Revisiting TESLA in the quantum random oracle model

Erdem Alkim[1], Nina Bindel[2], Johannes Buchmann[2], Özgür Dagdelen[3], Edward Eaton[4,5], Gus Gutoski[4], Juliane Krämer[2], and Filip Pawlega[4,5]

[1] Ege University, Turkey erdemalkim@gmail.com
[2] Technische Universität Darmstadt, Germany
{nbindel, buchmann, jkraemer}@cdc.informatik.tu-darmstadt.de
[3] BridgingIT GmbH, Germany oezdagdelen@googlemail.com
[4] ISARA Corporation, Canada
{ted.eaton, gus.gutoski, filip.pawlega}@isara.com
[5] University of Waterloo, Canada

Abstract. We study a scheme of Bai and Galbraith (CT-RSA'14), also known as TESLA. TESLA was thought to have a tight security reduction from the learning with errors problem (LWE) in the random oracle model (ROM). Moreover, a variant using chameleon hash functions was lifted to the quantum random oracle model (QROM). However, both reductions were later found to be flawed and hence it remained unresolved until now whether TESLA can be proven to be tightly secure in the (Q)ROM.
In the present paper we provide an entirely new, tight security reduction for TESLA from LWE in the QROM (and thus in the ROM). Our security reduction involves the adaptive re-programming of a quantum oracle. Furthermore, we propose parameter sets targeting 128 bits of security against both classical and quantum adversaries and compare TESLA's performance with state-of-the-art signature schemes.

Keywords: Quantum Random Oracle, Post Quantum Cryptography, Lattice-Based Cryptography, Signature Scheme, Tight Security Reduction

1 Introduction

Our interest in the present paper is in a quantum-resistant signature scheme proposed by Bai and Galbraith [6]. Those authors argue the security of their scheme via reductions from the *learning with errors (LWE)* and the *short integer solutions (SIS)* problems in the random oracle model (ROM). This scheme was subsequently studied by Alkim, Bindel, Buchmann, Dagdelen, and Schwabe under the name TESLA [4], who provided an alternate security reduction from the LWE problem only.

Since then, there have been several follow-up works on the Bai-Galbraith scheme [2, 4, 8, 47]. Most notably, a version of the scheme called ring-TESLA, whose security is based on the ring-LWE problem [2], has the potential to evolve into a practical, quantum-resistant signature scheme that might one day see widespread use as replacement for contemporary signature schemes such as ECDSA.

In what follows, we review the concepts of tightness and the quantum random oracle model as they relate to TESLA. We then list the contributions of the present paper and discuss related work by others.

1.1 Background

Security reduction and parameter choice. The security of digital signature schemes is often argued by reduction. A reductionist security argument typically proves a claim of the form, "any attacker $\mathcal{A}$ who can break the scheme can be used to build an algorithm $\mathcal{B}$ that solves some underlying hard computational problem". Hence, the security gap can be determined; it measures how much extra work $\mathcal{B}$ must perform in order to convert $\mathcal{A}$ into solving the underlying hard problem. If the run-time and probability of success of $\mathcal{B}$ are close to those of $\mathcal{A}$, *i.e.*, if the security gap is approximately 1, then the reduction is called *tight*. Achieving a small security gap, ideally a tight security reduction, is of theoretical interest in its own right, but it should also be an important consideration when selecting parameters for a concrete instantiation of a scheme. Specifically, the parameters of a signature scheme ought to be selected so that both (i) the effort needed to solve the underlying hard computational problem, and (ii) the security gap are taken into account. Hence, a tight security reduction is of advantage.

The need to instantiate schemes according to their security reductions and the role tight reductions play in these instantiations have been well argued by numerous authors. We refer the reader to [1, 18, 28] for a representative sample of these arguments.

The quantum random oracle model. Security arguments for the most efficient signature schemes—which therefore enjoy the most widespread real-world use—are typically presented in the ROM. (We refer to [31] by Koblitz and Menezes for discussion on why this might be the case.) The ROM postulates a truly random function that is accessible to attackers only through "black box" queries to an oracle for it—a random oracle. Any concrete proposal for a signature scheme must substitute a specific choice of hash function for the random oracle. An attacker armed with a quantum computer can be expected to evaluate that hash function in quantum superposition. Arguments that establish security even against such quantum-enabled attackers are said to hold in the quantum random oracle model (QROM).

It is conceivable that a signature scheme shown to be secure in the ROM may not be secure in the QROM. Thus, it is important that security arguments for quantum-resistant signature schemes hold not merely in the ROM, but also in the QROM.

Boneh *et al.* have proven that a security reduction in the ROM also holds in the QROM if it is *history-free* [15]. Unfortunately, many signature schemes have security reductions in the ROM that involve the *re-programming* of a random oracle; these reductions are not history-free. For these schemes, there remains a need to precisely clarify under what conditions these security reductions remain meaningful in the QROM.

Tightness in the QROM for TESLA. The security reduction presented by Bai and Galbraith for their signature scheme employs the Forking Lemma [41]. As such, it is non-tight and it involves re-programming, so it holds in the ROM but is not known to hold in the QROM.

As mentioned above, Alkim *et al.* presented an alternate security analysis for the Bai-Galbraith scheme, which they call TESLA. Their reduction is a tight reduction from LWE in the ROM. Moreover, those authors observed that their reduction can be made history-free at the cost of replacing a generic hash function with a chameleon hash function. It then follows from [15] that the history-free security reduction for TESLA holds also in the QROM. (Unfortunately, the use of a chameleon hash function would likely render any signature scheme too inefficient for widespread practical use.)

Unfortunately, a flaw in the original TESLA security reduction has been identified by the present authors. (The flaw was independently discovered by Chris Peikert.) This flaw is also present in several TESLA follow-up works, including ring-TESLA. As such, the status of the TESLA signature scheme and its derivative works has been open until now.

1.2 Our contribution

Our primary contributions are as follows:

New security reduction. We present a new security reduction from LWE to TESLA. Our new reduction is tight. It seems that the flaw in the original tight security reduction of TESLA does not admit a fix without a huge increase in the parameters; our new reduction is a significant re-work of the entire proof.

Security in the QROM with re-programming. Our new security reduction involves the adaptive re-programming of a random oracle and hence it is not history-free. Nevertheless, we show that it holds in the QROM by applying a seminal result from quantum query complexity due to Bennet, Bernstein, Brassard, and Vazirani [11]. It is possible that our approach can be abstracted so as to yield a general result on security reductions with re-programming in the QROM.

Our secondary contributions are as follows:

Parameter selection. We propose three sets of parameters for the concrete instantiation of TESLA: TESLA-0 and TESLA-1 targeting 96 and 128 bit security against a classical adversary, respectively; and TESLA-2, targeting 128 bits of security against a quantum adversary. All three parameter sets are chosen according to our (tight) security reduction.

The concrete parameter space admitted by our new security reduction is worse than that of previous reductions, but those previous reductions are either flawed or non-tight. Consequently, our proposed parameter sets lead to concrete instantiations of TESLA that are less efficient than previous proposals given in [4, 6, 47] that were not chosen according to the given security reduction.

Implementation. We provide a software implementation for the parameter sets TESLA-0 and TESLA-1. Our implementation targets Intel Haswell CPUs to provide a comparison of TESLA's performance with other signature schemes with different security levels. Unfortunately, the TESLA-2 parameter set does not seem to admit an implementation that can take advantage of the same fast parallel arithmetic instructions available on modern processors that were used in our implementations of TESLA-0 and TESLA-1, and so we do not provide a software implementation for TESLA at this parameter set. See Section 6 for details.

