# Quantum-Secure Key Exchange from LWE

*(Extended Abstract)*

Valeria Nikolaenko
Stanford University
Email: valerini@stanford.edu

Ilya Mironov
Google Inc.
Email: mironov@google.com

Ananth Raghunathan
Google Inc.
Email: pseudorandom@google.com

***Abstract*—Lattice-based cryptography offers the most attractive primitives believed to be resistant to quantum computers. Recently, following increasing interests by both private companies and government agencies in building practical quantum computers, Bos, Costello, Naehrig, and Stebila (IEEE S&P 2015) showed a practical *post-quantum* key exchange protocol based on hard problems on *ideal* lattices. In this paper, we develop and evaluate a secure and practical key exchange protocol based on hard problems on *generic* lattices (Learning With Errors). We initiate this study noting that the hardness of lattice problems on regular and ideal lattices merits further cryptanalysis and recently there have been significant strides in attacking *some* weak problem instances over ideal lattices as well as improved attacks on lattices.**

**We demonstrate the feasibility of LWE-based key exchange for internet deployment; in the process of which we introduce techniques to optimize communication bandwidth in lattice protocols that may be of independent interest. Our microbenchmark evaluations of our schemes are promising—requiring about 2.4× compute and about 2× bandwidth overhead to move from ideal to generic lattices and we mention practical research directions going forward.**

## I. Introduction

Traditional number-theoretic primitives such as the RSA problem, the closely related integer factorization problem, and the discrete log problem (over both multiplicative groups modulo a prime $p$ and elliptic curve groups) are vulnerable to polynomial-time quantum attacks. Lattice-based cryptography, beginning with the seminal work of Ajtai [4] (see the recent survey [48] for a more comprehensive list of relevant references) is an exciting field of research that offers the greatest promise of building practical quantum-resistant cryptographic primitives in addition to providing a richer diversity of underlying primitives.

Recently, there has been a renewed interest in developing practical quantum-secure cryptography under the umbrella of *post-quantum* cryptography. Companies [45, 53, 12, 25] and governments [52] alike are investing heavily in building quantum computers. Starting with four qubits to factor $15 = 3 \cdot 5$ [51, 36], the last two years has shown progress in applying quantum algorithms to factorizing larger numbers.[1] This has also received attention and encouragement from several government agencies—the National Institute of Standards and Technology (NIST) [43], the National Security Agency (NSA) [44], and the European Union's PQCRYPTO project [8].

### A. Key Exchange and Forward Secrecy

Cryptography used to secure the internet comprises several building blocks: key exchange protocols, public-key encryption schemes, signatures, symmetric-key encryption schemes, and message-authentication codes (MACs). The advent of practical quantum computers will affect existing deployments of these building blocks in different ways. Currently the only quantum speedups affecting symmetric-key schemes are generic speedups via Grover's algorithm [28]. It should suffice for symmetric-key encryption schemes and MACs to double their secret-key sizes to defend against quantum computers.

Signature schemes and certificates have several interesting post-quantum candidates currently under consideration. (See §II for more details.) Moreover, we note that the day a quantum computer capable of forging RSA and ECDSA signatures is built, integrity of future connections is suspect. The integrity of *past* data communicated under then-secure certificates and signatures still holds up.

With public-key encryption, specifically key exchange protocols, the secrecy of communications *today* will be compromised by a quantum computer built decades from now. This aligns with the notion of *perfect forward secrecy* [23] deployed in TLS or the decades-long time horizons for classified documents by government agencies [10, Table 4]. Therefore, in this paper, we focus our attention on the most pressing post-quantum crypto problem—that of secure, practical, post-quantum key exchange to secure internet traffic.

### B. Generic vs. Ideal Lattices

In this paper, we develop and evaluate a secure and practical key exchange protocol from the Learning With Errors (LWE) problem [49] with internet deployment in mind. LWE is a mature and well-studied [7, 35, 41, 16] cryptographic primitive that relies only on the *worst-case* hardness of a shortest vector problem in *generic* lattices. To contrast, the existing state-of-the-art key exchange scheme from lattices [47, 14] (which serves as inspiration for our work) relies on the (worst-case) hardness of a shortest vector problem on *ideal* lattices and the corresponding Ring Learning With Errors (RLWE) problem [40]. Ideal lattices are lattices generated by embeddings of ideals in polynomial rings. Another class of ideal lattices, the NTRU lattices [29], has also been used to build cryptosystems.

