Submission to NIST's post-quantum project: lattice-based digital signature scheme qTESLA

Name of the cryptosystem: qTESLA

Principal and auxiliary submitters:

Nina Bindel, (Principal submitter)	Technische Universität Darmstadt, Hochschulstrasse 10, 64289 Darmstadt, Germany, Email: `nbindel@cdc.informatik.tu-darmstadt.de`, Phone: 004961511620667
	Signature: Nina Bindel
Sedat Akleylek,	Ondokuz Mayis University, Turkey
Erdem Alkim,	Ege University, Turkey
Paulo S. L. M. Barreto,	University of Washington Tacoma, USA
Johannes Buchmann,	Technische Universität Darmstadt, Germany
Edward Eaton,	ISARA Corporation, Canada
Gus Gutoski,	ISARA Corporation, Canada
Juliane Krämer,	Technische Universität Darmstadt, Germany
Patrick Longa,	Microsoft Research, USA
Harun Polat,	Technische Universität Darmstadt, Germany
Jefferson E. Ricardini,	University of São Paulo, Brazil
Gustavo Zanon,	University of São Paulo, Brazil

Inventors of the cryptosystem:

All submitters by name based on a previous scheme by Shi Bai and Steven Galbraith and extensive previous works as explained in the body of this document.

Owners of the cryptosystem:

None (dedicate to the public domain)

Contents

1 Introduction

This document presents a detailed specification of qTESLA, a post-quantum signature scheme based on the hardness of the decisional ring learning with errors (R-LWE) problem. In contrast to other alternatives, qTESLA is a conservative yet efficient signature scheme that has been instantiated according to the provided security reduction. That is, qTESLA instantiations are *provably* secure in the (quantum) random oracle model. To this end, the scheme comes accompanied by a *non-tight* reduction in the random oracle model, and a *tight* reduction in the quantum random oracle model from R-LWE.

Concretely, qTESLA is designed to target *three* security levels:

- qTESLA-128: NIST's security category 1.
- qTESLA-192: NIST's security category 3.
- qTESLA-256: NIST's security category 5.

Despite the aforementioned security assurances in its parameter selection, qTESLA still achieves good performance with a competitive memory footprint. Furthermore, design decisions have been made towards enabling simple, easy-to-protect implementations.

In the remainder of this section, we describe previous works related to the proposed signature scheme qTESLA. In Section 2, we give the specification details of the scheme, including a basic and a formal algorithmic description, the functions that are required for its implementation, and the proposed parameter sets. In Section 3, we analyze the performance of our implementations. Section 4 includes the details of our known answer values. Then, we discuss the (provable) security of our proposal in Section 5, including an analysis of the concrete security level and the security against implementation attacks. Section 6 ends this document with a summary of the advantages and limitations of qTESLA.

1.1 Related work

The signature scheme proposed in this submission is the result of a long line of research. The first work in this line is the signature scheme proposed by Bai and Galbraith [14] which is based on the Fiat-Shamir construction of Lyubashevsky [50]. The scheme by Bai and Galbraith is constructed over standard lattices and comes with a (non-tight) security reduction from the learning with errors (LWE) and the short integer solution problem (SIS) in the random oracle model. Dagdelen *et al.* presented improvements and the first implementation of the Bai-Galbraith scheme [27]. The scheme was subsequently studied under the name TESLA by Alkim, Bindel, Buchmann, Dagdelen, Eaton, Gutoski, Krämer, and Pawlega [9], who provided an alternate security reduction from the LWE problem in the quantum random oracle model.

A variant of TESLA over ideal lattices was derived under the name ring-TESLA [1] by Akleylek, Bindel, Buchmann, Krämer, and Marson. Since then, subsequent works [16,41] have been presented. Most notably, a version of the scheme ring-TESLA called TESLA# [16] by Barreto, Longa, Naehrig, Ricardini, and Zanon included several implementation improvements. Moreover, there exist several works [19, 20, 36] concerned with the analysis of ring-TESLA with respect to implementation attacks, i.e., fault and side-channel attacks.

The signature scheme presented in the following assembles the advantages acquired in the prior works resulting in the quantum-secure signature scheme qTESLA.

Acknowledgments

We are grateful to Michael Naehrig for his valuable feedback.

SA and EA are partially supported by TÜBITAK under grant no. EEEAG-116E279.
NB is partially supported by the German Research Foundation (DFG) as part of project P1 within the CRC 1119 CROSSING.
JR is supported by the joint São Paulo Research Foundation (FAPESP)/Intel Research grant 2015/50520-6 "Efficient Post-Quantum Cryptography for Building Advanced Security Applications".

2 Specification

Next, we give an informal description of the basic scheme that is used to specify qTESLA. A formal specification of qTESLA's key generation, signing and verification algorithms then follows in Section 2.2. The correctness of the scheme is discussed in Section 2.3. We describe the implementation of the functions required by qTESLA in Section 2.4, and explain all the system parameters and the proposed parameter sets in Section 2.5.

2.1 Basic signature scheme

Informal descriptions of the algorithms that give rise to the signature scheme qTESLA are shown in Algorithms 1, 2 and 3. Below, we first define two basic terms that are required by the algorithms, namely, B*-short* and *well-rounded*.

An integer polynomial y is B*-short* if each coefficient is at most B in absolute value. We call an integer polynomial w *well-rounded* if w is $(\lfloor q/2 \rceil - L_E)$-short and $[w]_L$ is $(2^d - L_E)$-short, where $[\cdot]_L$ is the value represented by the d least significant bits of w. Similarly,

Algorithm 1 Informal description of the key generation

Require: -
Ensure: Secret key $sk = (s, e, a)$, public key $pk = (a, t)$

1: $a \leftarrow \mathcal{R}_q$ invertible ring element
2: Choose $s, e \in \mathcal{R}$ with entries from $\mathcal{D}_\sigma$.
3: If the h largest entries of e sum to L_E then sample new e and retry at step 2.
4: If the h largest entries of s sum to L_S then sample new s and retry at step 2.
5: $t = as + e \in \mathcal{R}_q$.
6: Return secret key $sk = (s, e)$ and public key $pk = (a, t)$.

Algorithm 2 Informal description of the signature generation

Require: Message m, secret key $sk = (s, e, a)$,
Ensure: Signature (z, c).

1: Choose y uniformly at random among B-short polynomials in $\mathcal{R}_q$.
2: $c \leftarrow H([ay]_M, m)$.
3: $z \leftarrow y + sc$.
4: If z is not $(B - L_S)$-short then retry at step 1.
5: If $ay - ec$ is not well-rounded then retry at step 1.
6: Return signature (z, c).

Algorithm 3 Informal description of the verification

Require: Message m, public key $pk = (a, t)$, purported signature (z, c)
Ensure: "Accept" or "reject".

1: If z is not $(B - L_S)$-short then return reject.
2: $w \leftarrow az - tc \mod q$
3: If $H([w]_M, m) \neq c$ then return reject.
4: Return accept.

$[\cdot]_M$ is the value represented by the corresponding most significant bits. For simplicity we assume that the hash oracle $H(\cdot)$ maps from $\{0,1\}^*$ to $\mathbb{H}$, where $\mathbb{H}$ denotes the set of polynomials $c \in \mathcal{R}$ with coefficients in $\{-1, 0, 1\}$ with exactly h nonzero entries, i.e., we ignore the encoding function F introduced in Section 2.2.

As can be seen, the description in Algorithm 2 implies that the signature scheme is non-deterministic, i.e., that different randomness is required for each signing operation, even if the message is the same. Specifically, this feature is fixed by the random generation of the polynomial y in Step 1 of Algorithm 2.

In Section 2.2, we discuss how the scheme can be converted to deterministic. Deterministic signatures have the advantage that different randomness is used for different messages with very high probability and that sampling can be implemented more easily since access to a source of high-quality randomness is not needed. We discuss the (dis-)advantages of deterministic vs. probabilistic signatures in more detail in Section 5.4.

2.2 Formal description of qTESLA

Below, we define all the necessary functions, sets, and system parameters in qTESLA.

The description of the scheme depends on the following system parameters: λ, κ, n, q, σ, L_E, L_S, B, d, and h. Let $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$, $\mathcal{R} = \mathbb{Z}[x]/\langle x^n+1\rangle$, $\mathcal{R}_q = \mathbb{Z}_q[x]/\langle x^n+1\rangle$, $\mathcal{R}_{q,[I]} = \{f \in \mathcal{R}_q \mid f = \sum_{i=0}^{n-1} f_i x^i,\ f_i \in [-I, I]\}$, and $\mathbb{H}_{n,h} = \{f \in \mathcal{R}_q \mid f = \sum_{i=0}^{n-1} f_i x^i,\ f_i \in \{-1,0,1\},\ \sum_{i=0}^{n-1} |f_i| = h\}$. Let $\mathcal{R}$ be a ring then we denote the inverse elements in this ring by $\mathcal{R}^\times$. Let $f = \sum_{i=0}^{n-1} f_i x^i \in \mathcal{R}$. Then we define the reduction $(f \mod q)$ of f modulo q to be $(f \mod q) = \sum_{i=0}^{n-1}(f_i \mod q)x^i \in \mathcal{R}_q$. Let $d \in \mathbb{N}$ and $c \in \mathbb{Z}$. We denote by $[c]_L$ the unique integer in $(-2^{d-1}, 2^{d-1}] \subset \mathbb{Z}$ such that $c = [c]_L$ modulo 2^d. Let $[\cdot]_M$ be the function $[\cdot]_M : \mathbb{Z} \to \mathbb{Z}, c \mapsto (c - [c]_L)/2^d$. Furthermore, let $f = \sum_{i=0}^{n-1} f_i x^i \in \mathcal{R}_q$, then $[f]_L = \sum_{i=0}^{n-1} [f_i]_L\, x^i$ and $[f]_M = \sum_{i=0}^{n-1} [f_i]_M\, x^i$. Let $f \in \mathcal{R}_q$ be a polynomial with coefficients being ordered (without losing any generality) as $|f_1| \geq |f_2| \geq \ldots \geq |f_n|$. Then we define $\max_i(f) = f_i$.

