $(n, p, q, \mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_r, \mathcal{L}_m)$ is a clean parameter set. Further suppose that (7) is satisfied for all $\mathbf{f} \in \mathcal{L}_f$, $\mathbf{g} \in \mathcal{L}_g$, $\mathbf{r} \in \mathcal{L}_r$, and $\mathbf{s} \in \mathcal{T}$. Then for all $(dk, ek) = ((\mathbf{f}, \mathbf{f}_p), \mathbf{h})$ output by* KeyGen_1*, all $coins \in \{0,1\}^*$, and all $\mathbf{m} \in \mathcal{L}_m$ We have*

$$\mathbf{m} = \mathsf{Dec}_1(dk, \mathsf{Enc}_1(ek, coins, \mathbf{m})),$$

i.e. $(\mathsf{KeyGen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ *is a correct probabilistic public key encryption scheme.*

Theorem 2. *With the conditions of Theorem 1* $(\mathsf{KeyGen}_2, \mathsf{Enc}_2, \mathsf{Dec}_2)$ *is a correct deterministic public key encryption scheme.*

Proof. For the purpose of correctness, Enc_1 and Enc_2 differ only in whether $\mathbf{r}$ is provided as an input. The first two lines of Dec_2 are identical to the first two lines of Dec_1. From these observations and Theorem 1 we have $\mathbf{m}' = \mathbf{m}$ in Line 2 of $\mathsf{Dec}_2(dk, \mathsf{Enc}_2(\mathbf{h}, (\mathbf{r}, \mathbf{m}))$. Observe that $\mathsf{Enc}_2(\mathbf{h}, (\mathbf{r}, \mathbf{m})) \equiv \mathbf{r}\mathbf{h} + \mathsf{Lift}(\mathbf{m}) \pmod{(q, \mathbf{\Phi}_n)}$, and the inverse in Line 6 of KeyGen_2 exists when the parameters are clean. It follows that the value $\mathbf{r}'$ computed in Lines $3 - 4$ of Dec_2 satisfies $\mathbf{r}' \equiv (\mathbf{r}\mathbf{h})\mathbf{h}_q \equiv \mathbf{r} \pmod{(q, \mathbf{\Phi}_n)}$. This establishes that $\mathbf{r}'$ is the reduced representative of the equivalence class $\mathbf{r} + (q, \mathbf{\Phi}_n)$. Finally note that $\mathbf{r} \in \mathcal{T}$, so $\mathbf{r} \bmod (q, \mathbf{\Phi}_n) = \mathbf{r}$ and $\mathbf{r}' = \mathbf{r}$. □

3.2 Correctness criteria for specific sample spaces.

Suppose that we take $\mathcal{L}_f, \mathcal{L}_g \subseteq \mathcal{T}_+$ and $\mathcal{L}_r \subseteq \mathcal{T}$. In (7), we have $\mathbf{r}, \mathbf{s} \in \mathcal{T}$ and $\mathbf{f}, \mathbf{g} \in \mathcal{T}_+$. Thus Equation (9) and the triangle inequality imply that

$$|p\mathbf{\Phi}_1 \circledast \mathbf{r} \circledast \mathbf{g} + \mathbf{\Phi}_1 \circledast \mathbf{s} \circledast \mathbf{f}|_\infty \le \sqrt{2}\,(p|\mathbf{r}|_2|\mathbf{g}|_2 + |\mathbf{s}|_2|\mathbf{f}|_2)\,. \tag{10}$$

We can obtain an upper bound on this quantity by maximizing over the relevant sample spaces. Note that

$$\max_{\mathbf{u}\in\mathcal{T}} |\mathbf{u}|_2 = \lfloor p/2 \rfloor \sqrt{n-1} \tag{11}$$

There are two important families of parameter sets, which we will refer to as *uniform* and *fixed norm*.

Uniform: If we consider clean parameters with $\mathcal{L}_f = \mathcal{L}_g = \mathcal{T}_+$ and $\mathcal{L}_r = \mathcal{T}$, then the assumptions of Theorems 1 and 2 are satisfied so long as $q > 2\sqrt{2}(p+1)\lfloor p/2 \rfloor^2(n-1)$. This follows from (7) and (10) by using (11) to bound each of $|\mathbf{f}|_2$, $|\mathbf{g}|_2$, $|\mathbf{r}|_2$, and $|\mathbf{s}|_2$. When $p = 3$ the condition is simply $q > 8\sqrt{2}(n-1)$.

Fixed norm: We can do better by taking $\mathcal{L}_f = \mathcal{L}_g = \mathcal{T}_+(d)$ and $\mathcal{L}_r = \mathcal{T}(d)$. Then we know that $|\mathbf{f}|_2 = |\mathbf{g}|_2 = |\mathbf{r}|_2 = \sqrt{d}$. Using (11) for a bound on the $|\mathbf{s}|_2$ term yields $q > 2\sqrt{2}\left(pd + \lfloor p/2 \rfloor \sqrt{d(n-1)}\right)$.

4 OW-CPA public key encryption schemes from NTRU-HPS.

In this section we present variants of NTRU-HPS that mirror the NTRU-HRSS variants above as closely as possible. To the best of our knowledge the encryption schemes presented in Figures 5 and 6 have not appeared elsewhere. That said, the only novelties are the elimination of invertibility tests in key generation and use of reduction modulo $\mathbf{\Phi}_n$ in decapsulation.

```
KeyGen$_3$($seed$)
$\mathbf{g} \leftarrow \mathsf{Sample}\mathcal{L}_g(seed, \mathtt{domg})$
$\mathbf{f} \leftarrow \mathsf{Sample}\mathcal{L}_f(seed, \mathtt{domf})$
$\mathbf{f}_q \leftarrow (1/\mathbf{f}) \bmod (q, \mathbf{\Phi}_n)$
$\mathbf{f}_p \leftarrow (1/\mathbf{f}) \bmod (p, \mathbf{\Phi}_n)$
$\mathbf{h} \leftarrow p\mathbf{g}\mathbf{f}_q \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
return $dk = (\mathbf{f}, \mathbf{f}_p),\ ek = \mathbf{h}$
```

```
Enc$_3$($\mathbf{h}, coins, \mathbf{m}$) with $\mathbf{m} \in \mathcal{L}_m$
$\mathbf{r} \leftarrow \mathsf{Sample}\mathcal{L}_r(coins, \mathtt{domr})$
$\mathbf{c} \leftarrow \mathbf{r}\mathbf{h} + \mathbf{m} \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
return $\mathbf{c}$
```

```
Dec$_3$($(\mathbf{f}, \mathbf{f}_p), \mathbf{c}$)
$\mathbf{a} \leftarrow \mathbf{c}\mathbf{f} \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
$\mathbf{m}' \leftarrow \mathbf{a}\mathbf{f}_p \bmod (p, \mathbf{\Phi}_n)$
if $\mathbf{m}' \notin \mathcal{L}_m$ return $\bot$
else return $\mathbf{m}'$
```