The hardness of lattice problems on regular lattices as well as ideal lattices merits more study. Although the algebraic structure of RLWE (and NTRU) make it more promising in getting to practical key-sizes and protocol communication, this algebraic structure *might* inspire less confidence in their security. Currently, the best algorithms against ideal lattices [20, 34] are the

[1] We can now factor $56153 = 233 \cdot 241$ [22].

same as those against regular lattices (modulo small polynomial speedups). However, in the past two years alone there have been several promising research results demonstrating the existence of, and attacks on weak instances of ideal lattices. (See [21, 26] and references therein about the Soliloquy attacks.)

### C. Our Contributions

Our contributions can be summarized into three main points.

1) We demonstrate the feasibility of LWE-based key-exchange protocol by designing and optimizing existing protocols to conserve bandwidth. We believe these optimization techniques will be of independent interest for practical lattice-based cryptography.
2) We incorporate parameter estimates that include recent developments in lattice cryptanalysis.
3) We evaluate our design[2] with the OpenSSL library and provide a broader context for its performance in practical systems. In microbenchmarks, our LWE-based key exchange is only **2.4×** slower and takes up bandwidth that is about **2×** larger than the RLWE scheme. Compared to typical user latencies, the compute overhead is negligible but the bandwidth still might give us pause.

This shows us that we need not rely exclusively on ideal lattices to construct practical key exchange schemes and our work motivates two interesting practical research directions going forward:

1) Can we improve bandwidth and memory footprint (particularly the in-memory parameter $\mathbf{A}$) of lattice-based crypto while retaining security reductions to problems over generic lattices?
2) Can we design noise distributions that are more efficient to sample (hence improving compute overhead) or rely on Learning with Rounding [9] with tighter parameters without correspondingly increasing lattice dimensions?

In §II, we describe related work. §III describes our main results—the new key exchange protocol from LWE, our generalized reconciliation mechanism to conserve bandwidth, and how to securely apply truncation to further reduce bits transmitted on the wire. We provide a statement of the proof of security in §IV and describe our evaluation in §V.

## II. Related Work

The Learning with Errors problem was introduced by Regev [49] who showed a (quantum) reduction to certain worst-case problems in lattices. Brakerski *et al.* [16] showed that LWE is classicaly reducible to worst-case problems in lattices. LWE has been used to construct a large variety of public-key cryptosystems [46, 27, 19, 3, 2, 18].

The ring analogue [40] also yields efficient constructions of public-key cryptosystems [47] including FHE [17, 42] and was the basis for the key-exchange protocol by Bos *et al.* [14]. Bos *et al.* design and prove secure an RLWE-based key exchange protocol and demonstrate its feasibility by evaluating benchmarks against ECDHE. Ding *et al.* [33] start off with the Regev public-key cryptosystem and give a DH-like key exchange protocol from both LWE and RLWE and analyze their asymptotic complexities.

$$\begin{array}{ccc}
 & \text{LWE parameters } q, n, \chi & \\
 & \mathbf{A} \in \mathbb{Z}_q^{n\times n} & \\
\underline{\text{Alice (Server)}} & & \underline{\text{Bob (Client)}} \\
\mathbf{S}, \mathbf{E} \xleftarrow{\chi} \mathbb{Z}_q^{n\times\overline{n}} & & \\
\mathbf{B} := \mathbf{AS} + \mathbf{E} & & \\
 & \xrightarrow{\mathbf{B} \in \mathbb{Z}_q^{n\times\overline{n}}} & \\
 & & \mathbf{S}', \mathbf{E}' \xleftarrow{\chi} \mathbb{Z}_q^{\overline{m}\times n} \\
 & & \mathbf{B}' := \mathbf{S}'\mathbf{A} + \mathbf{E}' \\
 & & \mathbf{E}'' \xleftarrow{\chi} \mathbb{Z}_q^{\overline{m}\times\overline{n}} \\
 & & \mathbf{V} := \mathbf{S}'\mathbf{B} + \mathbf{E}'' \\
 & & \mathbf{C} := \langle \mathbf{V} \rangle_{2^B} \\
 & \xleftarrow{\substack{\mathbf{B}' \in \mathbb{Z}_q^{\overline{m}\times n} \\ \mathbf{C} \in \mathbb{Z}_2^{\overline{m}\times\overline{n}}}} & \\
K := \mathsf{rec}(\mathbf{B}'\mathbf{S}, \mathbf{C}) & & K := \lfloor \mathbf{V} \rceil_{2^B}
\end{array}$$