1.3 Related work

Tightness from "lossy" keys. In order to avoid the non-tightness inherent in the use of the Forking Lemma, we take an approach that was introduced by Katz and Wang to obtain tightly-secure signatures from the decisional Diffie-Hellman problem [28].

The idea is to use the underlying hardness assumption to show that "real", properly-formed public keys for the signature scheme are indistinguishable from "lossy", malformed public keys. The task of forging a signature for a lossy key is then somehow proven to be intractable.

Any attacker must therefore fail to forge when given a lossy public key. Thus, any attacker who succeeds in forging a signature when given a real public key can be used to distinguish real keys from lossy keys, contradicting the underlying hardness assumption.

In the case of TESLA, the real keys are matrices A and $\mathsf{T} = \mathsf{AS} + \mathsf{E}$ for some matrices S, E with small entries. (See Section 2.2 for a proper definition of these matrices and the LWE problem.) We call these real keys LWE yes-instances. The lossy keys are LWE no-instances: matrices A, T selected uniformly at random, so that the existence of S, E as above occurs with only negligible probability. We prove that the task of forging a TESLA signature for lossy keys is intractable, so that any TESLA forger must be able to solve the decisional LWE problem.

A Fiat-Shamir transform for "lossy" identification schemes. The TESLA signature scheme could be viewed as the result of applying the Fiat-Shamir transform to a "lossy" identification scheme based on LWE. A tight security reduction for TESLA then follows from a general theorem of Abdalla, Fouque, Lyubashevsky, and Tibouchi (AFLT theorem) on the tight security of any signature scheme obtained in this way [1].

In order to leverage the AFLT theorem, one must propose an identification scheme and prove that it is lossy. Such a proof could be obtained by excerpting the relevant parts of our security reduction to establish the simulatability and lossiness properties of a suitably chosen identification scheme. Such an exercise might make our rather monolithic security reduction easier to digest by modularizing it and phrasing it in a familiar framework.

However, security reductions obtained by applying the AFLT theorem are guaranteed to hold only in the ROM. In order to fully recover our security reduction from this framework, one must first re-prove the AFLT theorem in the QROM. This limitation is due to the fact that the proof of the AFLT theorem involves adaptively re-programming a hash oracle. As such, it does not meet any known conditions for lifting a given proof from the ROM into the QROM.

Given that our security reduction in the QROM also involves the adaptive re-programming of a hash oracle, perhaps our approach could be mined for insights to establish the AFLT theorem in the QROM.

Other tightly-secure LWE or SIS signature schemes. Gentry, Peikert, and Vaikuntanathan present a signature scheme with a tight security reduction from SIS in the ROM using a trapdoor construction based on possessing a secret short basis of a lattice [25]. Boneh *et al.* observed that the security reduction for this scheme is history-free, and thus holds in the QROM [15].

Boyen and Li present a signature scheme with a tight security reduction from SIS in the *standard model* [17], also using a short basis trapdoor. Since standard model security reductions do not rely on any assumptions about a random oracle, these reductions hold in the QROM.

The use of a short-basis trapdoor in a signature scheme imposes an additional constraint on the concrete parameter space admitted by that scheme's security reduction. This additional constraint on the parameters of short-basis trapdoor schemes seems to render them too inefficient for practical use. Since TESLA and its derivatives do not use a trapdoor construction, they do not suffer from this impediment.

Other than TESLA, we are aware of only one example of a signature scheme based on the Fiat-Shamir transform with a tight security reduction from LWE or SIS. Prior to Bai and Galbraith, a variant of a scheme by Lyubashevsky [33] was shown to admit a tight security reduction in the ROM by Abdalla *et al.* as part of an illustration of the aforementioned AFLT theorem [1]. An artifact of this reduction required Abdalla *et al.* to increase the parameters of the scheme, rendering it too inefficient for practical use. As mentioned earlier, security reductions produced via the AFLT theorem are not known to hold in the QROM.

Re-programming a quantum oracle. Adaptive reprogramming of a quantum oracle has been addressed in some specific cases. Unruh considered a re-programmed quantum oracle in order to establish the security of a quantum position verification scheme [45]. It is not clear whether Unruh's results apply to our setting.

Eaton and Song present an asymptotic result on re-programming in the QROM [24] in a context quite different from ours. Since their result is asymptotic, it does not allow for concrete parameter selection, for which the tightness of the reduction needs to be explicit.

Our approach to re-programming is independent of these previous works, though some works—such as [15,24]—do draw upon the same result by Bennet *et al.* [11] that we employ. To our knowledge we are the first to present progress on re-programming in the QROM in the context of a cryptographic scheme with potential for quantum-resistant standardization.

A note on "lattice-based" cryptography. Part of the allure of cryptosystems based on LWE or SIS is that those problems enjoy worst-case to average-case reductions from fundamental problems about lattices such as the *approximate shortest independent vectors problem (SIVP)* or the *gap shortest vector problem (GapSVP)*. (See Regev [42] or the survey of Peikert [38] and the references therein.)

These reductions suggest that the ability to solve LWE or SIS on randomly chosen instances implies the ability to solve SIVP or GapSVP, even on the hardest instances. Indeed, cryptosystems based on LWE or SIS are often referred to as *lattice-based* cryptosystems, suggesting that the security of these cryptosystems ultimately rests upon the worst-case hardness of these lattice problems.

However, as observed by Chatterjee, Koblitz, Menezes, and Sarkar, existing worst-case to average-case reductions for LWE and SIS are highly non-tight [18]. We are not aware of a proposal for a concrete instantiation of a cryptosystem based on LWE or SIS with the property that the proposed parameters were selected according to such a reduction. Instead, it is common to instantiate such cryptosystems based on the best known algorithms for solving LWE or SIS. (In addition to TESLA, see for example [5,16].)

For TESLA, we take care to instantiate the scheme according to its security reduction from LWE. However, we are unable to instantiate TESLA according to reductions from underlying lattice problems, due to the non-tightness of these reductions.

2 Preliminaries

In this section we clarify our notation used throughout the paper. We assume familiarity with the fundamentals of quantum information, such as the Dirac ket notation $|\cdot\rangle$ for pure quantum states and the density matrix formalism for mixed quantum states. (Recall that a mixed state can be viewed as a probabilistic mixture of pure states.) For background on quantum information the reader is referred to the books [29,37].

2.1 Notation

Integer scalars are denoted using Roman letters and if not stated otherwise, q is a prime integer in this paper. For any positive integer n the set $\mathbb{Z}_n$ of integers modulo n is represented by $\{-\lfloor (n-1)/2 \rfloor, \ldots, \lfloor n/2 \rfloor\}$. Fix a positive integer d and define the functions $[\cdot], [\cdot]_L : \mathbb{Z} \to \mathbb{Z}$ as follows. For any integer x let $[x]_L$ denote the representative of x in $\mathbb{Z}_{2^d}$, *i.e.*, $x = [x]_L \pmod{2^d}$, and let $[x] = (x - [x]_L)/2^d$. Informally, $[x]_L$ is viewed as the *least significant bits* of x and $[x]$ is viewed as the *most significant bits* of x. The definitions are easily extended to vectors by applying the operators for each component. An integer vector y is *B-short* if each entry is at most B in absolute value.