Fig. 5: A probabilistic public key encryption scheme using the NTRU-HPS one-way function.

```
KeyGen$_4$($seed$)
$\mathbf{g} \leftarrow \mathsf{Sample}\mathcal{L}_g(seed, \mathtt{domg})$
$\mathbf{f} \leftarrow \mathsf{Sample}\mathcal{L}_f(seed, \mathtt{domf})$
$\mathbf{f}_q \leftarrow (1/\mathbf{f}) \bmod (q, \mathbf{\Phi}_n)$
$\mathbf{f}_p \leftarrow (1/\mathbf{f}) \bmod (p, \mathbf{\Phi}_n)$
$\mathbf{h} \leftarrow (p\mathbf{g}\mathbf{f}_q) \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
$\mathbf{h}_q \leftarrow (1/\mathbf{h}) \bmod (q, \mathbf{\Phi}_n)$
return $dk = (\mathbf{f}, \mathbf{f}_p, \mathbf{h}_q),\ ek = \mathbf{h}$
```

```
Enc$_4$($\mathbf{h}, (\mathbf{r}, \mathbf{m})$) with $\mathbf{r} \in \mathcal{L}_r, \mathbf{m} \in \mathcal{L}_m$
$\mathbf{c} \leftarrow (\mathbf{r}\mathbf{h} + \mathbf{m}) \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
return $\mathbf{c}$
```

```
Dec$_4$($(\mathbf{f}, \mathbf{f}_p, \mathbf{h}_q), \mathbf{c}$)
$\mathbf{a} \leftarrow (\mathbf{c}\mathbf{f}) \bmod (q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$
$\mathbf{m}' \leftarrow (\mathbf{a}\mathbf{f}_p) \bmod (p, \mathbf{\Phi}_n)$
$\mathbf{b} \leftarrow (\mathbf{c} - \mathbf{m}') \bmod (q, \mathbf{\Phi}_n)$
$\mathbf{r}' \leftarrow (\mathbf{b}\mathbf{h}_q) \bmod (q, \mathbf{\Phi}_n)$
if $(\mathbf{r}', \mathbf{m}') \notin \mathcal{L}_r \times \mathcal{L}_m$ return $\bot$
else return $(\mathbf{r}', \mathbf{m}')$
```

Fig. 6: A deterministic public key encryption scheme using the NTRU-HPS one-way function.

4.1 Correctness for the NTRU-HPS PPKE and DPKE schemes.

Line 2 of $\mathsf{Enc}_3(\mathbf{h}, coins, \mathbf{m})$ computes $\mathcal{E}_{\text{HPS}}(\mathbf{h}, \mathbf{r}, \mathbf{m})$, for some $\mathbf{r}$ determined by *coins*, and Lines 1-2 of $\mathsf{Dec}_1((\mathbf{f}, \mathbf{f}_p), \mathbf{c})$ compute $\mathcal{D}_{\text{HPS}}(\mathbf{f}, \mathbf{f}_p, \mathbf{c}) \bmod (p, \mathbf{\Phi}_n)$. A proof of correctness largely follows the reasoning around Equation (3). The extra reduction modulo $(p, \mathbf{\Phi}_n)$ allows us to replace the condition on $(\mathbf{f}, \mathbf{f}_p)$ that is expressed by Equation (1) with the more relaxed condition of Equation (4). As in Section 3.1, parameters must be chosen such that three conditions are satisfied:

1. Equations (4) and (2) hold for all $((\mathbf{f}, \mathbf{f}_p), \mathbf{h})$ output by KeyGen_3;
2. The inequality (3) holds for all $\mathbf{f} \in \mathcal{L}_f$, $\mathbf{g} \in \mathcal{L}_g$, $\mathbf{r} \in \mathcal{L}_r$, and $\mathbf{m} \in \mathcal{L}_m$; and
3. For all $\mathbf{m} \in \mathcal{L}_m$ we have $\mathbf{m} = \mathbf{m} \bmod (p, \mathbf{\Phi}_n)$.

The inverses in Lines 3 and 4 of KeyGen_3 exist when the parameters are clean. However, this alone does not imply that the Equation (2) is satisfied. To see the issue, suppose that $\mathbf{f}(1) = 0$ and that there does not exist $\mathbf{g} \in \mathcal{L}_g$ with $\mathbf{g}(1) = 0$. The issue can be resolved by taking taking $\mathcal{L}_g$ to be a subset of $\mathcal{T}_0$. This has the added benefit of ensuring that $\mathbf{h}(1) = 0$. The second condition can be enforced through the choice of q and the sample spaces. The third condition holds so long as $\mathcal{L}_m \subseteq \mathcal{T}$, hence it holds for clean parameters.

Theorem 3. *Suppose that $(n, p, q, \mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_r, \mathcal{L}_m)$ is a clean parameter set and that $\mathcal{L}_g$ and $\mathcal{L}_m$ are subsets of $\mathcal{T}_0$. Further suppose that (3) is satisfied for all $\mathbf{f} \in \mathcal{L}_f$, $\mathbf{g} \in \mathcal{L}_g$, $\mathbf{r} \in \mathcal{L}_r$, and $\mathbf{m} \in \mathcal{L}_m$. Then* $(\mathsf{KeyGen}_3, \mathsf{Enc}_3, \mathsf{Dec}_3)$ *is a correct probabilistic public key encryption scheme.*

Theorem 4. *With the conditions of Theorem 3* $(\mathsf{KeyGen}_4, \mathsf{Enc}_4, \mathsf{Dec}_4)$ *is a correct deterministic public key encryption scheme.*

Remark 2. The condition $\mathcal{L}_m \subseteq \mathcal{T}_0$ is not necessary for correctness, but it eliminates any impact that the $\mathbf{x} \mapsto 1$ homomorphism might have on the one-wayness of the scheme.

4.2 Correctness conditions for specific NTRU-HPS parameter sets.

Equation (8) and the triangle inequality give

$$|p\mathbf{r} \circledast \mathbf{g} + \mathbf{m} \circledast \mathbf{f}|_\infty \leq p|\mathbf{r}|_2|\mathbf{g}|_2 + |\mathbf{m}|_2|\mathbf{f}|_2. \tag{12}$$

Alternatively, the triangle inequality alone gives

$$|p\mathbf{r} \circledast \mathbf{g} + \mathbf{m} \circledast \mathbf{f}|_\infty \leq p|\mathbf{r}|_\infty|\mathbf{g}|_1 + |\mathbf{m}|_1|\mathbf{f}|_\infty. \tag{13}$$

For NTRU-HPS we will only consider *fixed norm* parameter sets.

Fixed norm. Consider a clean parameter set with $\mathcal{L}_g, \mathcal{L}_m \subseteq \mathcal{T}_0(d)$ and $\mathcal{L}_f, \mathcal{L}_r \subseteq \mathcal{T}(d)$ for some positive, even, integer d. The conditions of Theorems 3 and 4 are satisfied so long as $q > 2(p+1)d$. This follows directly from Equation (12). The same condition applies when $p = 3$, $\mathcal{L}_g, \mathcal{L}_m \subseteq \mathcal{T}_0(d)$, and $\mathcal{L}_f, \mathcal{L}_r \subseteq \mathcal{T}$. In this case the condition follows from Equation (13).

5 IND-CCA2 KEM constructions.

Having defined OW-CPA PPKE and OW-CPA DPKE schemes from both NTRU-HPS and NTRU-HRSS, we now have some freedom in constructing IND-CCA2 KEMs. The NTRUEncrypt and NTRU-HRSS-KEM submissions to the NIST process both build KEMs from PPKE schemes. The NTRU Prime submission builds a KEM from a DPKE scheme. Here we will focus on a OW-CPA DPKE to IND-CCA2 KEM conversion, which is tight in the random oracle model, so that we can make fair comparisons with Streamlined NTRU Prime parameter sets.