Figure I. The underlying (DH-like) LWE-based key exchange protocol.

Our protocol is similar to their LWE protocol but we incorporate (and extend) the reconciliation technique in [47] and further modify the protocol to conserve bandwidth.

Lattices also give us trapdoor functions whose security can be reduced to that of worst-case problems in lattices [27] giving us signature schemes [27, 39]. However, the most efficient signatures from lattices come from hash-and-sign constructions [37, 38, 24]. With a broader post-quantum cryptography context in mind, hash-based signatures [11] are very competitive and outperform lattice-based signatures.

## III. New Key Exchange Protocol

### A. The new key exchange protocol from LWE

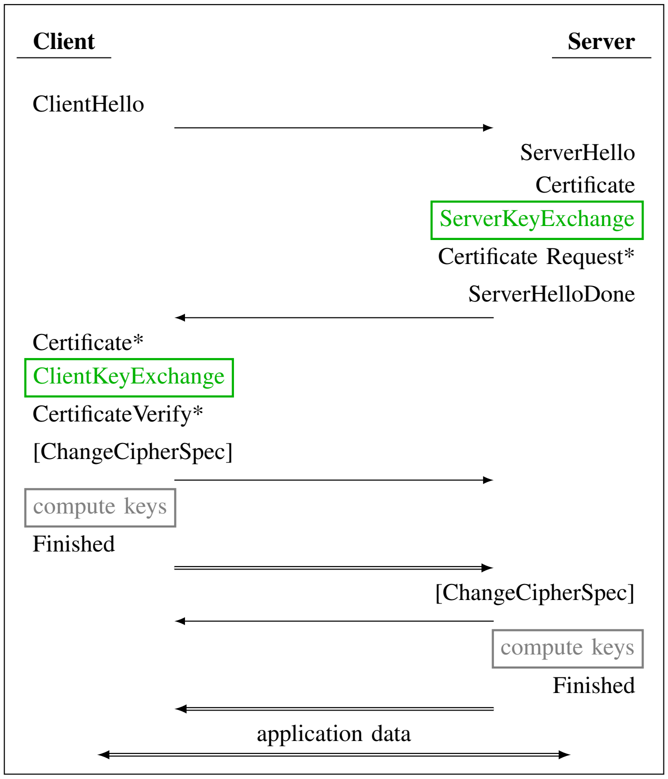
In this section we first describe an unauthenticated key exchange protocol based on the LWE problem (see Figure I). We then sketch the Transport Layer Security (TLS) protocol and explain how the new key exchange protocol fits into the message flow of TLS.

The LWE problem is characterized by three parameters: the modulus $q$, the dimension of the matrix $n$, and the distribution $\chi$. We let $\chi$ be a discrete Gaussian distribution on $\mathbb{Z}$ with center zero, standard deviation $\sigma$ and with a probability density function proportional to $\exp\left(\frac{-x^2}{2\sigma^2}\right)$, where $\sigma = \frac{\alpha q}{\sqrt{2\pi}}$ for $\alpha \ll 1$. We denote by $x \xleftarrow{\chi} X$ the process of sampling a value $x$ from the set $X$ according to the distribution $\chi$. We also sometimes denote by $x \leftarrow \mathcal{D}_{X,\alpha q}$ the process of sampling $x \xleftarrow{\chi} X$. LWE parameters $q, n, \sigma$ determine the security level of the protocol, their selection is described in §III-D.

In addition to the dimensionality of the LWE problem, the key exchange protocol depends on several more parameters: $\overline{n}$, $\overline{m}$, $B$ and $C$.