Vectors with entries in $\mathbb{Z}_q$ are viewed as column vectors and denoted with lowercase Roman letters in sans-serif font, *e.g.*, $\mathsf{y}, \mathsf{z}, \mathsf{w}$. Matrices with entries in $\mathbb{Z}_q$ are denoted with uppercase Roman letters in sans-serif font, *e.g.*, $\mathsf{A}, \mathsf{S}, \mathsf{E}$. The transpose of a vector or a matrix is denoted by v^T or M^T, respectively. We denote by $\|\mathsf{v}\|$ the Euclidean norm of a vector v, and by $\|\mathsf{v}\|_\infty$ its infinity norm. All logarithms are base 2. With $\mathcal{D}_\sigma$, we denote the centered discrete Gaussian distribution with standard deviation σ. For a finite set S, we denote sampling the element s uniformly from S with $s \leftarrow_\$ \mathcal{U}(S)$ or simply $s \leftarrow_\$ S$.

Let χ be a distribution over $\mathbb{Z}$, then we write $x \leftarrow \chi$ if x is sampled according to χ. Moreover, we denote sampling each coordinate of a matrix $\mathsf{A} \in \mathbb{Z}^{m \times n}$ with distribution χ by $\mathsf{A} \leftarrow \chi^{m \times n}$ with $m, n \in \mathbb{Z}_{>0}$. For an algorithm $\mathcal{A}$, the value $y \leftarrow \mathcal{A}(x)$ denotes the output of $\mathcal{A}$ on input x; if $\mathcal{A}$ uses randomness then $\mathcal{A}(x)$ is a random variable. $\mathcal{A}^\chi$ denotes that $\mathcal{A}$ can request samples from the distribution χ.

2.2 The Learning with Errors Problem

Informally the (decisional) learning with errors (LWE) problem with m samples is defined as follows: Given a tuple (A, t) with $\mathsf{A} \leftarrow_\$ \mathbb{Z}_q^{m \times n}$, decide whether $\mathsf{t} \leftarrow_\$ \mathbb{Z}_q^m$ or whether $\mathsf{t} = \mathsf{A}\mathsf{s} + \mathsf{e} \pmod q$ for a secret $\mathsf{s} \leftarrow \mathcal{D}_\sigma^n$ and error $\mathsf{e} \leftarrow \mathcal{D}_\sigma^m$. The security of the signature scheme covered in this paper is based on the matrix version of LWE (M-LWE): Given a tuple (A, T) with $\mathsf{A} \leftarrow_\$ \mathbb{Z}_q^{m \times n}$, decide whether $\mathsf{T} \leftarrow_\$ \mathbb{Z}_q^{m \times n'}$ is chosen uniformly random or whether $\mathsf{T} = \mathsf{A}\mathsf{S} + \mathsf{E} \pmod q$ for a secret $\mathsf{S} \leftarrow \mathcal{D}_\sigma^{n \times n'}$ and $\mathsf{E} \leftarrow \mathcal{D}_\sigma^{m \times n'}$. We call $(\mathsf{A}, \mathsf{T}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times n'}$ a **yes-instance** if T is generated by selecting $\mathsf{S} = (\mathsf{s}_1, ..., \mathsf{s}_{\mathsf{n}'})$ with $\mathsf{s}_1, ..., \mathsf{s}_{\mathsf{n}'} \leftarrow \mathcal{D}_\sigma^n$ and $\mathsf{E} \leftarrow \mathcal{D}_\sigma^{m \times n'}$, and setting $\mathsf{T} = \mathsf{A}\mathsf{S} + \mathsf{E} \pmod q$. Otherwise, when $(\mathsf{A}, \mathsf{T}) \leftarrow_\$ \mathcal{U}\left(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times n'}\right)$, we call (A, T) a **no-instance**. Similar concepts from the literature are also known as *lossy* [1, 10, 40] or *messy keys* [39].

We know that if an attacker can break LWE parametrized with n, m, and q in time t and with success probability ε/n', then he can solve M-LWE parametrized with n, n', m, and q in time t and with success probability ε. Intuitively this is correct since an adversary that can solve LWE has n' possibilities to solve M-LWE (see also [6, 16, 40]).

For the remainder of the paper, 'LWE' refers to the matrix version M-LWE, unless otherwise specified.

3 The Signature Scheme TESLA

In this section, we present the LWE-based signature scheme TESLA. Its orignal construction was proposed in 2014 by Bai and Galbraith [6]. It was later revisited by Dagdelen *et al.* [47] and by Alkim *et al.* [4].

TESLA's key generation, sign, and verify algorithms are listed informally in Algorithms 1, 2, and 3. More formal listings of these algorithms are given in Figure 1 in Section 5. Our proposed concrete parameter sets are derived in Section 5 and listed in Table 1.

Algorithm 1 KeyGen

```
Input: $\mathsf{A}$.
Output: Public key $\mathsf{T}$, secret key $(\mathsf{S}, \mathsf{E})$.
1: Choose entries of $\mathsf{S} \in \mathbb{Z}_q^{n \times n'}$ and $\mathsf{E} \in \mathbb{Z}_q^{m \times n'}$ from $\mathcal{D}_\sigma$
2: If $\mathsf{E}$ has a row whose $h$ largest entries sum to $L$ or more then retry at step 1.
3: If $\mathsf{S}$ has a row whose $h$ largest entries sum to $L_S$ or more then retry at step 1.
4: $\mathsf{T} \leftarrow \mathsf{AS} + \mathsf{E}$.
5: Return public key $\mathsf{T}$ and secret key $(\mathsf{S}, \mathsf{E})$.
```

TESLA is parameterized by positive integers q, m, n, n', h, d, B, L, L_S, U, a positive real σ, a hash oracle $\mathrm{H}(\cdot)$, and the publicly available matrix $\mathsf{A} \leftarrow_\$ \mathbb{Z}_q^{m \times n}$. Let $\mathbb{H}$ denote the set of vectors $\mathsf{c} \in \{-1, 0, 1\}^{n'}$ with exactly h nonzero entries. For simplicity we assume that the hash oracle $\mathrm{H}(\cdot)$ has range $\mathbb{H}$, *i.e.*, we ignore the encoding function F, cf. Table 1. We call an integer vector w *well-rounded* if w is $(\lfloor q/2 \rceil - L)$-short and $[\mathsf{w}]$ is $(2^d - L)$-short.

In contrast to earlier proposals [6,47], we add two additional checks. The first one is the check in Line 3 in Algorithm 1. It ensures that no coefficient of the matrix S is too large, which allows for more concrete bounds during the security reduction. The parameter L_S is chosen such that the probability of rejecting S is smaller than $2^{-\lambda}$, cf. Section 5. The second additional check is in Line 5 in Algorithm 2. To ensure correctness of the scheme, it checks that the absolute value of each coordinate of $\mathsf{Ay} - \mathsf{Ec}$ is less or equal than $\lfloor q/2 \rceil - L$.

Algorithm 2 Sign

Input: Message μ, secret key (S, E).
Output: Signature (z, c).

```
1: Choose $\mathsf{y}$ uniformly at random among $B$-short vectors from $\mathbb{Z}_q^n$.
2: $\mathsf{c} \leftarrow \mathrm{H}(\lfloor \mathsf{Ay} \rceil, \mu)$.
3: $\mathsf{z} \leftarrow \mathsf{y} + \mathsf{Sc}$.
4: If $\mathsf{z}$ is not $(B-U)$-short then retry at step 1.
5: If $\mathsf{Ay} - \mathsf{Ec}$ is not well-rounded then retry at step 1.
6: Return signature $(\mathsf{z}, \mathsf{c})$.
```

Algorithm 3 Verify

Input: Message μ, public key (A, T), purported signature (z, c).
Output: "Accept" or "reject".

```
1: If $\mathsf{z}$ is not $(B-U)$-short then reject.
2: If $\mathrm{H}(\lfloor \mathsf{Az} - \mathsf{Tc} \rceil, \mu) \neq \mathsf{c}$ then reject.
3: Accept.
```

4 Security Reduction for TESLA

Our main theorem on the security of TESLA informally states that as long as M-LWE can not be solved in time t and with success probability ε then no adversary $\mathcal{A}$ exists that can forge signatures of TESLA in time t' and with success probability ε', if $\mathcal{A}$ is allowed to make at most q_h hash und q_s sign queries. The main theorem is as follows.