For our purposes, it is important to check that there are no interactions between the choice of the CCA conversion and the choice of parameters. Such an interaction could affect the size vs. security graph. The NTRUEncrypt and NTRU-HRSS-KEM submissions have such interactions:

- The NTRU-HRSS-KEM submission includes a length-preserving hash of the message as part of the ciphertext. This hash accounts for 141 bytes of an ntruhrss701 ciphertext. The size of this hash can be reduced by changing how $\mathbf{m}$ is sampled (see [9, Section 5]), so the scheme has additional size vs. security vs. efficiency trade-offs.
- The NAEP transformation used by the NTRUEncrypt submission is incompatible with fixed-type $\mathbf{m}$. Using fixed-type $\mathbf{m}$, and optimizing q, can decrease size and increase security.

The KEM in Figure 7 makes use parameters $(n, p, q, \mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_r, \mathcal{L}_m)$ and algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ that define a correct DPKE. The parameter ℓ is the bit length of session key that is the output of the protocol. The function H is a random oracle $\mathcal{L}_r \times \mathcal{L}_m \to \{0,1\}^\ell$ and H' is a random oracle $\mathcal{R} \times \{0,1\}^\ell \to \{0,1\}^\ell$.

```
CCAKeyGen()
1. $seed \leftarrow_\$ \{0,1\}^\ell$
2. $s \leftarrow_\$ \{0,1\}^\ell$
3. $(dk', ek) \leftarrow \mathsf{KeyGen}(seed)$
4. $dk \leftarrow (dk', ek, s)$
5. return $(dk, ek)$
```

```
CCAEncaps($ek$)
1. $coins \leftarrow_\$ \{0,1\}^\ell$
2. $\mathbf{r} \leftarrow \mathsf{Sample}\mathcal{L}_r(coins, \mathtt{domr})$
3. $\mathbf{m} \leftarrow \mathsf{Sample}\mathcal{L}_m(coins, \mathtt{domm})$
4. $\mathbf{c} \leftarrow \mathsf{Enc}(ek, (\mathbf{r}, \mathbf{m}))$
5. $K \leftarrow H((\mathbf{r}, \mathbf{m}))$
6. return $(\mathbf{c}, K)$
```

```
CCADecaps($dk, \mathbf{c}$) with $dk = (dk', ek, s)$
1. $result \leftarrow \mathsf{Dec}(dk', \mathbf{c})$
2. if $result = \perp$ or $\mathbf{c} \neq \mathsf{Enc}(ek, result)$
3. then $K \leftarrow H'(\mathbf{c}, s)$
4. else $K \leftarrow H(result)$
5. return $K$
```

Fig. 7: A KEM that implicitly rejects invalid ciphertexts by producing a random session key.

The KEM in Figure 7 with $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}) = (\mathsf{KeyGen}_2, \mathsf{Enc}_2, \mathsf{Dec}_2)$ was described by Saito, Xagawa, and Yamakawa in [10]. See [10], and references therein, for security reductions.

5.1 Re-using partial results during re-encapsulation.

The re-encapsulation in Line 2 of CCADecaps may be able to re-use some values that are computed during the decapsulation in Line 1 of CCADecaps. For instance, the value $\mathsf{Lift}(\mathbf{m}')$ that is computed in Line 3 of Dec_2 can be re-used in Line 1 of Enc_2. More importantly, the value $\mathbf{b}$ that is computed in Line 3 of Dec_2 (resp. Line 3 of Dec_4) can be used to avoid the rather expensive computation of $\mathbf{r} \circledast \mathbf{h}$ in Line 1 of Enc_2 (resp. Line 1 of Enc_4). The procedure for doing so is given in Figure 8. Proposition 1 shows that the procedure is correct.

```
ReEnc2($\mathbf{b}, \mathbf{m}$)
1. $t \leftarrow (-\mathbf{b}(1)/n) \bmod q$
2. $\mathbf{c} \leftarrow \mathbf{b} + t\mathbf{\Phi}_n + \mathsf{Lift}(\mathbf{m}) \bmod q$
3. return $\mathbf{c}$
```

```
ReEnc4($\mathbf{b}, \mathbf{m}$)
1. $t \leftarrow (-\mathbf{b}(1)/n) \bmod q$
2. $\mathbf{c} \leftarrow \mathbf{b} + t\mathbf{\Phi}_n + \mathbf{m} \bmod q$
3. return $\mathbf{c}$
```

Fig. 8: Re-encapsulation procedures that re-use the value $\mathbf{b}$ computed in Dec_2 and Dec_4.

Proposition 1. *Suppose $(n, p, q, \mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_r, \mathcal{L}_m)$ meet the conditions of Theorem 2, that $((\mathbf{f}, \mathbf{f}_p, \mathbf{h}_q), \mathbf{h})$ is a key pair generated using* KeyGen_2*, and that $\mathbf{c} \in \mathcal{R}$. Suppose further that* $\mathsf{Dec}_2((\mathbf{f}, \mathbf{f}_p, \mathbf{h}_q), \mathbf{c})$ *outputs some $(\mathbf{r}', \mathbf{m}') \in \mathcal{L}_r \times \mathcal{L}_m$ and not $\perp$. Let $\mathbf{b}$ be the value produced in Line 3 of* $\mathsf{Dec}_2((\mathbf{f}, \mathbf{f}_p, \mathbf{h}_q), \mathbf{c})$*. Then* $\mathsf{ReEnc}_2(\mathbf{b}, \mathbf{m}') = \mathsf{Enc}_2(\mathbf{h}, (\mathbf{r}', \mathbf{m}'))$.

Proof. From Line 4 of $\mathsf{Dec}_2((\mathbf{f}, \mathbf{f}_p, \mathbf{h}_q), \mathbf{c})$ we have $\mathbf{r}' = (\mathbf{b}\mathbf{h}_q) \bmod (q, \mathbf{\Phi}_n)$, hence $\mathbf{b} \equiv \mathbf{r}'\mathbf{h} \pmod{(q, \mathbf{\Phi}_n)}$ and $\mathsf{ReEnc}_2(\mathbf{b}, \mathbf{m}') \equiv \mathsf{Enc}_2(\mathbf{h}, (\mathbf{r}', \mathbf{m}')) \pmod{(q, \mathbf{\Phi}_n)}$. Since $\mathbf{b}$ has degree at most $n-2$, $\mathsf{ReEnc}_2(\mathbf{b}, \mathbf{m}')$ has degree $n-1$ and is a reduced representative mod $(q, \mathbf{\Phi}_1\mathbf{\Phi}_n)$. To prove the claim it suffices to show that $\mathsf{ReEnc}_2(\mathbf{b}, \mathbf{m}') \equiv \mathsf{Enc}_2(\mathbf{h}, (\mathbf{r}', \mathbf{m}')) \pmod{(q, \mathbf{\Phi}_1)}$. Note that the right hand side is congruent to 0, and with $t = (-\mathbf{b}(1)/n) \bmod q$ we have $(\mathbf{b} + t\mathbf{\Phi}_n + \mathsf{Lift}(\mathbf{m}'))(1) \equiv 0 \pmod{q}$. □

Remark 3. The analogous proposition for ReEnc_4 needs the assumption that $\mathcal{L}_g \subseteq \mathcal{T}_0$ and $\mathcal{L}_m \subseteq \mathcal{T}_0$. The proof is identical modulo the use of Lift.

Remark 4. The ReEnc_2 and ReEnc_4 functions do not need $\mathbf{h}$, so implementations that use these routines can omit $\mathbf{h}$ from the decapsulation key. Note that the decapsulation key will still contain a copy of $\mathbf{h}_q$.