Matrix $\mathbf{A}$ will be generated at random once per the choice of the protocol parameters. In real-world implementations it will come as part of the encryption standard (similar to how elliptic curves are defined) and will be generated in a certifiably nothing-up-my-sleeves manner (e.g., as an output of the hash function).

[2] We plan to open-source our code shortly.



* indicates optional messages, single lines (→) denote unencrypted communication, double lines (⇒) denote encrypted and authenticated communication, rectangles highlight messages or procedures that will be different for LWE ciphersuite.

Figure II. TLS protocol message flow.

```
ServerKeyExchange:
1: $\mathbf{S}, \mathbf{E} \xleftarrow{\chi} \mathbb{Z}_q^{n \times \overline{n}}$
2: $\mathbf{B} := \mathbf{A}\mathbf{S} + \mathbf{E}$
3: return $\lfloor \mathbf{B} \rceil_{2^{\bar{C}}}$

ClientKeyExchange:
1: $\mathbf{S}', \mathbf{E}' \xleftarrow{\chi} \mathbb{Z}_q^{\overline{m} \times n}$
2: $\mathbf{B}' := \mathbf{S}'\mathbf{A} + \mathbf{E}'$
3: $\mathbf{E}'' \xleftarrow{\chi} \mathbb{Z}_q^{\overline{m} \times \overline{n}}, \mathbf{N} \xleftarrow{u} \left[-2^{C-1}, 2^{C-1}\right)^{n \times \overline{n}}$
4: $\mathbf{V} := \mathbf{S}'\left(2^{\mathbb{C}} \cdot \lfloor \mathbf{B} \rceil_{2^{\bar{C}}} + \mathbf{N}\right) + \mathbf{E}''$
5: $\mathbf{C} := \langle \mathbf{V} \rangle_{2^B}$
6: return $\lfloor \mathbf{B}' \rceil_{2^{\bar{C}}}, \mathbf{C}$

Client compute key:
1: $\mathsf{pms} \leftarrow \lfloor \mathbf{V} \rceil_{2^B}$
2: Compute master secret $\mathsf{ms}$ from $\mathsf{pms}$ as described in TLS specification [1, §8.1].

Server compute key:
1: $\mathsf{pms} \leftarrow \mathsf{rec}(\mathbf{B}'\mathbf{S}, \mathbf{C})$
2: Compute master secret $\mathsf{ms}$ from $\mathsf{pms}$ as described in TLS specification [1, §8.1].
```

Figure III. Messages and procedures in the TLS protocol described on Figure II that are different for the LWE ciphersuite.

The dimensions $\overline{n}$ and $\overline{m}$ should be chosen so that the resulting binary vector $K$ has at least $\lambda$ bits, where $\lambda$ is the security parameter. E.g., for the 128-bit security level, if we setup the reconciliation mechanism to extract $B$ bits from a single ring element, then it should be the case that $\overline{n} \cdot \overline{m} \cdot B \geq 128$. By changing the ratio between $\overline{n}$ and $\overline{m}$ we can trade the client's amount of uploaded data for server's computational load. This could be useful in mobile devices, where energy efficiency of uploads is at least half that of downstream traffic [50, 31].

The high-level view of the TLS protocol [1] appears in Figure II. The handshake part of the protocol that we primarily focus on aims at creating a pre-master-secret known to both communicating parties and not to attackers. The pre-master secret is then used to generate the encryption and MAC keys, as well as the Finished messages which allow the parties to verify that they both got the correct pre-master-secret (see [1, §8.1, §7.4.9 and §6.3]).

The LWE key exchange values are inserted into ServerKeyExchange and ClientKeyExchange messages, the computation of these messages as well as the pre-master-key derivation is shown on Figure III.

As lattice-based cryptography is undergoing period of intense development and scrutiny, break-through advances in cryptanalysis are possible and indeed likely. Consequently, a conservative approach to deployment of lattice-based cryptography is to pair it with legacy schemes, such as factoring- or EC-based. Since the message flow of our proposed solution is identical to the existing ECDHE key exchange protocol, we anticipate that the two can be run concurrently as part of the same "hybrid" ciphersuite, with outputs mixed in into the pre-master-secret via a KDF (similarly to Bos *et al.* [14]).