Theorem 1 (Security of TESLA). *Let q, m, n, n', h, d, B, L, L_S, U, σ, λ, κ be TESLA parameters that are convenient*[6] *(according to Definition 1 in Section 5.3) and that satisfy the bounds in Table 1.*

If M-LWE is (t, ε)-hard then TESLA is existentially $(t', \varepsilon', q_h, q_s)$-unforgeable against adaptively chosen message attacks with $t' \approx t$ in (i) the quantum *random oracle model with*

$$\varepsilon' < \varepsilon + \frac{3}{2^\lambda} + \frac{2^{m(d+1)+3\lambda+1}}{q^m}(q_h + q_s)^2 q_s^3 + 2(q_h + 1)\sqrt{\frac{1}{2^h \binom{n'}{h}}}, \tag{1}$$

and in (ii) the classical *random oracle model with*

$$\varepsilon' < \varepsilon + \frac{3}{2^\lambda} + \frac{2^{m(d+1)+3\lambda+1}}{q^m}(q_h + q_s)^2 q_s^3 + q_h \frac{1}{2^h \binom{n'}{h}}. \tag{2}$$

[6] It is not necessary that TESLA parameters be convenient in order to derive negligibly small upper bounds on ε'; the definition of convenience merely facilitates a simplified statement of those bounds.

The proof of Theorem 1 is given in the full version of the paper. Here we present a sketch of this proof and a selection of some intermediate results we feel are the most significant technical contributions of the present manuscript.

Let $\mathcal{F}$ be a forger that forges signatures of the TESLA scheme with probability $\Pr[\text{forge}(\mathsf{A},\mathsf{T})]$, where $\text{forge}(\mathsf{A},\mathsf{T})$ denotes the event that $\mathcal{F}$ forges a signature on input (A,T), which is a yes- or a no-instance of LWE. We build an LWE-solver $\mathcal{S}$ whose run time is close to that of $\mathcal{F}$ and who solves LWE with success bias close to $\Pr[\text{forge}(\mathsf{A},\mathsf{T})]$. It then follows from the presumed hardness of LWE that $\Pr[\text{forge}(\mathsf{A},\mathsf{T})]$ must be small.

Given an LWE input (A,T), the LWE-solver $\mathcal{S}$ treats (A,T) as a TESLA public key; $\mathcal{S}$ runs $\mathcal{F}$ on input (A,T) and outputs "yes" if and only if $\mathcal{F}$ succeeds in forging a TESLA signature.

In order to run $\mathcal{F}$, the LWE-solver $\mathcal{S}$ must respond in some way to $\mathcal{F}$'s quantum queries to the hash oracle and to $\mathcal{F}$'s classical queries to the sign oracle. Our description of $\mathcal{S}$ includes a procedure for responding to these queries.

That $\mathcal{S}$ solves LWE with success bias close to $\Pr[\text{forge}(\mathsf{A},\mathsf{T})]$ is a consequence of the following facts:

1. For yes-instances of LWE, the probability with which $\mathcal{S}$ outputs "yes" is close to $\Pr[\text{forge}(\mathsf{A},\mathsf{T})]$.
2. For no-instances of LWE, $\mathcal{F}$ successfully forges (and hence $\mathcal{S}$ outputs "yes") with only negligible probability.

4.1 Yes-Instances of LWE

We argue that $\mathcal{S}$'s responses to $\mathcal{F}$'s oracle queries are indistinguishable from the responses $\mathcal{F}$ would receive from real oracles, from which it follows that $\mathcal{S}$ reports "yes" with probability close to $\Pr[\text{forge}(\mathsf{A},\mathsf{T})]$.

Each time $\mathcal{S}$ simulates a call to the sign oracle, it must "re-program" its simulated hash oracle on one input. Because $\mathcal{F}$ is permitted to make quantum queries to the hash oracle, we must show that $\mathcal{F}$ is unlikely to notice when a quantum random oracle has been re-programmed.

To this end, let $\mathbb{Y}$ denote the set of vectors $\mathsf{y} \in \mathbb{Z}_q^n$ such that y is B-short and define the following quantities for each choice of TESLA keys $(\mathsf{A},\mathsf{T}),(\mathsf{S},\mathsf{E})$:

$\text{nwr}(\mathsf{A},\mathsf{E})$: The probability over $(\mathsf{y},\mathsf{c}) \in \mathbb{Y}\times\mathbb{H}$ that $\mathsf{Ay}-\mathsf{Ec}$ is not well-rounded.
$\text{coll}(\mathsf{A},\mathsf{E})$: The maximum over all $\mathsf{w} \in \{[\mathsf{x}] : \mathsf{x} \in \mathbb{Z}_q^m\}$ of the probability over $(\mathsf{y},\mathsf{c}) \in \mathbb{Y}\times\mathbb{H}$ that $[\mathsf{Ay}-\mathsf{Ec}]=\mathsf{w}$.

We prove the following in the full version of our paper.

Proposition 1 (Re-Programming in TESLA, Informal Statement). *The following holds for each choice of TESLA keys $(\mathsf{A},\mathsf{T}),(\mathsf{S},\mathsf{E})$, each hash oracle $\mathrm{H}(\cdot)$, and each $\gamma > 0$.*

Suppose the quantum state ρ_{H} was prepared by some party $\mathcal{D}$ using t quantum queries to $\mathrm{H}(\cdot)$. Let $\mathrm{H}'(\cdot)$ be a hash oracle that agrees with $\mathrm{H}(\cdot)$ except on a small number of randomly chosen inputs $(\cdot,\mu)$ for each possible message μ. Let $\rho_{\mathrm{H}'}$ be the state prepared when $\mathcal{D}$ uses hash oracle $\mathrm{H}'(\cdot)$ instead of $\mathrm{H}(\cdot)$.

Then $\|\rho_{\mathrm{H}'} - \rho_{\mathrm{H}}\|_{\mathrm{Tr}} < \gamma$ *except with probability at most*

$$\frac{t^2}{\gamma^2} \cdot \frac{\mathrm{coll}(\mathsf{A},\mathsf{E})}{1 - \mathrm{nwr}(\mathsf{A},\mathsf{E})} \tag{3}$$

over the choice of inputs upon which $\mathrm{H}(\cdot)$ *and* $\mathrm{H}'(\cdot)$ *differ.*

We also prove bounds on $\mathrm{nwr}(\mathsf{A},\mathsf{E})$ and $\mathrm{coll}(\mathsf{A},\mathsf{E})$ that hold with high probability over the choice of TESLA keys $(\mathsf{A},\mathsf{T}), (\mathsf{S},\mathsf{E})$.

4.2 No-Instances of LWE

We argue that, except with negligibly small probability over the choice of hash oracle $\mathrm{H}(\cdot)$ and LWE no-instance (A,T), a TESLA forger cannot forge a signature for (A,T) without making an intractably large number of queries to the hash oracle.