The centered discrete Gaussian distribution for $x \in \mathbb{Z}$ with standard deviation σ is defined to be $\mathcal{D}_\sigma = \rho_\sigma(x)/\rho_\sigma(\mathbb{Z})$, where $\sigma > 0$, $\rho_\sigma(x) = \exp(\frac{-x^2}{2\sigma^2})$, and $\rho_\sigma(\mathbb{Z}) = 1 + 2\sum_{x=1}^{\infty} \rho_\sigma(x)$. We write $c \leftarrow_\sigma \mathbb{Z}$ to denote sampling a value c with distribution $\mathcal{D}_\sigma$. For a polynomial $c \in \mathcal{R}$, we write $c \leftarrow_\sigma \mathcal{R}$ to denote sampling each coefficient of c with distribution $\mathcal{D}_\sigma$. For a finite set S, we denote sampling the element s uniformly from S with $s \leftarrow_\$ S$.

We define the following functions (refer to the specified sections for explicit details about their implementation):

- The generation of the polynomial a as $\mathrm{GenA} : \{0,1\}^\kappa \to \mathcal{R}_q^\times$ (cf. Section 2.4.3),
- an encoding function to encode hash values to polynomials $\mathrm{Enc} : \{0,1\}^\kappa \to \mathbb{H}_{n,h}$ (cf. Section 2.4.4),
- the two pseudo random functions $\mathrm{PRF}_1 : \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^\kappa$ and $\mathrm{PRF}_2 : \{0,1\}^\kappa \times \mathbb{Z} \to \mathcal{R}_{q,[B]}$ (cf. Section 2.4.5), and
- a hash function $H : \{0,1\}^* \to \{0,1\}^\kappa$ (cf. Section 2.4.5).

The details of qTESLA's key generation, signing and signature verification are given in Algorithms 6, 7, and 8, respectively. The two subroutines *checkE* and *checkS* that are called during key generation are depicted in Algorithms 4 and 5, respectively.

Algorithm 4 Subroutine *checkE* to ensure correctness of the scheme; *checkE* ensures that $\|ec\|_\infty \leq L_E$

Require: $e \in \mathcal{R}$
Ensure: $\{0, 1\}$ ▷ false, true

```
if $\sum_{i=1}^{h} \max_i(e) > L_E$ then
    return 0
end if
return 1
```

Algorithm 5 Subroutine *checkS* to simplify the security reduction; *checkS* ensures that $\|sc\|_\infty \leq L_S$

Require: $s \in \mathcal{R}$
Ensure: $\{0, 1\}$ ▷ false, true

```
if $\sum_{i=1}^{h} \max_i(s) > L_S$ then
    return 0
end if
return 1
```

Algorithm 6 qTESLA's key generation

Require: -
Ensure: $sk = (s, e, \mathrm{seed}_y, \mathrm{seed}_a)$, $pk = (\mathrm{seed}_a, t)$

```
1: $\mathrm{seed}_a, \mathrm{seed}_y \leftarrow_\$ \{0,1\}^\kappa$
2: $a \leftarrow \mathrm{GenA}(\mathrm{seed}_a)$
3: $s \leftarrow_\sigma \mathcal{R}$
4: if $checkS(s) = 0$ then
5:     Restart at step 3
6: end if
7: $e \leftarrow_\sigma \mathcal{R}$
8: if $checkE(e) = 0$ then
9:     Restart at step 7
10: end if
11: $t = as + e \mod q$
12: $sk \leftarrow (s, e, \mathrm{seed}_y, \mathrm{seed}_a)$
13: $pk \leftarrow (\mathrm{seed}_a, t)$
14: return $sk$, $pk$
```

Remark 1. *We note that the description of our scheme can be easily generalized to use more than one sample of the ring learning with errors problem. In particular, that would mean that the public key consist of* $seed_{a_1}, ..., seed_{a_k}$ *(corresponding to* $a_1, ..., a_k$*) and* $t_1, ..., t_k$*, and that the secret key consist of the polynomials* $s, e_1, ..., e_k, seed_y$*. Our analysis of the expected security also holds for a generalization with* $k > 1$*. However, the description and implementation of the scheme are substantially simpler for* $k = 1$*.*

2.3 Correctness of the scheme

According to Algorithms 6 and 7, the following holds for an honestly generated signature (c', z) with $c = \mathrm{Enc}(c')$ and elements from the key generation a, t, s, e:

Algorithm 7 qTESLA's signature generation

Require: m, $sk = (s, e, \text{seed}_y, \text{seed}_a)$
Ensure: c', z

1: $a \leftarrow \text{GenA}(\text{seed}_a)$
2: $\text{counter} \leftarrow 0$
3: $\text{rand} \leftarrow \text{PRF}_1(\text{seed}_y, m)$
4: $y \leftarrow \text{PRF}_2(\text{rand}, \text{counter})$
5: $v = ay \mod q$
6: $c' \leftarrow H([v]_M, m)$
7: $c \leftarrow \text{Enc}(c')$
8: $z \leftarrow y + sc$
9: **if** $z \notin \mathcal{R}_{q,[B-L_S]}$ **then**
10: $\quad$ $\text{counter}++$
11: $\quad$ Restart at step 4
12: **end if**
13: $w \leftarrow v - ec \mod q$
14: **if** $\|[w]_L\|_\infty > 2^d - L_E \vee \|w\|_\infty > \lfloor q/2 \rfloor - L_E$ **then**
15: $\quad$ $\text{counter}++$
16: $\quad$ Restart at step 4
17: **end if**
18: **return** (c', z)

Algorithm 8 qTESLA's signature verification

Require: $m, (c', z), pk = (\text{seed}_a, t)$
Ensure: $\{0, 1\}$ ▷ reject, accept

1: $c \leftarrow \text{Enc}(c')$
2: $a \leftarrow \text{GenA}(\text{seed}_a)$
3: $w \leftarrow az - tc \mod q$
4: **if** $z \in \mathcal{R}_{q,[B-L_S]} \wedge c = H([w]_M, m)$ **then**
5: $\quad$ **return** 1
6: **end if**
7: **return** 0

$z \in \mathcal{R}_{q,[B-U]}$, $\|sc\|_\infty \leq L_S$, $\|ec\|_\infty \leq L_E$, $\|[ay - ec]_L\|_\infty \leq 2^d - L_E$, and $\|ay - ec\|_\infty \leq \lfloor q/2 \rfloor - L_E$. In order for the verification algorithm to accept a signature it has to hold that: (i) $z \in \mathcal{R}_{q,[B-U]}$, which holds trivially, and (ii) $[ay]_M = [az - tc]_M$, which we argue next.

We know that

$$\begin{aligned} [az - tc]_M &= [ay + asc - asc - ec]_M & (1)\\ &= [ay - ec]_M & (2)\\ &= \frac{ay - ec - [ay - ec]_L}{2^d}. & (3) \end{aligned}$$

We know that $\| [ay - ec]_L \|_\infty < 2^d - L_E$ and $\|ay - ec\|_\infty \leq \lfloor q/2 \rfloor - L_E$. Hence, $\|ay - ec - [ay - ec]_L\|_\infty < q/2$, and thus, no wrap-around occurs. Furthermore, since $\|ec\|_\infty \leq L_E$ and $\| [ay - ec]_L \|_\infty \leq 2^d - L_E$, we know that $-ec - [ay - ec]_L = [-ec - (ay - ec)]_L$ and hence,

$$\frac{ay - ec - [ay - ec]_L}{2^d} = \frac{ay - [ay]_L}{2^d} = [ay]_M . \tag{4}$$

2.4 Implementation details of the required functions

2.4.1 Gaussian sampling

One of the advantages of qTESLA is that Gaussian sampling is only required during key generation to sample s and e (see Alg. 6). Nevertheless, certain applications might require an efficient and secure implementation of key generation and that, in particular, be protected against timing and cache attacks. In the following, we adopt the Gaussian sampler proposed in [16], which is an improvement upon the sampler proposed by Ducas *et al.* [29, Section 6].