### B. A generalized reconciliation mechanism

Reconciliation mechanism allows a sender and a recipient to derive the same uniform random secret from the secret values that they have approximately agreed on. A first such mechanism tailored to lattice-based cryptography is due by Peikert [47] and is capable of extracting a single uniform secret bit per ring element. We introduce a generalized version of this mechanism that allows to extract more secret bits at the cost of the increase in probability for the sender and the recipient to disagree on the resulting vector of secret bits.

Throughout this section we will focus on the case of where the modulus $q$ is a power of 2. Our approach can be generalized to the case of an arbitrary modulus using techniques described in [47].

The reconciliation mechanism has parameter $B$—the number of bits per ring elements that the parties extract. For notational convenience we define $\bar{B} = (\log_2 q) - B$.

| Bits extracted from one ring element ($B$) | Upper bound on the probability of key exchange failure |
|---:|---:|
| 16 | $10^{-275}$ |
| 17 | $10^{-69}$ |
| 18 | $10^{-17.5}$ |
| 19 | $10^{-4.38}$ |
| 20 | $\approx 13\%$ |
| 21 | close to 1 |

Figure IV. The upper bound on the probability of LWE key exchange failure.

We define three functions—the $B$ most significant bits of $v$, the difference between $v$ and its closest multiple of $2^{\bar{B}}$, and the sign of this difference:

$$\lfloor\cdot\rceil_{2^B} : \mathbb{Z}_q \to \{0, 1, \ldots, 2^B - 1\} \text{ s.t. } \lfloor v\rceil_{2^B} = \left\lfloor 2^{-\bar{B}} v \right\rceil (\text{mod } 2^B),$$

$$\{\cdot\}_{2^B} : \mathbb{Z}_q \to \{-2^{\bar{B}-1}, \ldots, 2^{\bar{B}-1} - 1\} \text{ s.t. } \{v\}_{2^B} = v - 2^{\bar{B}} \lfloor v\rceil_{2^B},$$

$$\langle\cdot\rangle_{2^B} : \mathbb{Z}_q \to \{0, 1\} \text{ s.t. } \langle v\rangle_{2^B} = \begin{cases} 0 & \text{if } \{v\}_{2^B} < 0 \\ 1 & \text{otherwise} \end{cases}.$$

The following two claims demonstrate that releasing $\langle v\rangle_{2^B}$ does not condition $\lfloor v\rceil_{2^B}$, but it can serve as a hint for the two parties having two sufficiently close numbers $w$ and $v$ and trying to agree on $B$ bits.

**Claim III.1.** *If $v \in \mathbb{Z}_q$ is uniformly random, then $\lfloor v\rceil_{2^B}$ is uniformly random given $\langle v\rangle_{2^B}$.*

The reconciliation function used by the protocol is defined as follows:

$$\mathsf{rec}(w, b) := \lfloor v\rceil_{2^B} \text{ where } v \text{ is closest to } w \text{ s.t. } \langle v\rangle_{2^B} = b.$$

**Claim III.2.** *If $|v - w| < 2^{\bar{B}-2}$, then $\mathsf{rec}(w, \langle v\rangle_{2^B}) = \lfloor v\rceil_{2^B}$.*

For correctness we need both parties to exactly agree on the key $K$, which they can do as long as the noise across all ring elements of $\mathbf{V}$ is bounded according to Claim III.2.

For our choice of parameters, i.e., $q = 2^{32}$, $n = 1024$, $\overline{n} \cdot \overline{m} = 9$, $\sigma = 8/\sqrt{2\pi}$, and $B$ varying between 16 and 21 bits, the probability of key exchange failure can be estimated and is computed in Table IV. (These estimates are conservative since the values can be reconciled even if the noise exceeds the bound of Claim III.2.)

Due to failures in internet connections the noticable percentage of key exchanges fail to complete, that is why we can allow our handshake to fail with probability that is small enough compared to the probability for key exchange to fail due to various other reasons (broken network connection, overloaded front-end server, etc).