To forge a signature for message μ, a forger must find a hash input (w,μ) whose output $\mathsf{c} = \mathrm{H}(\mathsf{w},\mu)$ has the property that there exists a $(B-U)$-short $\mathsf{z} \in \mathbb{Z}_q^n$ for which $[\mathsf{Az} - \mathsf{Tc}] = \mathsf{w}$. Let $\mathbb{H}(\mathsf{w},\mathsf{A},\mathsf{T}) \subset \mathbb{H}$ denote the set of all such c. A hash input (w,μ) is called *good* for $\mathrm{H}(\cdot)$ and (A,T) if $\mathrm{H}(\mathsf{w},\mu) \in \mathbb{H}(\mathsf{w},\mathsf{A},\mathsf{T})$. (Once a good hash input has been found, the forger must then somehow *find* the vector z witnessing this fact. For our purpose, we assume that the forger gets it for free.)

For each LWE no-instance (A,T), a given hash input (w,μ) is good for $\mathrm{H}(\cdot)$ and (A,T) with probability

$$\frac{\#\mathbb{H}(\mathsf{w},\mathsf{A},\mathsf{T})}{\#\mathbb{H}} \tag{4}$$

over the choice of hash oracle $\mathrm{H}(\cdot)$. In the full version of our paper, we argue that, except with negligibly small probability over the choice of $\mathrm{H}(\cdot)$ and (A,T), the fraction of hash inputs that are good is at most the expectation over LWE no-instances (A,T) of the ratio (4), maximized over all $\mathsf{w} \in \left\{[\mathsf{x}] : \mathsf{x} \in \mathbb{Z}_q^m\right\}$. We then prove the following

Proposition 2 (Good Hash Inputs are Rare).
If the TESLA parameters are convenient (according to Definition 1 in Section 5.3) then

$$\mathop{\mathrm{Ex}}_{(\mathsf{A},\mathsf{T})}\left[\max_{\mathsf{w}}\left\{\frac{\#\mathbb{H}(\mathsf{w},\mathsf{A},\mathsf{T})}{\#\mathbb{H}}\right\}\right] \le \frac{1}{\#\mathbb{H}}. \tag{5}$$

Thus, the fraction of good hash inputs is at most $1/\#\mathbb{H}$ except with vanishingly small probability over the choice of hash oracle $\mathrm{H}(\cdot)$ and LWE no-instance (A,T).

Since each hash input is good with a fixed probability independent of other hash inputs, the only way to discover a good input is via search through an unstructured space. It then follows from known lower bounds for quantum search over an unstructured space that the forger cannot find a good hash input—and thus a TESLA forgery—using only q_h quantum queries to the hash oracle.

5 Selecting Parameters for TESLA

In this section we propose parameter sets for TESLA. Moreover, we present a more detailed description of TESLA in Figure 1. Table 1 illustrates our concrete choice of parameters and Table 2 gives the hardness of the corresponding LWE instances. We propose three parameter sets: TESLA-0 that targets the same (classical) bit security of 96 bit as the instantiation proposed in [47], called DEG$^+$. TESLA-1 targets 128 bit of classical security and TESLA-2 targets 128 bit of security against quantum adversaries. Note that the parameter set DEG$^+$ was orignally proposed to give 128 bit of security, *i.e.*, $\lambda = 128$, but due to new methods to estimate the bit security its bit security is now only 96 bit.

```
Algorithm KeyGen

INPUT: $1^\lambda; \mathsf{A}, n, n', m, q, \sigma$
OUTPUT: $(\mathsf{S}, \mathsf{E}, s), \mathsf{T}$

1. $\mathsf{S} \leftarrow_\$ \mathcal{D}_\sigma^{n \times n'}$
2. $\mathsf{E} \leftarrow_\$ \mathcal{D}_\sigma^{m \times n'}$
3. if $\mathsf{checkE}(\mathsf{E}) = 0 \vee \mathsf{checkS}(\mathsf{S}) = 0$
4.     then Restart
5. $s \leftarrow_\$ \{0,1\}^\kappa$
6. $\mathsf{T} \leftarrow \mathsf{AS} + \mathsf{E} \pmod{q}$
7. $\mathrm{sk} \leftarrow (\mathsf{S}, \mathsf{E}, s), \mathrm{pk} \leftarrow \mathsf{T}$
8. return $(\mathrm{sk}, \mathrm{pk})$
```

```
Algorithm Verify

INPUT: $\mu, q, \mathsf{z}, c, \mathsf{A}, \mathsf{T}$
OUTPUT: {0,1}

1. $\mathsf{c} \leftarrow F(c)$
2. $\mathsf{w}' \leftarrow \mathsf{Az} - \mathsf{Tc} \pmod{q}$
3. $c' \leftarrow H(\left[\mathsf{w}'\right], \mu)$
4. if $c' = c \wedge \|\mathsf{z}\|_\infty \leq B - U$
5.     then return 1
6. return 0
```

```
Algorithm Sign

INPUT: $\mu, q, \mathsf{A}, \mathsf{S}, \mathsf{E}, s$
OUTPUT: $(\mathsf{z}, c)$

1. $j \leftarrow 0$
2. $\mathsf{k} \leftarrow PRF_1(s, \mu)$
3. $\mathsf{y} \leftarrow PRF_2(\mathsf{k}, j)$
4. $\mathsf{v} \leftarrow \mathsf{Ay} \pmod{q}$
5. $c \leftarrow H([\mathsf{v}], \mu)$
6. $\mathsf{c} \leftarrow F(c)$
7. $\mathsf{z} \leftarrow \mathsf{y} + \mathsf{Sc}$
8. $\mathsf{w} \leftarrow \mathsf{v} - \mathsf{Ec} \pmod{q}$
9. if $\|[\mathsf{w}]_L\|_\infty > 2^{d-1} - L_E$
     $\vee \|\mathsf{w}\|_\infty > \lfloor q/2 \rceil - L_E \vee \|\mathsf{z}\|_\infty > B - U$
10.   then $j \leftarrow j + 1$ and go to Step 1
11. return $(\mathsf{z}, c)$
```

Fig. 1. Specification of the signature scheme TESLA = (KeyGen, Sign, Verify); for details of the functions checkE and checkS see the explanation of the public parameters and definition of functions.

Public Parameters and Definition of Functions. TESLA is parameterized by the dimensions n, n', m of the matrices, the size κ of the output of the hash function, and the security parameter λ with $m > n > \kappa \geq \lambda$; by the matrix $\mathsf{A} \leftarrow_\$ \mathbb{Z}_q^{m \times n}$; by the hash function $H : \{0,1\}^* \to \{0,1\}^\kappa$, by the encoding function $F : \{0,1\}^\kappa \to \mathbb{H}$ (see [26] for more information), by the pseudo-random function

$PRF_1 : \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^\kappa$, and the pseudo-random generator $PRF_2 : \{0,1\}^\kappa \times \mathbb{Z} \to [-B, B]^n$. The remaining values, *i.e.*, the standard deviation σ, the number h of non-zero coefficients in the output of the encoding function, the number of rounded bits d, the value B defining the interval of the randomness during Sign, the value U defining (together with B) the rejection probability during rejection sampling, and the modulus q, are derived as shown in Table 1 and described in Sec. 5.1.

Moreover, we define the functions checkE, introduced in [47, Section 3.2], as follows: for a matrix E, define $\mathsf{E_i}$ to be the i-th row of E. The function $\max_k(\cdot)$ returns the k-th largest entry of a vector. The matrix E is rejected if for any row of E it holds that $\sum_{k=1}^{h} \max_k(\mathsf{E_i})$ is greater than some bound L. We apply a similar check checkS to S: The matrix S is rejected if for any row of S it holds that $\sum_{k=1}^{h} \max_k(\mathsf{S_i})$ is greater than some bound L_S.