6 Size vs. security comparisons.

Figures 9 and 10 depict size vs. security trade-offs for the parameter sets discussed in Sections 3.2 and 4.2. Figures 11 and 12 include Streamlined NTRU Prime parameter sets as well. Figures 13 and 14 show the effect of using $p \in \{2, 3, 5\}$ for uniform NTRU-HRSS with prime q.

In all of the graphs, size is measured as the total number of bytes in a public key plus the total number of bytes in a ciphertext. For prime q parameter sets, we assume an optimal encoding of $\{0, 1, \ldots, q-1\}^{n'}$ into $\lceil n' \log_2(q)/8 \rceil$ bytes ($n' = n - 1$ for NTRU-HPS and NTRU-HRSS, $n' = n$ for Streamlined NTRU Prime). Streamlined NTRU Prime ciphertexts include a plaintext confirmation hash that we have not included in our size calculation.

The security estimates in Figures 9 and 11 are based on a "Core-SVP" model that uses $2^{0.292\beta}$ for the cost of solving SVP in dimension β. See [1] for background on the Core-SVP model. The security estimates in Figures 10 and 11 are based on an analysis of Howgrave-Graham's hybrid attack. Our analysis is similar to those of [6,4,9].

We have included a hybrid attack analysis because it shows — more clearly than the Core-SVP model — that a small weight parameter can be detrimental to the size vs. security trade-off. For our hybrid attack analysis we assume that the cost of solving SVP in dimension β is $2^{C_1(\beta)}$ with $C_1(\beta) = (\beta/2e)\log_2(\beta) - \beta + 16$. We assume that the cost of meet-in-the-middle search on K coefficients is $2^{K \cdot S(P)/2}$ where $S(P)$ is the Shannon entropy of the probability distribution for a single coefficient. For fixed-weight and fixed-type vectors with parameter d we take $P(0) = 1 - d/n$ and $P(1) = P(-1)$. Fixed-weight/type vectors do not have independent and identically distributed (i.i.d.) coefficients, so $2^{K \cdot S(P)}$ is not necessarily a good approximation to the size of the set of "typical" K-coefficient patterns. Nevertheless, after comparing our results with [4], which used a more refined estimate for the size of the set of typical K-coefficient patterns, we believe that the impact of the i.i.d. assumption is small for parameter sets with $n/3 \leq d \leq 2n/3$.

Our software provides several options for the cost of solving SVP in dimension β. Using a cost of $2^{C_2(\beta)}$ with $C_2(\beta) = 0.000784314\beta^2 + 0.366078\beta - 6.125$ we have compared our security estimates for Streamlined NTRU Prime parameter sets to those reported in [4, Appendix P]. Apart from the i.i.d. assumption, our analysis differs from that of [4] in several important ways. First, we have not used the simulator of [5]. Second, we have ignored the polynomial number of calls to the SVP solver that are made by BKZ (we cost a single call). Third, we have omitted a factor of 2^7 from the cost of solving SVP. As expected, our security estimates are lower than those of [4] for all 212 parameter sets listed in [4, Appendix P]. For the 178 parameter sets with $n/3 \leq d \leq 2n/3$, our security estimates are between 3 and 7 bits lower[1]. The average discrepancy is 5.22 bits. The maximum discrepancy across all 212 parameter sets is 13 bits for a parameter set with weight $d = \lfloor 0.07n \rfloor$.

We have not incorporated the results of [11] into our hybrid attack cost estimates.

Filtering of parameter sets. Figures 9, 10, 11, and 12 only include parameter sets with $p = 3$. While $p = 2$ appears to be optimal in Figure 13, the very poor performance of $p = 2$ in Figure 14 leads us believe that $p = 3$ is optimal. This is in line with Howgrave–Graham's recommendation in [8] and common practice.

For uniform NTRU-HRSS[2], we have included every clean parameter set with $449 \leq n \leq 941$ and q the smallest prime that provides correctness. We have also included every clean parameter set with $449 \leq n \leq 941$ and q the smallest power of two that provides correctness.

[1] This is after accounting for two errors in [4, Appendix P]. The security of sntrup2437541 was listed as 150, but it should be 143. The security of sntrup4591761 was listed as 248, but it should be 236.

[2] For our security analysis of these parameters we have used the non-uniform coefficient distribution proposed in [9] ($P(0) = 6/16, P(1) = P(-1)$). Using the uniform distribution would lead to a slight increase in security.

For fixed norm NTRU-HPS and NTRU-HRSS, we have included every clean parameter set with $449 \leq n \leq 941$, $d \in \{\lfloor 3n/8 \rfloor, \lfloor n/2 \rfloor, \lfloor 3n/5 \rfloor, \lfloor 2n/3 \rfloor\}$ (or as close to these four values as possible), and q the smallest prime that provides correctness. We have also included parameter sets with n and d of this form and q the smallest power of two that provides correctness. The choice of weight parameters is somewhat arbitrary. However, by comparing the relative position of points between Figures 9 and 10, one can see that the hybrid attack has a greater impact on the relative security of parameter sets with $d = \lfloor 3n/8 \rfloor$ than it does on parameter sets with $d \in \{\lfloor n/2 \rfloor, \lfloor 3n/5 \rfloor, \lfloor 2n/3 \rfloor\}$. We suspect that improved combinatorial attacks will further separate these cases, and that the optimal choice of d lies in $[n/2, 2n/3]$.

We have included all of the Streamlined NTRU Prime parameter sets that were listed in [4, Appendix P]; i.e. all parameter sets with (n, q) that are consistent with the requirements of the scheme, have $500 < n < 950$, and have $n < q < 20000$. The weight parameter of each parameter set is the minimum of $2\lfloor q/32 \rfloor$ and $2\lfloor n/3 \rfloor$. Some of these parameter sets have small weight or unnecessarily large q. For comparison with our NTRU-HPS and NTRU-HRSS parameter sets, we have highlighted the Streamlined NTRU Prime parameter sets with $n/3 \leq d \leq 2n/3$ and $q < 18n$.

7 Notes on specific parameter sets.

Naming conventions. We write ntruhrss[n], e.g. ntruhrss701, for an NTRU-HRSS parameter set with $p = 3$, $q = 2^{\lceil 7/2 + \log_2(n) \rceil}$, $\mathcal{L}_f = \mathcal{L}_g = \mathcal{T}_+$, and $\mathcal{L}_r = \mathcal{L}_m = \mathcal{T}$. We write ntruhps[q][n], e.g. ntruhps2048509, for an NTRU-HPS parameter set with $p = 3$, and d the largest even value that provides correctness when $\mathcal{L}_g = \mathcal{L}_m = \mathcal{T}_0(d)$ and $\mathcal{L}_f = \mathcal{L}_r = \mathcal{T}$. We write sntrup[q][n] for Streamlined NTRU Prime parameter sets.

Uniform NTRU-HRSS *parameters.* The prime q parameter sets in this family have fairly consistent size vs. security trade-offs. They are available at wide range of security levels, and there is little that distinguishes any particular parameter set. On the other hand, there is a sharp discontinuity in the size vs. security graph for parameter sets that use power of two q. The parameter set ntruhrss701 has the best size vs. security trade-off among uniform NTRU-HRSS parameter sets with power of two q.

Fixed-weight NTRU-HRSS *parameters.* It is unlikely that fixed-weight NTRU-HRSS parameter sets are more efficient than fixed-type NTRU-HPS parameter sets. Since the latter typically provide better size vs. security trade-offs, we have chosen not to highlight any fixed-weight NTRU-HRSS parameter sets.