The basic idea of the Gaussian sampler by Ducas *et al.* [29, Algorithms 10–12] is to start from a distribution that approximates the desired Gaussian distribution. From there, a high-quality Gaussian is obtained by rejection sampling guided by Bernoulli distributions $\mathcal{B}_\rho$ with parameters ρ related to the standard deviation σ of the desired Gaussian distribution. Ducas *et al.* implement those Bernoulli distributions by decomposing them into ℓ certain base distributions $(\mathcal{B}_{\rho_0}, \mathcal{B}_{\rho_1}, \ldots, \mathcal{B}_{\rho_{\ell-1}})$ where the ρ constants are precomputed to the desired accuracy, and then sampling from those base distributions to that accuracy. Even though this Bernoulli decomposition is reportedly quite efficient, its running time highly depends on the private bits. Besides that, each $\mathcal{B}_{c_\rho}$ must be sampled to the same precision as the target distribution, which is why the total amount of entropy needed to obtain one Gaussian sample is much higher than theoretically necessary, roughly $O(\ell\lambda)$ bits rather than $O(\lambda)$ for security level λ.

However, because qTESLA only needs a basic Gaussian sampler for key generation, it is possible to obtain a much simpler construction [16]. In particular, only one Bernoulli distribution $\mathcal{B}_\rho$ is needed, instead of ℓ base distributions $(\mathcal{B}_{\rho_0}, \mathcal{B}_\rho, \ldots, \mathcal{B}_{\rho_{\ell-1}})$. Thus, the bias is simply computed by $\rho = \exp(-t/2\sigma^2)$ using well-known exponentiation techniques. The value ρ is an approximation of a real number in the interval $[0, 1]$ to the desired precision. For more details, refer to [16] and [29, Section 6].

2.4.2 Deterministic random bit generation

qTESLA requires the deterministic generation of random bits to produce seeds from random pre-seed values. Specifically, the key generation algorithm requires the generation of seeds seed_a and seed_y in Step 1 (Alg. 6). This is done with the SHA-3 derived extendable output function cSHAKE. The format to call this function is given by cSHAKE($X, L,$ " ", S) for an input bit string X and a domain separator S [45] (note that the function-name bit string is left empty). The function returns a bit string of L bits as output.

2.4.3 Generation of a: GenA

In qTESLA, a polynomial a is freshly generated per secret/public keypair using a seed seed_a. This seed is then stored as part of the public key so that the signing and verification operations can regenerate a.

The approach above permits to save bandwidth since we only need κ bits to store seed_a instead of the $n\lceil \log(q) \rceil$ bits that are required to represent the full polynomial. Moreover, the use of a fresh a per keypair makes more difficult the introduction of backdoors and reduces drastically the scope of all-for-the-price-of-one attacks [10, 16].

The procedure to generate a is as follows. First, a pre-seed is obtained from the system RNG. This pre-seed is then hashed using cSHAKE to obtain seed_a, as described in Section 2.4.2. Finally, to generate a via the expansion of seed_a, we use cSHAKE [45] such that the output size is enough to fill out all the coefficients of the polynomial. Moreover, the output of cSHAKE is filtered to make sure that a belongs to the correct ring. Note that, as a precaution, we avoid exposing directly the output of the system RNG through seed_a, and use a hashed value instead.

2.4.4 Encoding function

The encoding function Enc takes the output of the hash function H and maps it to a vector with entries in $\{-1, 0, 1\}$ of length n and weight h (representing a polynomial of degree $n - 1$). In the signature generation we need to map the hash input $([v]_M, m)$ to a polynomial $c \in \mathbb{H}_{n,h} \subset \mathcal{R}_q$ (cf. line 6 and 7 of Algorithm 7). We break this up into $\text{Enc}(H([v]_M, m)) = \text{Enc}(c') = c$ to obtain smaller signatures $(c', z) \in \{0, 1\}^\kappa \times \mathcal{R}_q$.

We implement the encoding function Enc as in [1] and as depicted in Algorithm 9. The elements $r_1, ..., r_h$ are chosen randomly by a PRF, given $c' \leftarrow H([v]_M, m)$ as input. The value c_{pos} is the (pos)-th element of the vector $c \in \mathbb{H}_{n,h}$, which is initialized as a zero vector. This algorithm is an extension of an algorithm originally proposed in [32, Section 4.4] which in turn relies on [29].

Algorithm 9 Encoding function Enc

Require: $c' \in \{0,1\}^\kappa$
Ensure: $c \in \mathbb{H}_{n,h}$

$r_1, ..., r_{h-1}, r_h \leftarrow \mathrm{PRF}(c')$
for $i = 1, ..., h$: **do**
    $pos \leftarrow (r_i \ll 8) \vee (r_{i+1})$
    if $r_{i+2} \mod 2 = 1$ **then**
        $c_{pos} \leftarrow -1$
    else
        $c_{pos} \leftarrow 1$
    end if
end for
return c

2.4.5 Hash and pseudo-random functions

qTESLA's signing procedure requires the hash function H as well as the pseudo-random functions PRF_1 and PRF_2. We adopt SHA-3 [33] for function H, and cSHAKE [45] for functions PRF_1 and PRF_2.

PRF_1 takes as input the seed seed_y and the message m and maps it to a byte array, i.e., $\mathrm{PRF}_1 : \{0,1\}^\kappa \times \{0,1\}^* \rightarrow \{0,1\}^\kappa$ (cf. line 3 of Algorithm 7). To do this we use the output of cSHAKE.

PRF_2 takes as input the values rand and counter and maps them to a ring element, i.e., $\mathrm{PRF}_2 : \{0,1\}^\kappa \times \mathbb{Z} \rightarrow \mathcal{R}_{q,[B]}$ (cf. line 4 of Algorithm 7). To do this we use the output of cSHAKE and split it into n chunks representing the coefficients of the polynomial y in $\mathcal{R}_{q,[B]}$.

It is worth noting that we take the hash output size κ to be larger or equal to the security level λ. This is consistent with the use of the hash in a Fiat-Shamir style signature scheme such as qTESLA. In the Fiat-Shamir paradigm for signatures, preimage resistance is relevant while collision resistance is much less, given that we take the hash size to be enough to resist preimage attacks[1].

[1]We chose the hash size aiming for security of Category 5, according to NIST's categories of security

2.5 System parameters and parameter selection

In this section, we describe qTESLA's system parameters and our choice of parameter sets. We summarize all bounds and our concrete parameter sets in Table 1. We explain how we estimate the bit security of our signature scheme in Section 5.2.

Herein, we propose three parameter sets that we classify according to NIST's categories of security as follows:

qTESLA-128:	NIST's security category 1,
qTESLA-192:	NIST's security category 3,
qTESLA-256:	NIST's security category 5.

Our parameters are chosen according to the security reduction provided in Theorem 6, Section 5.1. This implies the following: suppose that parameters are constructed for a certain security level. By virtue of our security reduction these parameters correspond to an instance of the R-LWE problem. Since our parameters are chosen according to the provided security reduction, this reduction provably guarantees that our scheme has the selected security level as long as the corresponding R-LWE instance is intractable. In other words, hardness statements for R-LWE instances have a provable consequence for the security levels of our scheme.

Since the presented reduction is tight, the tightness gap of our reduction is equal to 1 for our choice of parameters and, hence, the concrete bit security of our signature scheme is essentially the same as the bit hardness of the underlying R-LWE instance. We make our sage script used to choose parameters available. It is called `parameterchoice.sage` and can be found in the submission folder "Script_to_choose_parameters".

Let λ be the security parameter, i.e., the targeted bit security of the instantiation is λ. Let $n \in \mathbb{Z}_{>0}$ be the dimension, i.e., $n-1$ is the polynomial degree. To use efficient polynomial multiplication, i.e., the number theoretic transform (NTT) in the ring $\mathcal{R}_q$, we restrict ourselves to a polynomial degree of a power of two, i.e, $n = 2^l$ for $l \in \mathbb{N}$. Let σ be the standard deviation of the centered discrete Gaussian distribution that is used to sample the coefficients of the secret and error polynomials. To use the fast Gaussian sampler as described in Section 2.4.1, we choose $\sigma = \frac{\xi}{\sqrt{2 \ln 2}}$ for some $\xi \in \mathbb{Z}_{>0}$. The parameter κ defines the output (resp., input) length of random functions described in Section 2.4.5. The parameter h defines the encoding function described in Section 2.4.4. More concretely, it defines the number of non-zero elements of the output of the encoding function.

The values L_E and L_S are used to bound the coefficients in the error and secret polynomials

for preimage resistance. In a scenario that excludes Groover'a algorithm a hash function with an output length of λ is expected to have preimage resistance of 2^{λ}. When considering the quadratic acceleration of Groover's algorithm, the preimage resistance is only $\approx 2^{\lambda/2}$. In such a case, the hash output length should be 2λ for an aspired security level of λ.