Remark 1 (Deterministic signature). Note that signing is deterministic for each message μ since the randomness is determined by the vector y which is deterministically computed by the secret key and the message to-be-signed. In the original scheme by Bai and Galbraith [6] the vector y was sampled uniformly random in $[-B, B]^n$. The idea to use a pseudo-random function to generate signatures deterministically was deployed several times before [9,12,28,36,46].

5.1 Derivation of System Parameters

Our security reduction for TESLA minimizes the underlying assumptions which allows us to choose secure parameters from a greater set of choices compared to [6,47]. More precisely, our parameters do not have to involve a hard instance of the SIS assumption as it was done by Bai and Galbraith [6] before. We summarize the bounds and conditions of each parameter in Table 1 and explicate the computation of some of the listed parameters in the following. Furthermore, we state the resulting key and signature sizes in the table.

Compared to [6,47], we introduce the parameter n' as the column dimension of the secret matrices S and E to get more flexibility in the choice of parameters. The value of n' influences the parameters h (and hence B, U, q, and the encoding function F) and the size of the secret key.

Another important parameter of the signature scheme is the value L. In the original work [6], it is set to $L = 7h\sigma$, whereas it is set to $L = 3h\sigma$ in [47]. We choose L to be roughly $L = 2.8h\sigma$. We note that the smaller the value L, the higher the probability of acceptance in the signature algorithm (Line 9, Figure 1) becomes.

We add checkS to the key generation algorithm and the corresponding parameter L_S to bound $\|\mathsf{Sc}\| \leq L_S$ in the security reduction. We determine the value L_S such that S is rejected only with negligibly small (in the security parameter λ) probability. Hence, we do not decrease the size of the key space further. We choose L_S to be $14h\sigma$.

The acceptance probabilities of a signature δ_{Sign} and of a secret key (S, E) in Table 1 are determined experimentally.

Table 1. Concrete instantiation TESLA-2 of 128 bit of security against classical and quantum adversaries, and TESLA-0 of 96 bit and TESLA-1 of 128 bit of security against classical adversaries; comparison with the instantiation proposed in [47], called DEG$^+$, of 96 bit security (classically); sizes are given in kilo byte [KB]; sizes are theoretic sizes for fully compressed keys and signatures; for sizes used by our software see Table 3.

Parameter	Bound	DEG$^+$	TESLA-0	TESLA-1	TESLA-2
λ		128	96	128	128
κ		256	256	256	256
n		532	644	804	1300
n'		532	390	600	1036
m		840	3156	4972	4788
σ	$> 2\sqrt{n}$	43	55	57	73
L	$3h\sigma$ or $2.8h\sigma$, see Sec. 5.1	2322	5082	6703	17987
L_S	$14\sigma h$	-	25410	33516	89936
h	$2^h\binom{n'}{h} \geq 2^{3\lambda}$ (classically)	18	33	42	-
	$2^h\binom{n'}{h} \geq 2^{5\lambda}$ (quantumly)	-	-	-	88
B	$\geq 14n\sqrt{h}\sigma$	$2^{21}-1$	$2^{22}-1$	$2^{22}-1$	$2^{24}-1$
U	$\lceil 14\sqrt{h}\sigma \rceil$	2554	4424	5172	9588
d	$(1-2L/2^d)^m \geq 0.3$	23	25	26	27
q	satisfying the bound in Eq. 7, $\geq \left(2^{m(d+1)+4\lambda+1}(q_h+q_s)^2 q_s^3\right)^{1/m}$	$2^{29}-3$	$2^{31}-99$	$2^{31}-19$	40582171961 $\approx 2^{35.24}$
δ_{KeyGen}	empirically, see Sec. 5.1	0.99	1	1	future work
δ_{Sign}		0.314	0.307	0.154	future work
H	$\{0,1\}^* \to \{0,1\}^\kappa$	SHA-256			
F	$\{0,1\}^\kappa \to \mathbb{H}_{n',\omega}$	see [26]			
PRF_1	$\{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^\kappa$	-	SHA-256		
PRF_2	$\{0,1\}^\kappa \times \mathbb{Z} \to [-B,B]^n$	-	ChaCha20		
public-key size	$mn'\lceil \log_2(q) \rceil$	1 582	4 657	11 288	21 799
secret-key size	$(nn'+mn')\lceil \log_2(14\sigma) \rceil$	891	1 809	4 230	7 700
signature size	$n\lceil \log_2(2(B-U)) \rceil + \kappa$	1.4	1.8	2.3	4.0

To ensure both correctness and security of our signature scheme, we choose parameters with respect to our reduction, hence, we choose parameters such that $\epsilon' \approx \epsilon$ in Equation (1) and (2). We propose to choose $q_h \leq 2^\lambda$ and $q_s \leq 2^{\lambda/2}$, since a hash query is merely the evaluation of a publicly available function and hence the adversary can use all its computational power to make hash queries. The number of sign queries is somewhat limited since it involves more complicated operations. We refer to [30] (especially, Section 7) for further discussion.

5.2 Concrete Bit Security of TESLA

Choosing our parameters such that $\varepsilon \approx \varepsilon$ and $t \approx t'$ in Theorem 1 implies that we do not lose bits of security due to our security reduction. However, we lose

$\lceil \log(n') \rceil$ bits of security due to the reduction from LWE to M-LWE. Hence, we have to choose an LWE instance with slightly higher bit hardness than the targeted bit security of the TESLA instances.

To estimate the classical hardness we use a recent fork [43, 44] of the LWE-Estimator by Albrecht, Player, and Scott [3]. The extension takes the number of given LWE samples into account.

To estimate the quantum hardness of LWE we use the same method: we use the LWE-Estimator which already includes (from commit-id b929691 on) the run time estimates for a quantumly enhanced sieving algorithm [32] as a subroutine of the lattice reduction algorithm BKZ 2.0 [20]. Moreover, we apply a recently published quantum algorithm [35] to the currently fastest enumeration estimations by Micciancio and Walter [34] and add the resulting estimations as a subroutine to be used in BKZ 2.0. We summarize the estimations using quantum sieving and quantum enumeration in Table 2.

Table 2. Estimation of the hardness of LWE instances given in TESLA-0, TESLA-1, and TESLA-2 against the decoding attack and the (dual and standard) embedding approach, in comparison to the parameter sets proposed by Dagdelen *et al.* [47], called DEG$^+$; estimations are computed using the LWE-Estimator with a restricted number of samples [3, 44].

Problem	Attack	DEG$^+$	TESLA-0	TESLA-1	TESLA-2
Classical Hardness [bit]					
LWE	Decoding	156	**110**	**142**	**204**
	Dual Embedding	**96**	**110**	**142**	205
	Standard Embedding	164	111	143	205
Post-Quantum Hardness [bit]					
LWE	Decoding	73	74	98	146
	Dual Embedding	**61**	**71**	**94**	**142**
	Standard Embedding	111	**71**	95	**142**

5.3 Convenient Parameters

We make some simplifying assumptions on the choice of TESLA parameters. These assumptions are not necessary in order to derive a negligibly small upper bound on the forger's success probability—they merely facilitate a simplified statement of the upper bound in Theorem 1 in Section 4.

Let $\Delta\mathbb{H}$ be the set of differences of elements in $\mathbb{H}$. That is, $\Delta\mathbb{H} \overset{\text{def}}{=} \{c - c' : c, c' \in \mathbb{H}\}$. In the full version of our paper we compute the size of $\Delta\mathbb{H}$, but for a trivial upper bound one can note that $\#\Delta\mathbb{H} \leq (\#\mathbb{H})^2$.