Fixed-type NTRU-HPS *parameters.* Again, the prime q parameter sets in this family have fairly consistent size vs. security trade-offs for any particular weight. Some parameter sets with power of two q stand out as having particularly good size vs. security trade-offs. We have highlighted a few of these parameter sets.

1. The parameter set ntruhps2048509 has an excellent size vs. security trade-off, especially considering the weight parameter ($254 = \lfloor n/2 \rfloor$) and the use of power of two q. With a Core-SVP cost estimate of 2^{106}, it is possible that an attack on the corresponding KEM would be as costly as key search on a block cipher with a 128-bit key.

2. The parameter set ntruhps2048677 also has an excellent size vs. security trade-off, especially considering the use of power of two q. The weight parameter is fairly low ($254 = \lceil 3n/8 \rceil$), and may lower confidence in our security analyses. Nevertheless, the security analyses that we have considered place the parameter set higher than ntruhrss701. The combined public key + ciphertext size of this parameter set is also 416 bytes smaller than that of ntruhrss701.

3. The parameter set ntruhps4096701 is among the best analogues for sntrup4591761 in terms of size and security. The weight parameter ($466 = \lfloor 2n/3 \rfloor$) is higher than that of sntrup4591761 ($286 = 1 + \lfloor 3n/8 \rfloor$). A more detailed comparison of the two might consider whether a low weight parameter is more or less worrisome than the ring structure that Streamlined NTRU Prime was designed to avoid. A similar parameter set was described in [3].

4. The parameter set ntruhps4096821 provides an excellent size vs. security trade-off and a high security level. We have plotted this parameter set with weight $492 = \lfloor 3n/5 \rfloor$, however the weight can be increased to $510 = \lfloor 0.622n \rfloor$ while preserving correctness. With a Core-SVP cost estimate of 2^{178} it is possible that an attack on the corresponding KEM would be as costly as key search on a block cipher with a 192-bit key.

Streamlined NTRU Prime parameters.

1. The parameter set sntrup4591761 was used as a case study in [4]. It was also recommended in the NTRU Prime NIST submission. It has an excellent size vs. security tradeoff. Its weight parameter is $286 = \lfloor 0.376 \cdot n \rfloor$.
2. The parameter set sntrup7541743 is a good analogue of sntrup4591761 with a larger weight parameter ($470 = \lfloor 0.633 \cdot n \rfloor$).
3. The parameter sets sntrup11923709 and sntrup12241727 are good analogues of ntruhrss701. Both have $q > 16n$, so by [4, Theorem 2.1] they are compatible with the use of uniform sampling routines.
4. The parameter set sntrup5167857 provides an excellent size vs. security trade-off and a high security level. It is a good point of comparison with ntruhrss4096821.

References

1. Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the LWE, NTRU schemes! Cryptology ePrint Archive, Report 2018/331, 2018. https://eprint.iacr.org/2018/331. 9
2. Daniel J. Bernstein. Message to the NIST pqc-forum mailing list, 12 May 2018. https://groups.google.com/a/list.nist.gov/d/msg/pqc-forum/l5IaJTe_pUI/QKaLZ4uMAAAJ. 2
3. Daniel J. Bernstein. Lattice-based public-key cryptosystems. Invited lecture at PQCRYPTO Mini-School. Institute for Information Science, Academia Sinica, Taipei, 27 June 2018. https://cr.yp.to/talks.html#2018.06.27-1. 10
4. Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. NTRU Prime: reducing attack surface at low cost. In *International Conference on Selected Areas in Cryptography*, pages 235–260. Springer, 2017. 9, 10, 11
5. Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 1–20. Springer, 2011. http://www.iacr.org/archive/asiacrypt2011/70730001/70730001.pdf. 9
6. Jeff Hoffstein, Jill Pipher, John M. Schanck, Joseph H. Silverman, William Whyte, and Zhenfei Zhang. Choosing parameters for NTRUEncrypt. In Helena Handschuh, editor, *Cryptographers' Track at the RSA Conference – CTA-RSA 2017*, volume 10159 of *LNCS*, pages 3–18. Springer, 2017. https://eprint.iacr.org/2015/708. 9
7. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe P. Buhler, editor, *Algorithmic Number Theory – ANTS-III*, volume 1423 of *LNCS*, pages 267–288. Springer, 1998. http://dx.doi.org/10.1007/BFb0054868. 2
8. Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *LNCS*, pages 150–169. Springer, 2007. http://www.iacr.org/archive/crypto2007/46220150/46220150.pdf. 9
9. Andreas Hülsing, Joost Rijneveld, John Schanck, and Peter Schwabe. High-speed key encapsulation from NTRU. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 232–252. Springer, 2017. 3, 4, 8, 9
10. Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 520–551. Springer, 2018. 1, 4, 8
11. Thomas Wunderer. Revisiting the hybrid attack: Improved analysis and refined security estimates. Cryptology ePrint Archive, Report 2016/733, 2016. https://eprint.iacr.org/2016/733. 9