Table 1: Description and bounds of the parameters according to the tight security reduction in the quantum random oracle model with $q_h = 2^{128}$ and $q_s = 2^{64}$; we choose $M = 0.3$; we write parameters used in the implementation in **bold**

Param.	Description	Requirement	qTesla-128	qTesla-192	qTesla-256
λ $\mathbf{n}$	security parameter dimension ($n-1$ is the poly. degree)	- power-of-two	128 **1 024**	192 **2 048**	256 **2 048**
σ, ξ	standard deviation of centered discrete Gaussian distribution	$\sigma = \frac{\xi}{\sqrt{2\ln 2}}$	**8.5**, 10		
$\mathbf{q}$	modulus	$q = 1 \mod 2n$, $q^n \geq \|\Delta\mathbb{S}\| \cdot \|\Delta\mathbb{L}\| \cdot \|\Delta\mathbb{H}\|$, $q^n \geq 2^{4\lambda+n(d+1)} 3q_s^3(q_s+q_h)^2$	**8 058 881** $\leq 2^{23}$	**12 681 217** $\leq 2^{24}$	**27 627 521** $\leq 2^{25}$
$\mathbf{h}$	# of non-zero entries of output elements of Enc	$2^h \cdot \binom{n}{h} \geq 2^{2\lambda}$	**36**	**50**	**72**
κ	output length hash function H and input length GenA, PRF_1, PRF_2, Enc	$\kappa \geq \lambda$	256		
$\mathbf{L_E}$, η_E $\mathbf{L_S}$, η_S	bound in *checkE* bound in *checkS*	$\eta_E \cdot h \cdot \sigma$ $\eta_S \cdot h \cdot \sigma$	**798**, 2.48 **758**, 2.61	**1 117**, 2.68 **1 138**, 2.63	**1 534**, 2.48 **1 516**, 2.51
$\mathbf{B}$	determines the interval the randomness is chosen in during sign	$B \geq \frac{\sqrt[n]{M}+2L_S-1}{2(1-\sqrt[n]{M})}$, near to power-of-two	$\mathbf{2^{20}-1}$	$\mathbf{2^{21}-1}$	$\mathbf{2^{22}-1}$
$\mathbf{d}$	number of rounded bits	$\left(1-\frac{2\cdot L_E+1}{2^d}\right)^n \geq 0.3$, $d > \log_2(B)$	**21**	**22**	**23**
$\|\Delta\mathbb{H}\|$ $\|\Delta\mathbb{S}\|$ $\|\Delta\mathbb{L}\|$	see definition below in the text	$\sum_{j=0}^{h}\sum_{i=0}^{h-j}\binom{n'}{2i}2^{2i}\binom{n'-2i}{j}2^j$ $(4(B-L_S)+1)^n$ $(2^{d+1}+1)$	$\approx 2^{447}$ $\approx 2^{22526}$ $2^{22}+1$	$\approx 2^{675}$ $\approx 2^{47102}$ $2^{23}+1$	$\approx 2^{898}$ $\approx 2^{49150}$ $2^{24}+1$
δ_w δ_z	acc. prob. of w in line 19 during sign acc. prob. z in line 19 during sign	experimentally experimentally	0.50 0.50	0.33 0.25	0.33 1.00
δ_{keygen}	acc. prob. of key pairs	experimentally	1.00		
sig size pk size sk size	theoretical size signature [byte] theoretical size public key [byte] theoretical size secret key [byte]	$\kappa + n(\lceil\log_2(B-L_S)\rceil+1)$ $n(\lceil\log_2(q)\rceil)+\kappa$ $2n(\lceil\log_2(t\cdot\sigma+1)\rceil)+2\kappa$ with $t = 13.4$, 16.4, or 18.9	2 720 2 976 1 856	5 664 6 176 4 160	5 920 6 432 4 128

during *checkE* and *checkS*, respectively. However, since the rejection probability of key pairs during the key generation is close to zero for our parameter sets (as determined experimentally) the key space is not restricted noticeably. Both bounds, L_E and L_S, impact the rejection probability during the signature generation, as follows. Larger the values of L_E and L_S will increase the acceptance probability during the key generation. But they will also decrease acceptance probability in the signature generation line 14 and line 9, respectively. We determine the best trade-off between those two acceptance probabilities experimentally. We start choosing $L_E = \eta_E \cdot h \cdot \sigma$ (resp., $L_S = \eta_S \cdot h \cdot \sigma$) with $\eta_E = \eta_S = 2.8$

and try different values for $\eta_E, \eta_S \in [2.0, 3.0]$. Let $M = 0.3$ be a value of our choosing that determines (together with L_S and B) the acceptance probability of the rejection sampling in line 9 Algorithm 7. The parameter B defines the interval of the random polynomial y (cf. line 4 of Algorithm 7) and it is determined by M and the parameter L_S as follows:

$$\left(\frac{2B - 2L_S + 1}{2B + 1}\right)^n \geq M \Leftrightarrow B \geq \frac{\sqrt[n]{M} + 2L_S - 1}{2(1 - \sqrt[n]{M})}.$$

We select the rounding value d to be larger than $\log_2(B)$ and such that the acceptance probability of the check $\| [w]_L \|_\infty > 2^d - L_E$ in Line 14 of Algorithm 7 is upper bounded by 0.7 when using the sage script to choose parameters. Changing the value L_E as described above, impacts the rejection probability of w as well. We determine the acceptance probability δ_z of z and δ_w of w during sign and the acceptance probability of key pairs δ_{keygen} experimentally and summarize the result in Table 1.

The parameter q is chosen to fulfill several bounds and assumptions that are motivated by the security reduction or efficient implementation requirements. To simplify our statement in the security reduction we ensure that $q^n \geq |\Delta\mathbb{S}| \cdot |\Delta\mathbb{L}| \cdot |\Delta\mathbb{H}|$ with the following definition of sets: $\mathbb{S}$ is the set of polynomials $z \in \mathcal{R}_{q,[B-L_S]}$ and $\Delta\mathbb{S} = \{z - z' \ : \ z, z' \in \mathbb{S}\}$, $\mathbb{H}$ is the set of polynomials $c \in \mathcal{R}_{q,[1]}$ with exactly h nonzero coefficients and $\Delta\mathbb{H} = \{c - c' \ : \ c, c' \in \mathbb{H}\}$, and $\Delta\mathbb{L} = \{x - x' : x, x' \in \mathcal{R}$ and $[x]_M = [x']_M \in \mathcal{R}_{q,[2^d-1]}\}$. To choose parameters according to the security reduction the following equation (cf. Theorem 6) has to hold:

$$\frac{2^{3\lambda + n(d+1)} \cdot 3 \cdot q_s^3 (q_s + q_h)^2}{q^n} \leq 2^{-\lambda} \Leftrightarrow q \geq \left(2^{4\lambda + n(d+1)} \cdot 3 \cdot q_s^3 (q_s + q_h)^2\right)^{1/n}.$$

To be able to use fast polynomial multiplication we choose q to be a prime integer such that $q \mod 2n = 1$.

As stated in the NIST call for proposals (Section 4.A.4), we choose the number of classical queries to the sign oracle to be $q_s = 2^{64}$ for all our parameter sets. Moreover, we choose the number of queries of a hash function to be $q_h = 2^{128}$.

Key and signature sizes Given all parameters as explained above, we determine the key and signature sizes as follows. The theoretical length of the signature in bits is given by $\kappa + n \cdot (\lceil \log_2(B - L_S) \rceil + 1)$ and the public key is represented by $n \cdot (\lceil \log_2(q) \rceil) + \kappa$ bits. To determine the size of the secret key we note that for $t > 0$ it holds that $Pr_{x \leftarrow_\sigma \mathbb{Z}} [|x| > t\sigma] \leq 2e^{-t^2/2}$. For example for $t = 13.4$, $t = 16.4$, and $t = 18.9$ the probability $Pr_{x \leftarrow_\sigma \mathbb{Z}} [|x| > t\sigma]$ is less or equal 2^{-128}, 2^{-192}, and 2^{-256}, respectively. Therefore, the theoretical size of the secret key is given by $n \cdot (\lceil \log_2(14\sigma + 1) \rceil) + n \cdot (\lceil \log_2(t \cdot \sigma + 1) \rceil) + 2\kappa$ bits with $t = 13.4$, $t = 16.4$, and $t = 18.9$ for qTesla-128, qTesla-192, and qTesla-256, respectively.

Table 2: Different key and signature sizes of our proposed parameter sets; we abbreviate theoretical sizes with TS and sizes as used in the implementations with IS; sizes are given in bytes.

Parameter set	TS/IS	public key	secret key	signature
qTesla-128	TS	2 976	1 856	2 720
	IS	4 128	2 112	3 104
qTesla-192	TS	6 176	4 160	5 664
	IS	8 224	8 256	6 176
qTesla-256	TS	6 432	4 128	5 920
	IS	8 224	8 256	6 176

We determined the key and signature sizes in our reference implementation as smallest suitable data type which can hold $max((\lceil\log_2(14\sigma+1)\rceil),(\lceil\log_2(t\cdot\sigma+1)\rceil))$, which is byte for qTesla-128, and 16 bit integer for qTesla-192, and qTesla-256. Table 2 shows key and signature sizes according to the theoretical sizes and sizes as in the implementations for our three proposed parameter sets in comparison.

3 Performance analysis

The submission package includes a simple yet efficient reference implementation written exclusively in C.

To evaluate the performance of the provided implementation, we ran our benchmarking suite on a machine powered by a 2.40 GHz Intel Core i5-6300U (Skylake) processor, running Ubuntu 16.04.3 LTS. As is standard practice, TurboBoost was disabled during the tests. For compilation we used clang version 3.8.0 with the command `clang -O3`. See Table 3 for the results.

Scheme	`keygen`	`sign`	`verify`	**total** (`sign + verify`)
`qTESLA-128`	3 402	2 495	520	3 015
`qTESLA-192`	5 875	9 686	1 065	10 751
`qTESLA-256`	12 433	26 063	1 310	38 496

Table 3: Performance (in thousands of cycles) of qTESLA on a 2.40 GHz Intel Core i5-6300U (Skylake) processor. Cycle counts are rounded to the nearest 10^3 cycles.