Definition 1 (Convenient TESLA Parameters). *TESLA parameters are* convenient *if the following bounds hold:*

$$2mL\left(\frac{1}{2^d}+\frac{1}{q}\right)+\sqrt{\frac{2^\lambda(q+1)}{(2B-1)^n}} < 1/2 \tag{6}$$

$$\#\varDelta\mathbb{H}(4(B-U)-1)^n(2^{d+1}-1)^m < q^m. \tag{7}$$

All our proposed parameter sets for TESLA meet this condition.

6 Results and Comparison

To evaluate the performance of our proposed parameter sets we present a software implementation targeting the Intel Haswell microarchitecture. The starting point for our implementation is the software presented by Dagdelen *et al.* [47], which we obtained from the authors. Our software offers the same level of protection against timing attacks as the software presented in [47]. The software makes use of the fast AVX2 instructions on vectors of four double-precision floating-point numbers.

Table 3 gives benchmarking results for TESLA-0 and TESLA-1, and compares those benchmarks to state-of-the-art results from the literature. Due to the large values q and B of the parameter set TESLA-2, certain elements do no fit into the 53-bit mantissa of a double-precision floating point variable. Hence, we do not compare the performance of TESLA-2 in Table 3.

We obtain our benchmarks on an Intel Core-i7 4770K (Haswell) processor while disabling Turbo Boost and hyperthreading. Benchmarks of TESLA for signing are averaged over 100,000 signatures; benchmarks of TESLA for verification are the median of 100 verifications. The reason for not reporting the median for TESLA signing performance is that because of the rejection sampling, it would be overly optimistic. For all software results we report the sizes of keys and signatures actually produced by the software, not the theoretically smallest possible sizes with full compression.[7]

As can be seen in Table 3, TESLA is several magnitudes faster and sizes are smaller than the only other lattice-based signature scheme that is also proven tightly secure in the quantum random oracle model for the same (classical) security of 96 bits. However, the signature generation and verification algorithms of TESLA-0 are much slower than the implementation of [47] for the same level of security. This is due to the large difference of the parameters chosen, *e.g.*, the matrix dimension m in TESLA-0 is 3156, while $m = 840$ in the parameter set DEG$^+$ proposed by Dagdelen *et al.* [47]. Note that the parameter set TESLA-0 is chosen according to our security reduction, while the set DEG$^+$ is not chosen according to the (non-tight) security reduction given in [6].

[7] We make an exception for BLISS. The authors of the software obviously did not spend any effort on reducing the size of signatures and keys; we report sizes with "trivial" compression through choosing native data types of appropriate sizes.

In the (as of yet quite small) realm of signatures that offer 128 bits of post-quantum security, TESLA-2 offers an alternative to SPHINCS. Public and secret keys of TESLA-2 are much larger than SPHINCS keys, but signatures are several magnitudes smaller. The post-quantum multivariate-based signature scheme Rainbow5640 [19, 21] performs best among all listed schemes but unfortunately, comes with no security reduction to its underlying problem.

Acknowledgments

We are especially grateful to Peter Schwabe for contributions to our software implementation and to the presentation of the paper. We thank Chris Peikert for pointing out a flaw in previous security reductions for TESLA in the random oracle model. We thank Steven Galbraith and anonymous reviewers for valuable feedback on an earlier version of this manuscript.

This work has been supported by the German Research Foundation (DFG) as part of project P1 within the CRC 1119 CROSSING, by TÜBITAK under 2214-A Doctoral Research Program Grant and 2211-C PhD Scholarship, by Ege University under project 2014-FEN-065, and by CryptoWorks21.

References

1. Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-Secure Signatures from Lossy Identification Schemes. In *EUROCRYPT 2012*, LNCS. Springer, 2012.
2. Sedat Akleylek, Nina Bindel, Johannes A. Buchmann, Juliane Krämer, and Giorgia Azzurra Marson. An Efficient Lattice-Based Signature Scheme with Provably Secure Instantiation. In *AFRICACRYPT 2016*, volume 9646 of *LNCS*. Springer, 2016.
3. Martin R. Albrecht, Rachel Player, and Sam Scott. On the Concrete Hardness of Learning with Errors. *Journal of Mathematical Cryptology*, 9, 2015.
4. Erdem Alkim, Nina Bindel, Johannes Buchmann, Özgür Dagdelen, and Peter Schwabe. TESLA: Tightly-Secure Efficient Signatures from Standard Lattices. Cryptology ePrint Archive, Report 2015/755, version 20161117:055833, 2015.
5. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-Quantum Key Exchange – a New Hope. In *25th USENIX Security Symposium*. USENIX Association, 2016.
6. Shi Bai and Steven D. Galbraith. An Improved Compression Technique for Signatures Based on Learning with Errors. In *Topics in Cryptology – CT-RSA 2014*, volume 8366 of *LNCS*. Springer, 2014.
7. Rachid El Bansarkhani and Johannes Buchmann. Improvement and Efficient Implementation of a Lattice-Based Signature Scheme. In *Selected Areas in Cryptography*, volume 8282 of *LNCS*. Springer, 2013.
8. 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.
9. George Barwood. Digital Signatures Using Elliptic Curves. message `32f519ad.19609226@news.dial.pipex.com` posted to sci.crypt, 1997. `http://groups.google.com/group/sci.crypt/msg/b28aba37180dd6c6`.

10. Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and Impossibility Results for Encryption and Commitment Secure under Selective Opening. In *EUROCRYPT 2009*, volume 5479 of *LNCS*. Springer, 2009.
11. Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and Weaknesses of Quantum Computing. *SIAM J. Comput.*, 26(5), 1997.
12. Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-Speed High-Security Signatures. In *Cryptographic Hardware and Embedded Systems – CHES 2011*, volume 6917 of *LNCS*. Springer, 2011.
13. Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: Practical Stateless Hash-Based Signatures. In *EUROCRYPT 2015*, volume 9056 of *LNCS*. Springer, 2015.
14. Daniel J. Bernstein and Tanja Lange. eBACS: ECRYPT Benchmarking of Cryptographic Systems. `http://bench.cr.yp.to` (accessed 2015-05-19).
15. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random Oracles in a Quantum World. In *ASIACRYPT 2011*, volume 7073 of *LNCS*. Springer, 2011.
16. Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE. In *CCS 2016*. ACM, 2016.
17. Xavier Boyen and Qinyi Li. Towards Tightly Secure Lattice Short Signature and Id-Based Encryption. In *ASIACRYPT 2016*, volume 10032 of *LNCS*. Springer, 2016.
18. Sanjit Chatterjee, Neal Koblitz, Alfred Menezes, and Palash Sarkar. Another Look at Tightness II: Practical Issues in Cryptography. Cryptology ePrint Archive, Report 2016/360, 2016.
19. Anna Inn-Tung Chen, Ming-Shing Chen, Tien-Ren Chen, Chen-Mou Cheng, Jintai Ding, Eric Li-Hsiang Kuo, Frost Yu-Shuang Lee, and Bo-Yin Yang. SSE Implementation of Multivariate PKCs on Modern x86 CPUs. In *CHES 2009*, volume 5747 of *LNCS*. Springer, 2009.
20. Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In *ASIACRYPT 2011*, volume 7073 of *LNCS*. Springer, 2011.
21. Jintai Ding and Dieter Schmidt. Rainbow, a New Multivariable Polynomial Signature Scheme. In *Applied Cryptography and Network Security*, volume 3531 of *LNCS*. Springer, 2005.
22. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice Signatures and Bimodal Gaussians. In *CRYPTO 2013*, volume 8042 of *LNCS*. Springer, 2013.
23. Léo Ducas. Accelerating Bliss: the Geometry of Ternary Polynomials. Cryptology ePrint Archive, Report 2014/874, 2014.
24. Edward Eaton and Fang Song. Making Existential-Unforgeable Signatures Strongly Unforgeable in the Quantum Random-Oracle Model. In *10th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2015*, 2015.
25. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *STOC 2008*. ACM, 2008.
26. Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems. In *CHES 2012*, volume 7428 of *LNCS*. Springer, 2012.