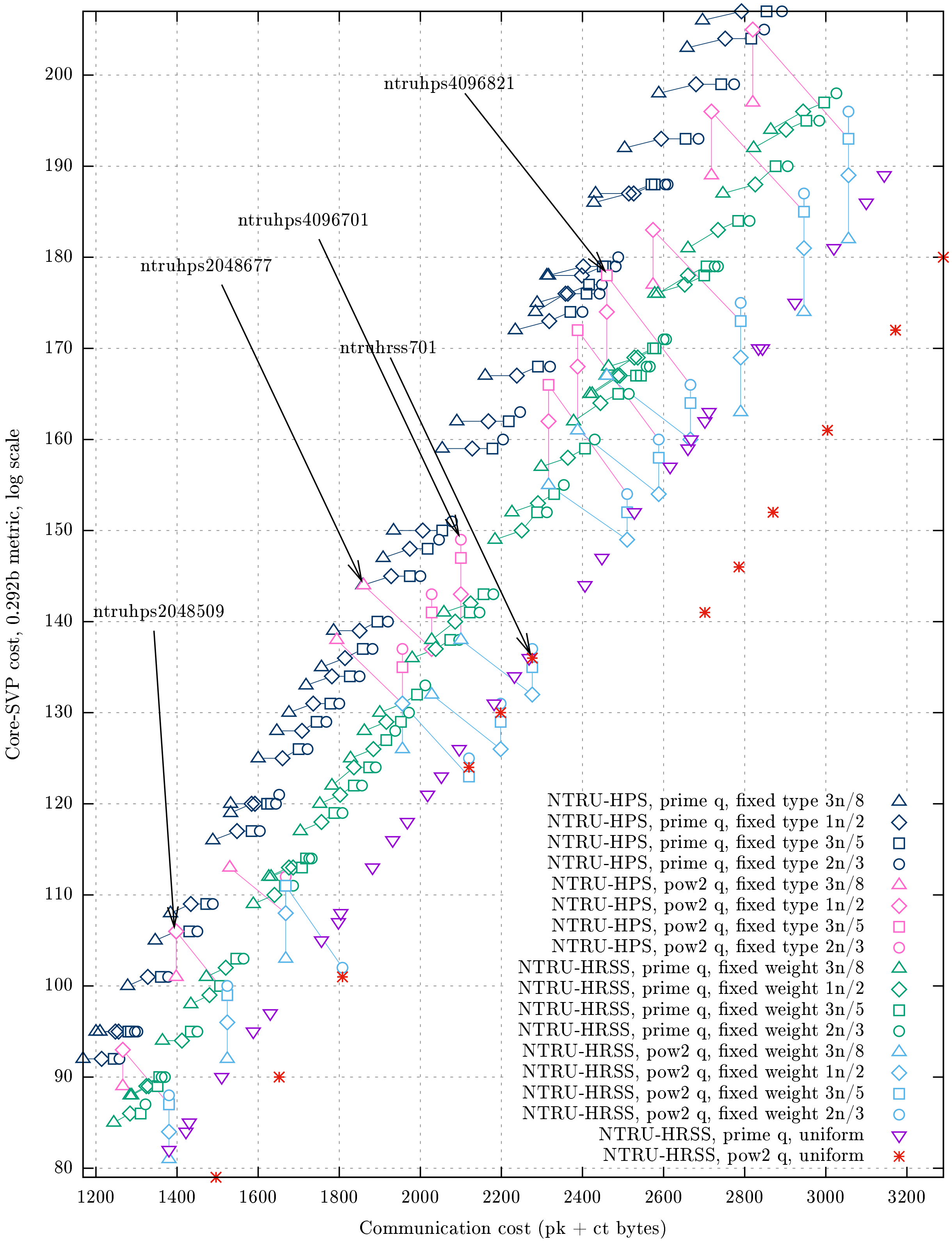

Fig. 9: Size vs. security trade-offs as described in Section 6. Lines connect parameter sets that use the same n. All of the parameter sets use $p = 3$. The "fixed type d" parameter sets take $\mathcal{L}_g = \mathcal{L}_m = \mathcal{T}_0(d)$ and $\mathcal{L}_f = \mathcal{L}_r = \mathcal{T}$. The "fixed weight d" parameter sets take $\mathcal{L}_f = \mathcal{L}_g = \mathcal{T}_+(d)$, $\mathcal{L}_r = \mathcal{T}(d)$, and $\mathcal{L}_m = \mathcal{T}$. All parameter sets are clean, correct, and use the smallest q available. Security is evaluated using a Core-SVP model.

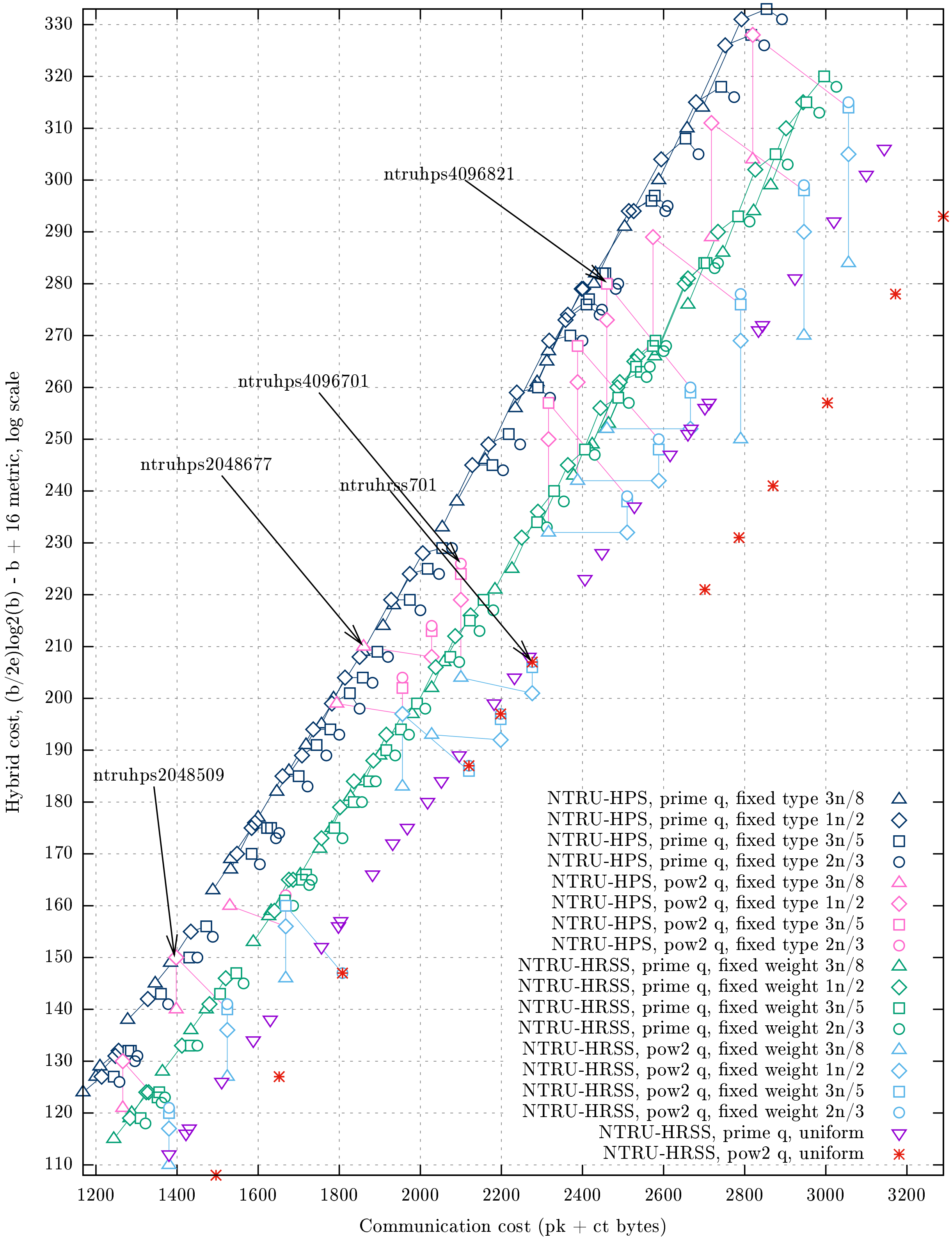

Fig. 10: Size vs. security trade-offs as described in Section 6. Lines connect parameter sets that use the same n. All of the parameter sets use $p = 3$. The "fixed type d" parameter sets take $\mathcal{L}_g = \mathcal{L}_m = \mathcal{T}_0(d)$ and $\mathcal{L}_f = \mathcal{L}_r = \mathcal{T}$. The "fixed weight d" parameter sets take $\mathcal{L}_f = \mathcal{L}_g = \mathcal{T}_+(d)$, $\mathcal{L}_r = \mathcal{T}(d)$, and $\mathcal{L}_m = \mathcal{T}$. All parameter sets are clean, correct, and use the smallest q available. Security is evaluated with respect to the hybrid attack.