The results in Table 3 correspond to a relatively simple implementation of qTESLA. Nevertheless, they demonstrate that the scheme is practical for most applications. We expect significant improvements in the future with a fully optimized implementation.

4 Known answer values

The submission includes KAT values with tuples that contain message size (`mlen`), message (`msg`), public key (`pk`), secret key (`sk`), signature size (`smlen`) and signature (`sm`) values for all the proposed security levels. The KAT files can be found in the media folder: `\KAT\PQCsignKAT_qTesla-128.rsp`, `\KAT\PQCsignKAT_qTesla-192.rsp`, and `\KAT\PQCsignKAT_qTesla-256.rsp` for qTESLA-128, qTESLA-192 and qTESLA-256, respectively.

5 Expected security strength

It this section we discuss the expected security strength of and possible attacks against qTESLA. This includes two statements about the theoretical security and the parameter choices depending on them. To this end we first define the hardness assumptions qTESLA is based on.

We define the ring short integer solution problem (R-SIS) similar to [30].

Definition 2 (Ring short integer solution problem $R-SIS_{n,k,q,\beta}$)**.** *Given* $a_1, ..., a_k \leftarrow_{\$} \mathcal{R}_q$*. Then the ring short integer solution problem* $R-SIS_{n,k,q,\beta}$ *is to find solutions* $u_1, ..., u_k, u_{k+1} \in \mathcal{R}_q$*,* $u_i \neq 0$ *for at least one* i*, such that* $(a_1, ..., a_k, 1) \cdot (u_1, ..., u_{k+1})^T = a_1u_1 + ... + a_ku_k + u_{k+1} = 0 \mod q$ *and* $\|u_1\|, ..., \|u_{k+1}\| \leq \beta$*.*

We define the learning with errors distribution and the ring learning with errors problem (LWE) in the following.

Definition 3 (Learning with Errors Distribution)**.** *Let* $n, q > 0$ *be integers,* $s \in \mathcal{R}$*, and* χ *be a distribution over* $\mathcal{R}$*. We define by* $\mathcal{D}_{s,\chi}$ *the LWE distribution which outputs* $(a, \langle a, s\rangle + e) \in \mathcal{R}_q \times \mathcal{R}_q$*, where* $a \leftarrow_{\$} \mathcal{R}_q$ *and* $e \leftarrow \chi$*.*

Since our signature scheme is based on the decisional learning with errors problem, we omit the definition of the search version and state only the decisional learning with errors problem.

Definition 4 (Ring Learning with Errors Problem $R-LWE_{n,m,q,\chi}$)**.** *Let* $n, q > 0$ *be integers and* χ *be a distribution over* $\mathcal{R}$*. Moreover, let* $s \in \mathcal{R}$ *and* $\mathcal{D}_{s,\chi}$ *be the learning with errors distribution. Given* m *tuples* $(a_1, t_1), ..., (a_m, t_m)$*, the decisional ring learning with errors problem* $R-LWE_{n,m,q,\chi}$ *is to distinguish whether* $(a_i, t_i) \leftarrow \mathcal{U}(\mathcal{R}_q \times \mathcal{R}_q)$ *or* $(a_i, t_i) \leftarrow \mathcal{D}_{s,\chi}$ *for all* i*.*

5.1 Provable security in the (quantum) random oracle model

The security of our scheme qTESLA is supported by two statements reducing the hardness of lattice-based assumptions to the security of our proposed signature scheme in the (quantum) random oracle model. In this subsection we give the two statements but we do not give formal security proofs since they are very close to the original results as explained below.

The first reduction (cf. Theorem 5) follows the approach proposed by Bai and Galbraith [14] closely and gives a non-tight reduction from R-LWE and R-SIS to the existentially unforgeability under chosen-message attack (EUF-CMA) of qTESLA in the random oracle model.

Theorem 5. *Let $2^n \cdot \binom{n}{h} \geq 2^\lambda$, $(2R+1)^2 \geq q^n 2^\kappa$, and $q > 4B$. If there exists an adversary A that forges a signature of the signature scheme $qTESLA$ described in Section 2.2 in time t_Σ and with success probability ϵ_Σ, then there exists a reduction R that solves either*

- *the $R-LWE_{n,m,q,\sigma}$ with $m = 1$ problem in time $t_{LWE} \approx t_\Sigma$ with $\epsilon_{LWE} \geq \epsilon_\Sigma/2$, or*
- *the $R-SIS_{n,k,q,\beta}$ problem with $\beta = \max\{k2^{d-1}, 2(B-U)\} + 2hR$ in time $t_{SIS} \approx 2t_\Sigma$ with $\epsilon_{SIS} \geq \frac{1}{2}(\epsilon_\Sigma - \frac{1}{2^\kappa})\left(\frac{(\epsilon_\Sigma - \frac{1}{2^\kappa})}{q_h} - \frac{1}{2^\kappa}\right) + \epsilon_\Sigma/2$ with our choice of parameters.*

The second security reduction (cf. Theorem 6) gives a tight reduction in the *quantum* random oracle model from R-LWE to EUF-CMA of qTESLA. In our opinion the second theorem is much stronger since it shows security against adversaries that have quantum access to a quantum random oracle and we will therefore always refer to Theorem 6 when we talk about the security of the scheme. We emphasize that Theorem 6 gives a reduction from the decisional ring learning with errors problem where in Theorem 5 also the decisional ring SIS problem is used. Currently, Theorem 6 holds assuming a conjecture as stated and explained below.

Theorem 6. *Let the parameters be as in Table 1. Furthermore, assume that Conjecture 7 holds. If there exists an adversary A that forges a signature of the signature scheme $qTESLA$ described in Section 2.2 in time t_Σ and with success probability ϵ_Σ, then there exists a reduction R that solves the $R-LWE_{n,m,q,\sigma}$ problem with $m = 1$ in time $t_{LWE} \approx t_\Sigma$ with $\epsilon_\Sigma \leq \frac{2^{3\lambda+(d+1)} \cdot 3 \cdot q_s^3 (q_s+q_h)^2}{q} + \frac{2q_h+5}{2^\lambda} + \epsilon_{LWE}$ with our choice of parameters.*

The proof follows the approach proposed in [9] except for the computation of the two probabilities $\mathrm{coll}(a, e)$ and $\mathrm{nwr}(a, e)$ that we explain in the following. For simplicity we assume that the randomness is sampled uniformly random in $\mathcal{R}_{q,[B]}$ as in Algorithm 2. We define $\Delta\mathbb{L}$ to be the set $\{x - x' : x, x' \in \mathcal{R}$ and $[x]_M = [x']_M \in \mathcal{R}_{q,[2^d-1]}\}$. Furthermore, we call a polynomial w *well-rounded* if w is in $\mathcal{R}_{q,[\lfloor q/2 \rfloor - L]}$ and $[w] \in \mathcal{R}_{q,[(2^d-L)]}$. We define the following quantities for keys (a, t), (s, e)

$$\text{nwr}(a, e) \overset{\text{def}}{=} \Pr_{(y,c)\in\mathbb{Y}\times\mathbb{H}} [ay - ec \text{ not well-rounded }] \tag{5}$$

$$\text{coll}(a, e) \overset{\text{def}}{=} \max_{(w)\in\mathbb{W}} \left\{ \Pr_{(y,c)\in\mathbb{Y}\times\mathbb{H}} [[ay - ec]_M = w] \right\}. \tag{6}$$

Informally speaking $\text{nwr}(a, e)$ refers to the probability over random (y, c) that $ay - ec$ is not well-rounded. This quantity varies as a function of a, e. In contrast to [9], we cannot upper bound this in general in the ring setting. Hence, we first assume that $\text{nwr}(a, e) < \frac{2}{3}$ and afterwards check experimentally that this holds true. As our acceptance probability of w in line 19 of Algorithm 7 (signature generation) is at least 0.34 for all parameter sets (cf. δ_w in Table 1), the bound $\text{nwr}(a, e) < \frac{2}{3}$ holds.

Secondly, we need to bound the probability $\text{coll}(a, e)$. In [9, Lemma 4] the corresponding probability $\text{coll}(A, E)$ for standard lattices is upper bounded. Unfortunately, we were not able to transfer the proof to the ring setting for the following reason. In the proof of [9, Lemma 4], it is used that if the randomness y is not equal to 0 the vector Ay is uniformly random distributed over $\mathbb{Z}_q$ and hence also $Ay - Ec$ is uniformly random distributed over $\mathbb{Z}_q$. This does not necessarily hold if the *polynomial* y is chosen uniformly in $\mathcal{R}_{q,[B]}$. Moreover, in Equation (99) in [9], ψ denotes the probability that a random vector $x \in \mathbb{Z}_q^m$ is in $\Delta\mathbb{L}$:

$$\psi \overset{\text{def}}{=} \Pr_{x\in\mathbb{Z}_q^m} [x \in \Delta\mathbb{L}] \leq \left(\frac{2^{d+1}}{q} \right)^m . \tag{7}$$

The quantity ψ is a function of the TESLA parameters q, m, d. It is negligibly small.

We cannot prove a similar statement for the signature scheme qTESLA over ideals. Instead, we need to *conjecture* the following.