27. Tim Güneysu, Tobias Oder, Thomas Pöppelmann, and Peter Schwabe. Software Speed Records for Lattice-Based Signatures. In *Post-Quantum Cryptography*, volume 7932 of *LNCS*. Springer, 2013.
28. Jonathan Katz and Nan Wang. Efficiency Improvements for Signature Schemes with Tight Security Reductions. In *CCS 2003*. ACM, 2003.
29. Phillip Kaye, Raymond Laflamme, and Michele Mosca. *An Introduction to Quantum Computing*. Oxford University Press, Inc., New York, NY, USA, 2007.
30. Neal Koblitz and Alfred Menezes. Another Look at "Provable Security". II. In *INDOCRYPT 2006*, volume 4329 of *LNCS*. Springer, 2006.
31. Neal Koblitz and Alfred Menezes. The Random Oracle Model: a Twenty-Year Retrospective. *Designs, Codes and Cryptography*, 77(2), 2015.
32. Thijs Laarhoven, Michele Mosca, and Joop van de Pol. Finding Shortest Lattice Vectors Faster Using Quantum Search. *Designs, Codes and Cryptography*, 2015.
33. Vadim Lyubashevsky. Lattice Signatures without Trapdoors. In *EUROCRYPT 2012*, volume 7237 of *LNCS*. Springer, 2012.
34. Daniele Micciancio and Michael Walter. Fast Lattice Point Enumeration with Minimal Overhead. In *SODA 2015*. SIAM, 2015.
35. Ashley Montanaro. Quantum Walk Speedup of Backtracking Algorithms. arXiv preprint arXiv:1509.02374, 2016.
36. David M'Raïhi, David Naccache, David Pointcheval, and Serge Vaudenay. Computational Alternatives to Random Number Generators. In *Selected Areas in Cryptography*, volume 1556 of *LNCS*. Springer, 1998.
37. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, New York, 2000.
38. Chris Peikert. A Decade of Lattice Cryptography. Cryptology ePrint Archive, Report 2015/939, 2015.
39. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A Framework for Efficient and Composable Oblivious Transfer. In *CRYPTO 2008*, volume 5157 of *LNCS*. Springer, 2008.
40. Chris Peikert and Brent Waters. Lossy Trapdoor Functions and Their Applications. In *STOC 2008*. ACM, 2008.
41. David Pointcheval and Jacques Stern. Security Proofs for Signature Schemes. In *EUROCRYPT 1996*, volume 1070 of *LNCS*. Springer, 1996.
42. Oded Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In *STOC 2005*. ACM, 2005.
43. Markus Schmidt. Estimation of the Hardness of the Learning with Errors Problem with a Restricted Number of Samples. GitHub at `https://bitbucket.org/Ma_Schmidt/lwe-estimator`, 2017.
44. Markus Schmidt and Nina Bindel. Estimation of the Hardness of the Learning with Errors Problem with a Restricted Number of Samples. Cryptology ePrint Archive, Report 2017/140, 2017.
45. Dominique Unruh. Quantum Position Verification in the Random Oracle Model. In *CRYPTO 2014*, volume 8617 of *LCNS*. Springer, 2014.
46. John Wigley. Removing need for RNG in Signatures. message `5gov5dpad@wapping.ecs.soton.ac.uk` posted to sci.crypt, 1997. `http://groups.google.com/group/sci.crypt/msg/a6da45bcc8939a89.`
47. 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 *LATINCRYPT 2014*, volume 8895 of *LNCS*. Springer, 2015.

Table 3. Overview of state-of-the-art post-quantum signature schemes; signature sizes are given in byte [B], key sizes are given in kilo byte [KB]; the column “ROM?, tight?” states whether the scheme has a security reduction in the random oracle model and whether this reduction is tight; “QROM?, tight?” states the same for the quantum random oracle model; “Security (PreQ)” lists the claimed pre-quantum security level; “Security (PostQ)” lists the claimed post-quantum security level, if available

Scheme/ Software	Comp. Assum.	ROM? Tight?	QROM? Tight?	Security (PreQ)	 (PostQ)	CPU	Key Size [KB]	Sig. Size [B]	Cycle counts
Selected signature schemes over standard lattices									
GPV [7, 25]	SIS	yes yes	yes yes	96	59	AMD Opteron 8356 (Barcelona)	vk: 27,840 sk: 12,064	30,105	sign: 312,800,000 verify: 50,600,000
DEG$^+$ [6, 47]	SIS, LWE	yes no	–	96	?	Intel Core i7-4770K (Haswell)	vk: 1,581 sk: 891	1,495	sign: 1,203,924 verify: 335,072
TESLA-0 (this paper)	LWE	yes yes	yes yes	96	?	Intel Core-i7-4770K (Haswell)	vk: 4,808 sk: 2,895	1,964	sign: 27,243,747 verify: 5,374,884
TESLA-1 (this paper)	LWE	yes yes	yes yes	128	?	Intel Core-i7-4770K (Haswell)	vk: 11,653 sk: 6,769	2,444	sign: 143,402,231 verify: 19,284,672
Selected signatures schemes over ideal lattices									
GPV-poly [7, 25]	R-SIS	yes yes	yes yes	96	59	AMD Opteron 8356 (Barcelona)	vk: 55 sk: 26	32,972	sign: 80,500,000 verify: 11,500,000
GLP [26, 27, 47][a]	DCK	yes no	–	75–80	?	Intel Core i5-3210M (Ivy Bridge)	vk: 1.5 sk: 0.25	1,186	sign: 452,223 verify: 34,004
Bliss-bI [22, 23][b]	R-SIS, NTRU	yes no	–	128	?	“Intel Core @ 3.4GHz”	vk: 7 sk: 2	1,559	sign: ≈358,400 verify: 102,000
Selected other post-quantum signature schemes									
SPHINCS-256 [13]	Hash collisions 2nd preimage	yes no[c]	yes no[c]	>128	128	Intel Xeon E3-1275 (Haswell)	vk: 1 sk: 1	41,000	sign: 51,636,372 verify: 1,451,004
Rainbow5640 [19, 21][d]	MQ, EIP[e]	–	–	80	?	Intel Xeon E3-1275 (Haswell)	vk: 43 sk: 84	37	sign: 42,700 verify: 36,072

[a] In the benchmarks we include the improvements by Dagdelen et al. presented in [47].

[b] We report sizes of keys and signatures with “trivial” compression as explained in the text.

[c] The security of SPHINCS is reduced tightly from the hardness of finding hash collisions and non-tightly from the hardness of finding 2nd preimages in the standard model. Hence the reduction also holds in the ROM and QROM.

[d] Benchmark on Haswell CPU from [14].

[e] The security of Rainbow5640 is based on the Multivariate Quadratic polynomial (MQ) and the Extended Isomorphism of Polynomials (EIP) problem, but no security reduction has been given yet.