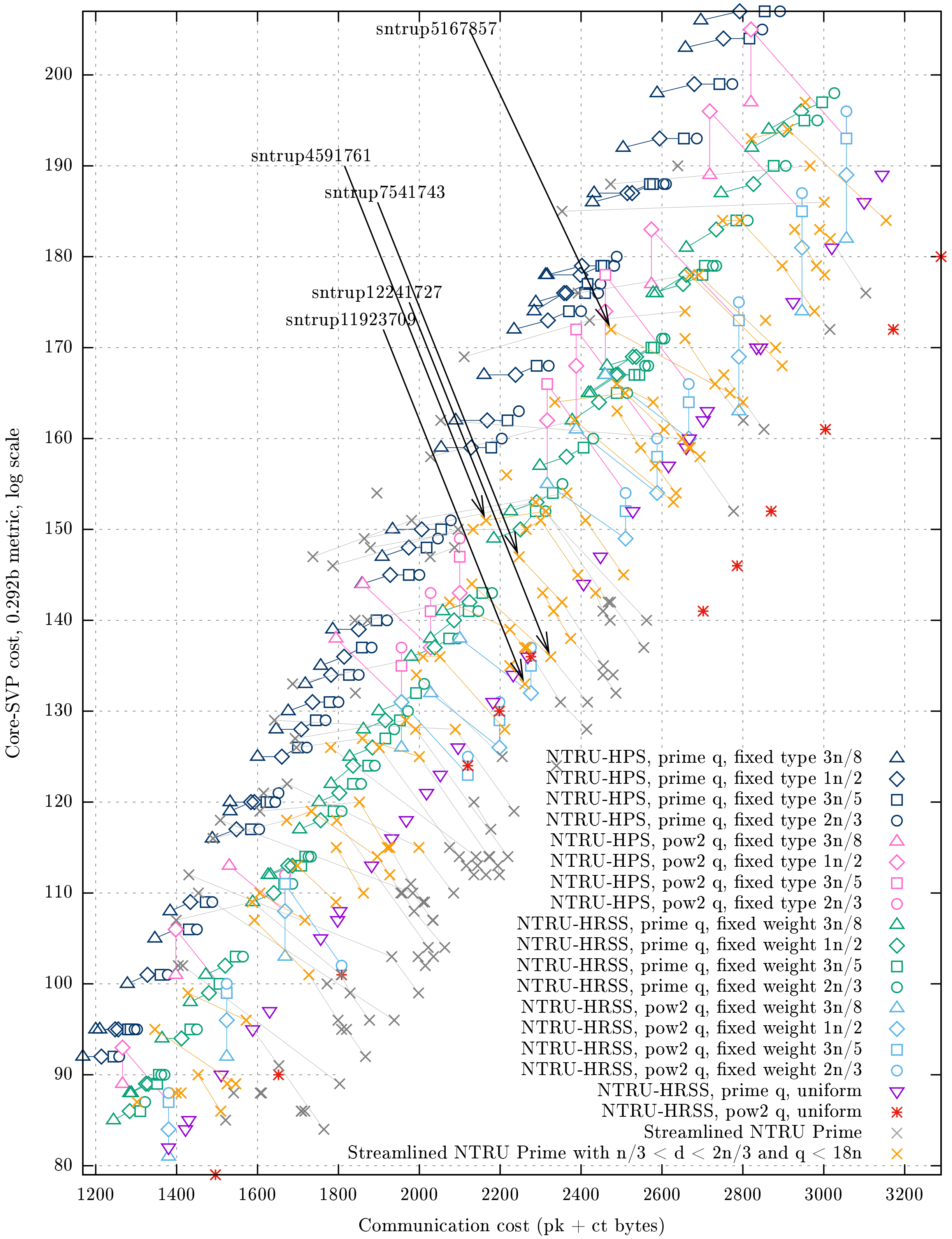

Fig. 11: The data of Figure 9 plotted alongside Streamlined NTRU Prime parameters. The same Core-SVP analysis has been applied to the Streamlined NTRU Prime parameters.

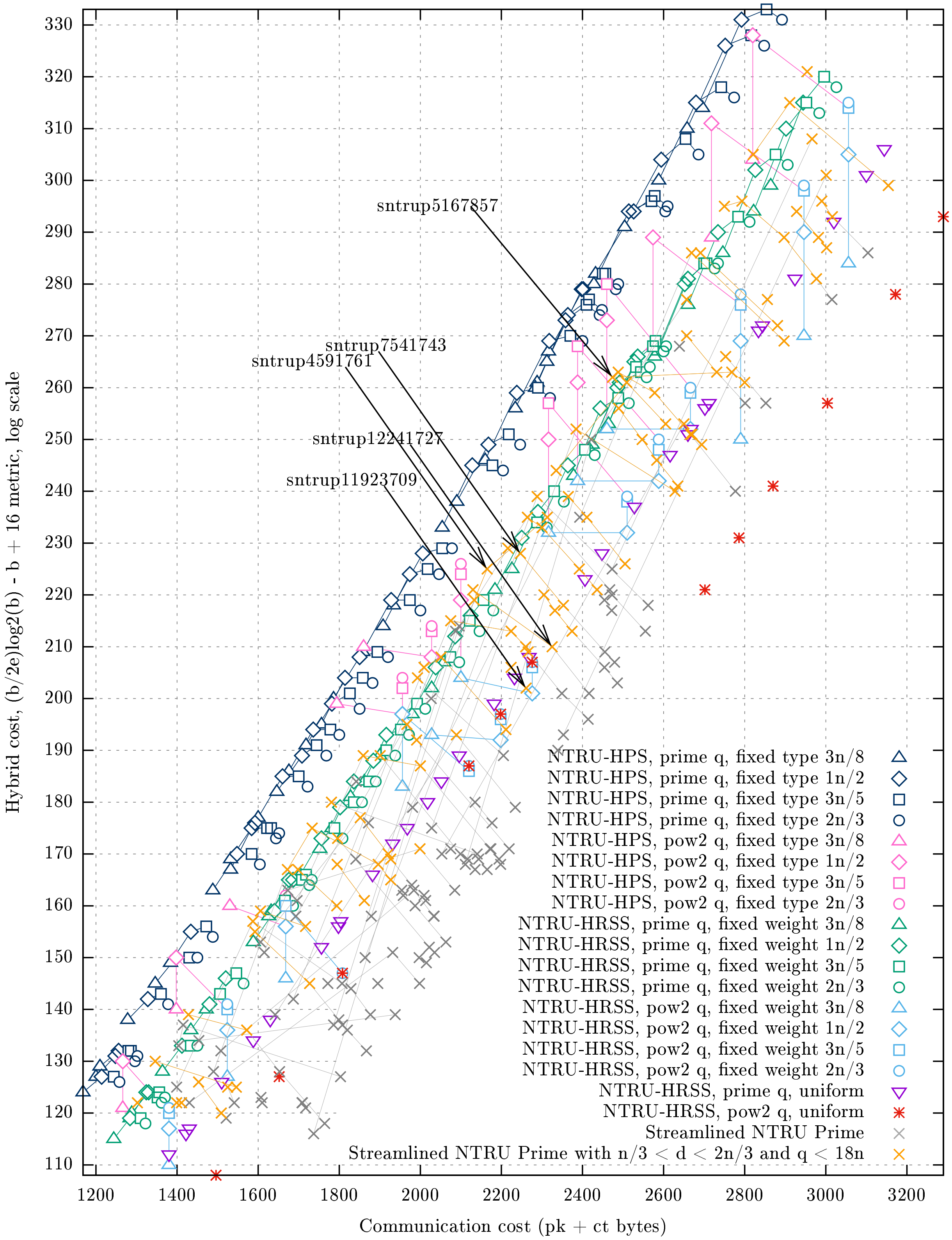

Fig. 12: The data from Figure 10 plotted alongside Streamlined NTRU Prime parameters. The same hybrid attack analysis has been applied to the Streamlined NTRU Prime parameters.