Conjecture 7. *Let I be a non-zero ideal in $\mathcal{R}_q$ and let $r \in \mathcal{R}_q$ be a fixed choice of ring elements. Then it holds that the probability over a uniformly distributed element $x \leftarrow_{\$} I$ that $x + r \in \Delta\mathbb{L}$ is negligibly small.*

The intuition behind our conjecture is as follows. Let ψ_I denote the probability that a random element from the ideal I lands in $\Delta\mathbb{L}$. We know that ψ_I is small when the ideal $I = \mathcal{R}_q$, i.e., a negligibly small fraction of elements from $\mathcal{R}_q$ are in $\Delta\mathbb{L}$. Furthermore, the set $\Delta\mathbb{L}$ appears to have no relationship with the ideal structure of the ring, so it seems reasonable to view each ideal as a "random" subset of $\mathcal{R}_q$ in the following sense: No larger or smaller portion of elements in the ideal I is in $\Delta\mathbb{L}$ than that portion of elements of $\mathcal{R}_q$ that is in $\Delta\mathbb{L}$.

Hence, the corresponding statement described above and needed in [9, Lemma 4] translates for qTESLA to the following. If $y \neq 0$ then ay is a uniformly random element of some non-zero ideal I. The polynomial c is fixed and the polynomial e is independent of the polynomial a, and y. Hence, by our conjecture (with $x = ay$ and $r = ec$) it holds that the probability of Equation (107) in [9] is negligibly small. Thus, assuming that our conjecture holds true, [9, Lemma 4] and hence the security reduction in [9] holds for qTESLA as well.

5.2 Bit security of our proposed parameter sets

In the following we describe how we estimate the concrete security of our proposed parameters. To this end, we first describe how the security of our scheme depends on the hardness of R-LWE and afterwards we describe how we derive the bit hardness of the underlying R-LWE instance. We classify our three parameter sets according to NIST's categories of security in Section 2.5.

5.2.1 Correspondence between security and hardness

The security reduction given in Section 5.1, Theorem 6 provides a reduction from the hardness of the decisional ring learning with errors problem and bounds *explicitly* the forging probability with the success probability of the reduction. More formally, let ϵ_Σ and t_Σ denote the success probability and the run time of a forger against our signature scheme and let ϵ_{LWE} and t_{LWE} denote analogous quantities for the reduction presented in the proof of Theorem 6. We say that R-LWE is *η-bit hard* if $t_{LWE}/\epsilon_{LWE} \geq 2^\eta$; and we say that the signature scheme is *λ-bit secure* if $t_\Sigma/\epsilon_\Sigma \geq 2^\lambda$.

Since we choose parameters such that $\epsilon_{LWE} \approx \epsilon_\Sigma$ and $t_\Sigma \approx t_{LWE}$, the bit hardness of the R-LWE instance is the same as the bit security of our signature scheme.

5.2.2 Estimation of the hardness of R-LWE

Since the introduction of the learning with errors problem over rings [52], it is an open question whether the R-LWE problem is as hard as the LWE problem. Several results exist that exploit the ideal structure of some ideal lattices [23, 26, 35, 37]. However, up to now, these results are not known to be applicable to R-LWE. In particular, the found weaknesses do not apply to our instances. Consequently, we estimate the hardness of R-LWE using state-of-the-art attacks against LWE.

Albrecht, Player, and Scott [8] presented the *LWE-Estimator*, a software to estimate the hardness of LWE given the matrix dimension n, the modulus q, the relative error rate $\alpha = \frac{\sqrt{2\pi}\sigma}{q}$, and the number of given LWE samples. The LWE-Estimator estimates the hardness against the fastest LWE solvers currently known, i.e., it outputs an upper (conservative) bound on the number of operations an attack needs to break a given LWE instance.

In particular, the following attacks are considered in the *LWE-Estimator*: The meet-in-the-middle exhaustive search, the coded Blum-Kalai-Wassermann algorithm [42], the dual lattice recently published [3], the enumeration approach by Linder and Peikert [49], the primal attack described in [6,15], and the Arora-Ge algorithm [11] using Gröbner bases [4]. Moreover, the latest analysis to compute the block sizes used in the lattice basis reduction BKZ published recently by Albrecht *et al.* [2] are implemented.

Furthermore, quantum speed-ups for the sieving algorithm used in BKZ [47,48] are considered. Another recent quantum attack, called quantum hybrid attack, by Göpfert, van Vredendaal, and Wunderer [40] is not considered in our analysis (and the *LWE-Estimator*). The hybrid attack is most efficient on the learning with errors problem with very small secret and error, e.g., binary or ternary. Since the coefficients of the secret and error of qTESLA are chosen Gaussian distributed, the attack is not efficiently applicable on our instances.

The *LWE-Estimator* is the result of many different contributions and contributors. It is open source and hence easily checked and maintained by the community. Hence, we find the *LWE-Estimator* to be a suitable tool to estimate the hardness of our chosen LWE instances. We integrated the LWE-Estimator with commit-id `9302d42` on 2017-09-27 in our sage script.

In the following we describe very briefly the most efficient LWE solvers for our instances, i.e., the decoding attack and the embedding approach, following closely the description of [18]. The Blum-Kalai-Wasserman algorithm [5, 46] is omitted since it requires exponentially many samples.

The embedding attack. The standard embedding attack solves LWE via reduction to the unique shortest vector problem (uSVP). During the reduction an $m+1$-dimensional lattice that contains the error vector e is created. Since e is very short for typical LWE instances, this results in a uSVP instance that is usually solved by applying basis reduction.

Let $(A, c = As + e \mod q)$ and t be the distance $\mathrm{dist}(c, L(A)) = \|c - x\|$ where $x \in L(A)$, such that $\|c-x\|$ is minimized. Then the lattice $L(A)$ can be embedded in the lattice $L(A')$, with $A' = \begin{pmatrix} A & c \\ 0 & t \end{pmatrix}$. If $t < \frac{\lambda_1(L(A))}{2\gamma}$, the higher-dimensional lattice $L(A')$ has a unique shortest vector $c' = (-e, t) \in Z_q^{m+1}$ with length $\|c'\| = \sqrt{m\alpha^2 q^2/(2\pi) + |t|^2}$ [27,51]. In the LWE-Estimator $t = 1$ is used. Therefore, e can be extracted from c', As is known, and s can be solved for. Based on Albrecht *et al.* [7], Göpfert shows [39, Section 3.1.3] that the standard embedding attack succeeds with non-negligible probability if $\delta_0 \leq \left(\frac{q^{1-\frac{n}{m}}\sqrt{\frac{1}{e}}}{\tau\alpha q}\right)^{\frac{1}{m}}$, where m is the number of LWE samples. The value τ is experimentally determined to be $\tau \leq 0.4$ for a success probability of $\epsilon = 0.1$ [7].

The efficiency of the embedding attack highly depends on the number of samples. In case of LWE instances with limited number of samples, the lattice $\Lambda_q^{\perp}(A_o) = \{v \in \mathbb{Z}^{m+n+1} | A_o \cdot v = 0 \bmod q\}$ with $A_o = [A|I|b]$ can be used as the embedding lattice.

The decoding attack. The decoding attack treats an LWE instance as an instance of the bounded distance decoding problem (BDD). The attack can be divided into two phases: Basis reduction and finding closest vector to target vector. In the first phase, basis reduction algorithms like BKZ [55] are applied. Afterwards, in the second phase, the nearest plane algorithm [13] (or variants) are applied to find the closest vector to As and thereby eliminate the error vector e of the LWE instance. Now, the secret can be accessed, as the closest vector equals an LWE instance's As.

5.3 Resistance to implementation attacks

Recently, the scheme ring-TESLA [1] was analyzed with respect to cache side channels with the software tool CacheAudit [20]. It was the first time that a post-quantum scheme was analyzed with program analysis. The authors found potential cache side channels, proposed countermeasures, and showed the effectiveness of their mitigations with CacheAudit. Since the implementation of ring-TESLA is similar to our implementation of qTESLA, we implemented all countermeasures proposed in [20] to secure our scheme against bit leakage via cache side channels.

The implementation of ring-TESLA was also analyzed regarding fault attacks [19, 36] and it was found that ring-TESLA is vulnerable to fewer fault attacks then, e.g., the signature scheme BLISS [29]. Due to the similarities of the implementations of ring-TESLA and qTESLA, the results from [19] are transferable to qTESLA. Another possible fault attack is described in Section 5.4.

5.4 Deterministic vs. probabilistic signature scheme

The following discussion is about how to generate the randomness y in Algorithm 7, line 4-6, and how different approaches prevent or enable different attacks.

In the current description in Algorithm 7, signatures are generated deterministically, i.e., for the same message always the same signature is generated. To this end an additional secret seed_y is part of the secret key. The value seed_y is used to generate a randomness rand and afterwards, rand is used to generate the polynomial y. The advantage of this approach is that a different randomness is used for different messages with very high probability. Hence, attacks that exploit a fixed randomness, such as done for Sony's playstation 3 [22], are prevented. Another advantage is that no access to a source of high-quality randomness is needed.

Our approach, however, might open a vulnerability to a fault attack proposed in [53] and briefly described in the following: Assume a signature (z, c) is generated for message m. Afterwards, a signature for the same message m is asked again. However, during the generation of the second signature a fault is injected on the hash value c yielding the value c_{faulted}, hence the second signature is $(z_{\text{faulted}}, c_{\text{faulted}})$. Computing $z - z_{\text{faulted}} = sc - sc_{\text{faulted}} = s(c - c_{\text{faulted}})$, gives the s since $c - c_{\text{faulted}}$ is known to the attacker. The authors of [53] argue that the attack is rather realistic and that it is applicable to all deterministic *Schnorr-like* signatures. To prevent the fault attack but to still get new randomness for every message one could use *weak* randomness as input for the PRF. For example, instead of using the same seed_y from the secret key, $\text{seed}_y \leftarrow_{\$} \{0,1\}^{\kappa}$ could be sampled freshly every time. This would yield again a probabilistic signature scheme. Hence, we decided to stick to our proposal. Furthermore, in [53] the attack is only described against ECDSA and EdDSA signatures. Due to the rejection sampling and other correctness checks during the signature generation, this fault attack might not be as successful on our signature scheme as it is on ECDSA and EdDSA signatures.

6 Advantages and limitations

In this section we summarize the advantages and limitations of our proposed signature scheme qTESLA. Within that we compare our scheme with other post-quantum and classical signatures.

Security of our signature scheme. Our signature scheme is provably EUF-CMA secure: a security reduction from the hardness of the decisional ring learning with errors problem to EUF-CMA security of our scheme is given. Our security reduction (cf. Theorem 6) is given in the quantum random oracle model, i.e., a quantum adversary is allowed to ask the random oracle in super position. Our security reduction is based on a variant of our scheme over standard lattices [9]. To port the reduction given in [9], we use a heuristic argument as explained in Section 5.1. Our security reduction is explicit, i.e., we can explicitly give the relation between the success probabilities of solving the R-LWE problem and to forge signatures of qTESLA. Our security reduction is tight which is a desirable property because when choosing the scheme's parameters according to security reductions, tight reductions lead to smaller parameters and hence better performance.

Choice of parameters. Parameters can be chosen either heuristically or according to existing security reductions. The heuristic approach identifies the security level of an instantiation of a scheme by a certain parameter set with the hardness level of the instance of the underlying lattice problem that corresponds to these parameters regardless of the tightness gap of the provided security reduction. The parameter choice according to a reduction can be considered as a more convincing security argument since it provably guarantees that our scheme has the selected security level as long as the corresponding R-LWE instance is intractable. Our three parameter sets are chosen regarding our given quantum security reduction.

The security of our proposed parameter sets are estimated against known state-of-the-art classical and quantum algorithms to solve the learning with errors problem. Furthermore, our parameters are chosen with a comfortable gap between the targeted and the estimated bit security they provide such that they might be secure against improved or unknown LWE solvers as well. Moreover, our choice of parameters is easy comprehensible: All relations between the parameters are explained and we make our sage script used to choose parameters available[2]. Hence, if more parameter sets are needed they can be chosen easily.

Ease of Implementation. qTESLA has a very compact structure consisting of a few, ease-to-implement functions. Moreover, in contrast to popular R-LWE based schemes, qTESLA does not enforce the use of the number theoretic transform (NTT), i.e., its use is optional and the scheme remains fully compatible with an implementation that uses a straightforward schoolbook polynomial multiplication. This design decision enables the possibility of even simpler implementations. Another advantage of qTESLA is that Gaussian sampling is only required during key generation. Even if the fast Gaussian sampler included in this document is not used, most applications will not be impacted by the use of a slower Gaussian sampler.

Implementation attacks. We protect the signature generation against cache side channels by implementing the countermeasures proposed in [20]. Furthermore, the predecessor of our proposed scheme was already analyzed with respect to fault attacks [19, 36].

Applicability of our scheme. Our proposal is a good candidate to be integrated to hybrid signature schemes easing the transition from classical to post-quantum cryptography. The key sizes of all three parameter sets are small enough to be used in hybrid signature schemes [21]. Following [21] it should be appropriate to be used in X.509 standard version 3 [25], to be used in TLSv1.2 [28] for most browsers and libraries tested in [21], and to

[2] It is called `parameterchoice.sage` and can be found in the submission folder "Script_to_choose_parameters".

be used in the Cryptographic Message Syntax (CMS) [43] that is the main cryptographic component of S/MIME [54].

Comparison with selected state-of-the-art signature schemes. In the following we give a comparison of the key and signature sizes with selected classical and post-quantum signature schemes. We do not compare qTESLA with other post-quantum signatures regarding the running time because cycle counts, in particular for lattice-based signature schemes, are usually given for optimized implementations that utilize fast AVX2 arithmetic. Such optimizations, however, are not requested by NIST. A comparison of cycle counts obtained from different platforms might be misleading.

Table 4 summarizes the key and signature sizes of selected signature schemes. Moreover, it also states the underlying computational assumptions although not all construction do rely *provably* on the corresponding hardness assumption. Furthermore, only few of the parameters in the table are chosen according to provided security reductions and the bit security of the parameters are not always estimated against classical and quantum adversaries. We distinguish the different was to choose parameters in the table.

As can be seen in Table 4, qTESLA is among the post-quantum schemes with the smallest signature size if parameters are chosen with regard to quantum algorithms. In particular, the signature size of qTESLA is several magnitudes smaller than hash-based and multivariate signatures. Only the lattice-based scheme BLISS has noticeably smaller signatures. The parameters proposed for BLISS, however, are not chosen with state-of-the-art methods, not according to the provided security reduction, and the bit security is not estimated against quantum adversaries.

In comparison with the classical signature schemes RSA and ECDSA for the same security level, qTESLA has larger signature sizes. However, qTESLA is comparable with RSA-3072 in view of secret key size.

Table 4: Overview of selected state-of-the-art post-quantum and classical signature schemes; signature and key sizes are given in byte [B]; we write "–" if no corresponding data is available

Software/ Scheme	Comp. Assum.	Bit Security	Key Size [B]	Sig. Size [B]
Selected lattice-based signatures schemes				
qTESLA qTesla-128[a] (this document)	R-LWE	128[b]	pk: 2 976 sk: 1 856	2 720
qTESLA qTesla-192[a] (this document)	R-LWE	192[b]	pk: 6 176 sk: 4 160	5 664
qTESLA qTesla-256[a] (this document)	R-LWE	256[b]	pk: 6 432 sk: 4 128	5 920
Dilithium -high [30]	module SIS module LWE	125[b]	pk: 1 472 sk: –	2 700
GPV-poly[a] [34, 38]	R-SIS	96[c]	pk: 55 705 sk: 26 316	32 972
BLISS-B-IV [31, 57]	R-SIS, NTRU	182[c]	pk: 896 sk: 384	812
Selected other post-quantum signature schemes				
gravity-SPHINCS [12]	Hash collisions, 2nd preimage	128[b]	pk: 32 sk: 64	22 304
SPHINCS-256 [17]	Hash collisions, 2nd preimage	128[b]	pk: 1 056 sk: 1 088	41 000
MQDSS-31-64 [24]	Multivariate Quadratic system	128[b]	pk: 72 sk: 64	40 952
Selected classic signature schemes				
RSA-3072 [56]	Integer Factorization	128[d]	pk: 384 sk: 1 728	384
ECDSA (P-256) [44]	Elliptic Curve Discrete Logarithm	128[d]	pk: 64 sk: 96	64

[a] Parameters are chosen according to given security reduction in the quantum random oracle model.
[b] Bit security analyzed against classical and quantum adversaries.
[c] Bit security analyzed against classical adversaries.
[d] Broken against quantum computers (bit security analyzed against classical adversaries).

References

[1] Sedat Akleylek, Nina Bindel, Johannes A. Buchmann, Juliane Krämer, and Giorgia Azzurra Marson. An efficient lattice-based signature scheme with provably secure instantiation. In David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *Progress in Cryptology - AFRICACRYPT 2016 - 8th International Conference on Cryptology in Africa*, volume 9646 of *LNCS*, pages 44–60. Springer, 2016.

[2] Martin Albrecht, Florian Göpfert, Fernando Vidria, and Thomas Wunderer. Revisiting the Expected Cost of Solving uSVP and Applications to LWE. In *ASIACRYPT 2017 - Advances in Cryptology, to appear*. Springer, 2017.

[3] Martin R. Albrecht. On Dual Lattice Attacks Against Small-Secret LWE and Parameter Choices in HElib and SEAL. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, volume 10211 of *LNCS*, pages 103–129, 2017.

[4] Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. Algebraic algorithms for LWE problems. *ACM Comm. Computer Algebra*, 49(2):62, 2015.

[5] Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. On the complexity of the BKW algorithm on LWE. *Designs, Codes and Cryptography*, 74(2):325–354, 2015.

[6] Martin R. Albrecht, Robert Fitzpatrick, and Florian Göpfert. On the efficacy of solving LWE by reduction to unique-svp. In Hyang-Sook Lee and Dong-Guk Han, editors, *Information Security and Cryptology - ICISC 2013*, volume 8565 of *LNCS*, pages 293–310. Springer, 2013.

[7] Martin R. Albrecht, Robert Fitzpatrick, and Florian Göpfert. On the efficacy of solving LWE by reduction to unique-SVP. In Hyang-Sook Lee and Dong-Guk Han, editors, *ICISC 13: 16th International Conference on Information Security and Cryptology*, volume 8565 of *Lecture Notes in Computer Science*, pages 293–310, Seoul, Korea, November 27–29, 2014. Springer, Heidelberg, Germany.