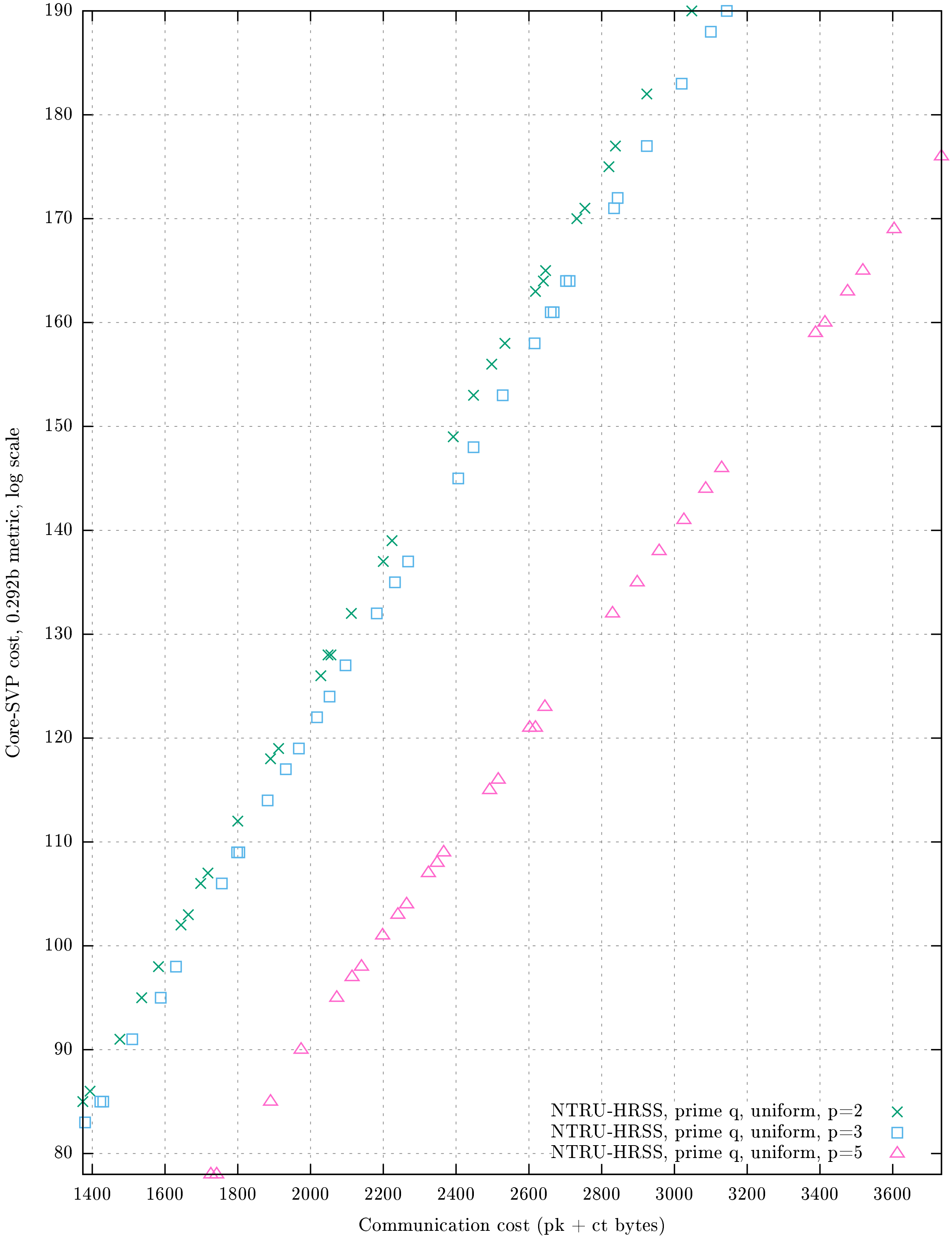

Fig. 13: Size vs. security trade-offs for NTRU-HRSS parameters with q prime and $p \in \{2, 3, 5\}$. Security is evaluated using a Core-SVP model.

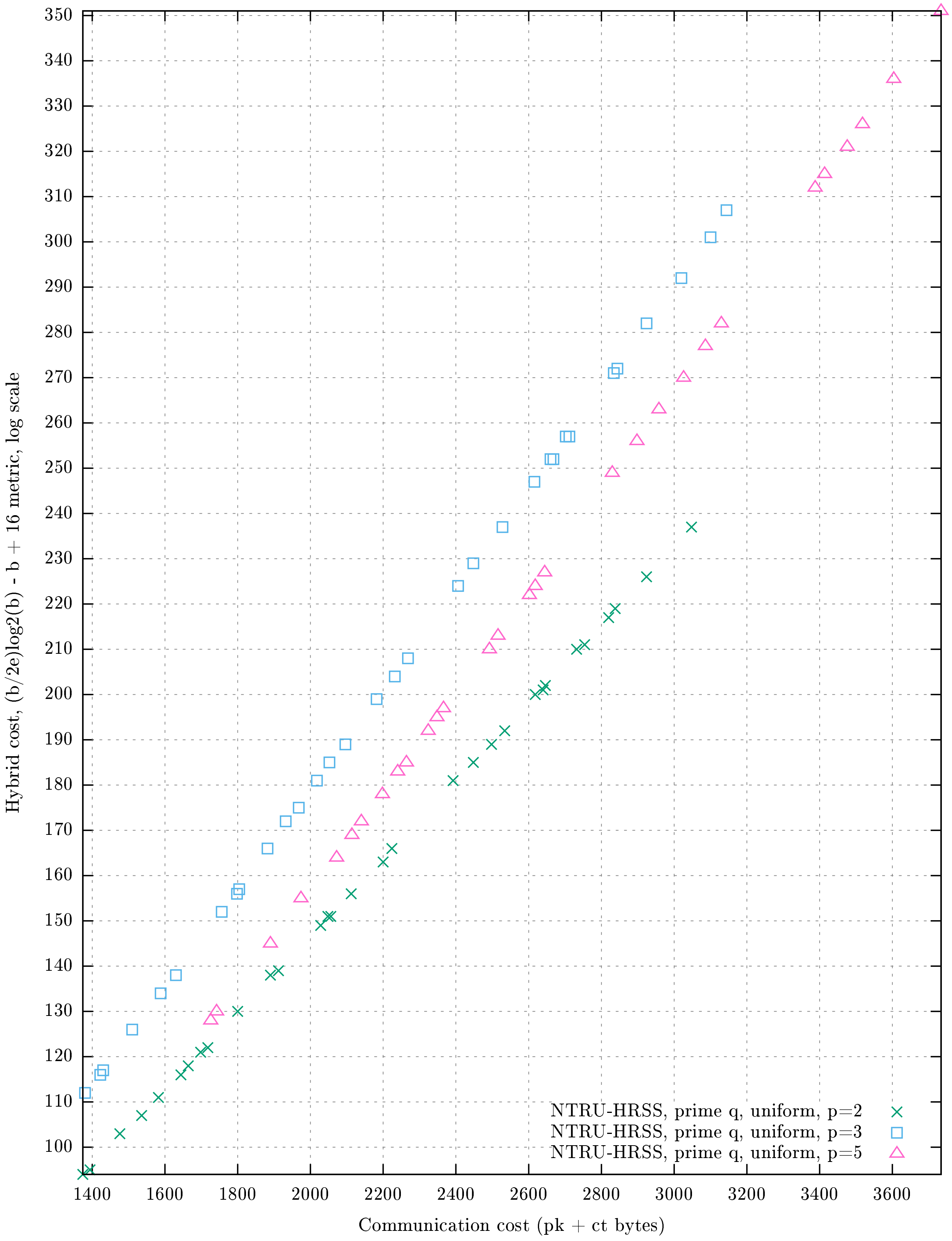

Fig. 14: Size vs. security trade-offs for NTRU-HRSS parameters with q prime and $p \in \{2, 3, 5\}$. Security is evaluated using with respect to the hybrid attack.