[8] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

[9] Erdem Alkim, Nina Bindel, Johannes A. Buchmann, Özgür Dagdelen, Edward Eaton, Gus Gutoski, Juliane Krämer, and Filip Pawlega. Revisiting TESLA in the quantum random oracle model. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum*

Cryptography - 8th International Workshop, PQCrypto 2017, volume 10346 of *LNCS*, pages 143–162. Springer, 2017.

[10] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *25th USENIX Security Symposium, USENIX Security 16*, pages 327–343. USENIX Association, 2016.

[11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, *Automata, Languages and Programming*, volume 6755 of *LNCS*, pages 403–415. Springer, 2011.

[12] Jean-Philippe Aumasson and Guillaume Endignoux. Improving stateless hash-based signatures. Cryptology ePrint Archive, Report 2017/933, 2017. https://eprint.iacr.org/2017/933.

[13] László Babai. On lovász' lattice reduction and the nearest lattice point problem. In K. Mehlhorn, editor, *STACS 1985*. Springer, 1985.

[14] Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. In Josh Benaloh, editor, *Topics in Cryptology – CT-RSA 2014*, volume 8366 of *Lecture Notes in Computer Science*, pages 28–47, San Francisco, CA, USA, February 25–28, 2014. Springer, Heidelberg, Germany.

[15] Shi Bai and Steven D. Galbraith. Lattice decoding attacks on binary LWE. In Willy Susilo and Yi Mu, editors, *ACISP 14: 19th Australasian Conference on Information Security and Privacy*, volume 8544 of *Lecture Notes in Computer Science*, pages 322–337, Wollongong, NSW, Australia, July 7–9, 2014. Springer, Heidelberg, Germany.

[16] Paulo S. L. M. Barreto, Patrick Longa, Michael Naehrig, Jefferson E. Ricardini, and Gustavo Zanon. Sharper ring-lwe signatures. Cryptology ePrint Archive, Report 2016/1026, 2016. http://eprint.iacr.org/2016/1026.

[17] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: practical stateless hash-based signatures. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 9056 of *LNCS*, pages 368–397. Springer, 2015.

[18] Nina Bindel, Johannes Buchmann, Florian Göpfert, and Markus Schmidt. Estimation of the hardness of the learning with errors problem with a restricted number of samples. Cryptology ePrint Archive, Report 2017/140, 2017. https://eprint.iacr.org/2017/140.

[19] Nina Bindel, Johannes Buchmann, and Juliane Krämer. Lattice-based signature schemes and their sensitivity to fault attacks. In *2016 Workshop on Fault Diagnosis*

and Tolerance in Cryptography, FDTC 2016, pages 63–77. IEEE Computer Society, 2016.

[20] Nina Bindel, Johannes Buchmann, Juliane Krämer, Heiko Mantel, Johannes Schickel, and Alexandra Weber. Bounding the cache-side-channel leakage of lattice-based signature schemes using program semantics. In *Proceedings of the 10th International Symposium on Foundations & Practice of Security (FPS)*, 2017. To appear.

[21] Nina Bindel, Udyani Herath, Matthew McKague, and Douglas Stebila. Transitioning to a quantum-resistant public key infrastructure. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017*, volume 10346 of *LNCS*, pages 384–405. Springer, 2017.

[22] bushing, marcan, and sven. Console hacking 2010 – ps3 epic fail. 27th Chaos Communication Congress, 2010. https://events.ccc.de/congress/2010/Fahrplan/events/4087.en.html.

[23] Peter Campbell, Michael Groves, and Dan Shepherd. SOLILOQUY: A cautionary tale. ETSI 2nd Quantum-Safe Crypto Workshop, 2014. http://docbox.etsi.org/Workshop/2014/201410_CRYPTO/S07_Systems_and_Attacks/S07_Groves_Annex.pdf.

[24] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. From 5-pass $\mathcal{MQ}$ -based identification to $\mathcal{MQ}$ -based signatures. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security*, volume 10032 of *LNCS*, pages 135–165, 2016.

[25] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (Proposed Standard), May 2008.

[26] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 9666 of *LNCS*, pages 559–585. Springer, 2016.

[27] Özgür Dagdelen, Rachid El Bansarkhani, Florian Göpfert, Tim Güneysu, Tobias Oder, Thomas Pöppelmann, Ana Helena Sánchez, and Peter Schwabe. High-speed signatures from standard lattices. In Diego F. Aranha and Alfred Menezes, editors, *Progress in Cryptology – LATINCRYPT 2014*, volume 8895 of *LNCS*, pages 84–103. Springer, 2015.

[28] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008.

[29] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference*, volume 8042 of *LNCS*, pages 40–56. Springer, 2013.

[30] Leo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehle. CRYSTALS – Dilithium: Digital Signatures from Module Lattices. Cryptology ePrint Archive, Report 2017/633, 2017. http://eprint.iacr.org/2017/633.

[31] Léo Ducas. Accelerating bliss: the geometry of ternary polynomials. Cryptology ePrint Archive, Report 2014/874, 2014. http://eprint.iacr.org/2014/874/.

[32] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. Cryptology ePrint Archive, Report 2013/383, 2013. https://eprint.iacr.org/2013/383.

[33] M. J. Dworkin. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. National Institute of Standards and Technology (NIST), Gaithersburg (MD), USA, 8 2015.

[34] Rachid El Bansarkhani and Jan Sturm. An efficient lattice-based multisignature scheme with applications to bitcoins. In Sara Foresti and Giuseppe Persiano, editors, *CANS 2016*, pages 140–155, Cham, 2016. Springer International Publishing.

[35] Yara Elias, Kristin E. Lauter, Ekin Ozman, and Katherine E. Stange. Provably weak instances of ring-lwe. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference*, volume 9215 of *LNCS*, pages 63–92. Springer, 2015.

[36] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Loop-abort faults on lattice-based fiat-shamir and hash-and-sign signatures. In Roberto Avanzi and Howard M. Heys, editors, *Selected Areas in Cryptography - SAC 2016 - 23rd International Conference*, volume 10532 of *Lecture Notes in Computer Science*, pages 140–158. Springer, 2017.

[37] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 1–17, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany.

[38] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing (STOC 2008)*, pages 197–206. ACM, 2008.

[39] Florian Göpfert. *Securely Instantiating Cryptographic Schemes Based on the Learning with Errors Assumption*. PhD thesis, Darmstadt University of Technology, Germany, 2016.

[40] Florian Göpfert, Christine van Vredendaal, and Thomas Wunderer. A hybrid lattice basis reduction and quantum search attack on LWE. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017*, volume 10346 of *LNCS*, pages 184–202. Springer, 2017.

[41] Shay Gueron and Fabian Schlieker. Optimized implementation of ring-TESLA. GitHub at https://github.com/fschlieker/ring-TESLA, 2016.

[42] Qian Guo, Thomas Johansson, and Paul Stankovski. Coded-bkw: Solving LWE using lattice codes. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, volume 9215 of *LNCS*, pages 23–42. Springer, 2015.

[43] R. Housley. Cryptographic Message Syntax (CMS). RFC 5652 (INTERNET STANDARD), September 2009.

[44] James Howe, Thomas Pöppelmann, Máire O'neill, Elizabeth O'sullivan, and Tim Güneysu. Practical lattice-based digital signature schemes. *ACM Trans. Embed. Comput. Syst.*, 14, 2015.

[45] John Kelsey. Sha-3 derived functions: cshake, kmac, tuplehash, and parallelhash. *NIST Special Publication*, 800:185, 2016.

[46] Paul Kirchner and Pierre-Alain Fouque. An improved BKW algorithm for lwe with applications to cryptography and lattices. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, volume 9215 of *LNCS*, pages 43–62. Springer, 2015.

[47] Thijs Laarhoven. *Search problems in cryptography*. PhD thesis, Eindhoven University of Technology, 2016.

[48] Thijs Laarhoven, Michele Mosca, and Joop Pol. Solving the Shortest Vector Problem in Lattices Faster Using Quantum Search. In Philippe Gaborit, editor, *Post-Quantum Cryptography*, volume 7932 of *Lecture Notes in Computer Science*, pages 83–101. Springer Berlin Heidelberg, 2013.

[49] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, volume 6558 of *LNCS*, pages 319–339. Springer, 2011.

[50] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany.

[51] Vadim Lyubashevsky and Daniele Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference*, pages 577–594. Springer, 2009.

[52] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany.

[53] Damian Poddebniak, Juraj Somorovsky, Sebastian Schinzel, Manfred Lochter, and Paul Rösler. Attacking deterministic signature schemes using fault attacks. Cryptology ePrint Archive, Report 2017/1014, 2017. http://eprint.iacr.org/2017/1014.

[54] B. Ramsdell and S. Turner. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification. RFC 5751 (Proposed Standard), January 2010.

[55] Claus P. Schnorr and Taras Shevchenko. Solving subset sum problems of densioty close to 1 by randomized BKZ-reduction. Cryptology ePrint Archive, Report 2012/620, 2012. http://eprint.iacr.org/2012/620.

[56] Mikael Sjöberg. *Post-quantum algorithms for digital signing in Public Key Infrastructures*. PhD thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2017.

[57] 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.