### C. Round-and-truncate

To conserve bandwidth we observe that the lower-order bits of $\mathbf{B}$ and $\mathbf{B}'$ exchanged by the parties have vanishingly small influence on the negotiated key. Prompted by this observation, we describe a version of the protocol where entries of $\mathbf{B}$ and $\mathbf{B}'$ are rounded to multiples of $2^C$, and the lower $C$ bits, which are now zeros, are never transmitted.

It is instructive to compare the round-and-truncate technique with two closely related, but different approaches from lattice-based crypto literature: learning-with-rounding (LWR) and modulus switching.

Rounding of scalar products was introduced by Banerjee *et al.* [9] as an alternative to the additive Gaussian noise. Even taking into account refined analyses by Alwen *et al.* [6] and Bogdanov *et al.* [13], LWR in the regime of our parameters (specifically, $n \gg 2^C/\sigma$) is not reducible to the hardness of LWE. Consequently, we still add Gaussian noise (matrices $\mathbf{E}$ and $\mathbf{E}'$ in Figure III), and rely on LWE in our proof of security; rounding is used exclusively for improved efficiency rather than as a security device.

Modulus switching [18, 15] is a technique that scales the modulus in proportion with accumulated noise. We follow similar intuition but use a different modulus only for bandwidth reduction.

Since $\mathbf{B}$ and $\mathbf{B}'$ are now transmitted with lower accuracy, this introduces another source of error in the reconciliation process. Aditionally, in $\mathsf{ClientKeyExchange}$, we sample uniform "lower order bits noise" ($\mathbf{N}$ in Figure III) that we add back in Step 4. This ensure that truncation does not affect the proof of security (see Theorem IV.1). Reworking analysis of §III-B, we have a new bandwidth-reliability tradeoff, summarized in the following table:

| | $B$ | | | | |
|---:|---:|---:|---:|---:|---:|
| | 14 | 15 | 16 | 17 | 18 |
| $C = 4$ | $10^{-1419}$ | $10^{-356}$ | $10^{-89.4}$ | $10^{-22.6}$ | $10^{-5.69}$ |
| 5 | $10^{-469}$ | $10^{-118}$ | $10^{-29.8}$ | $10^{-7.52}$ | $1.84\%$ |
| 6 | $10^{-128}$ | $10^{-32.3}$ | $10^{-8.18}$ | $1.22\%$ | $\approx 1$ |
| 7 | $10^{-33}$ | $10^{-8.36}$ | $1.09\%$ | $\approx 1$ | $\approx 1$ |
| 8 | $10^{-8.4}$ | $1.06\%$ | $\approx 1$ | $\approx 1$ | $\approx 1$ |

Figure V. The upper bound on the probability of LWE key exchange failure as a function of the extraction ($B$) and truncation ($C$) parameters.

### D. Parameters selection

The LWE problem is characterized by three parameters: the modulus $q$, the dimension of the matrix $n$, and the standard deviation of the Gaussian distribution for generating secret random matrices $\sigma$. Along the lines of [14] we chose the following parameters: $q = 2^{32}$, $n = 1024$, $\sigma = 8/\sqrt{2\pi} \approx 3.192$. Albrecht *et al.* [5] studied various methods for solving LWE and published a script that gives estimates of the running time of various classical algorithms, taking as input a triplet $(q, n, \sigma)$. Their study shows that for our choice of parameters the best algorithm for solving LWE requires $2^{153}$ operations, which suffices for 128-bits security level (the choice of 128-bit security level is explained in §V-A). The runtime of the best known quantum attack is less clear as the Grover's algorithm does not necessarily halves the security level.

To reproduce the result, run the sage script (https://bitbucket.org/malb/lwe-estimator) with the code below:

```
load("https://bitbucket.org/malb/lwe-estimator/
      raw/HEAD/estimator.py")
n, alpha, q = 1024, alphaf(8,2^32), 2^32
set_verbose(1)
_ = estimate_lwe(n, alpha, q, skip=["arora-gb"])
```

## IV. Proof of Security

The security of the key exchange protocol can be reduced to the Learning With Errors problem.

**Definition IV.1** (LWE [49])**.** For parameters $n$, $q$, $\alpha$, distribution $\chi$ over $\mathbb{Z}$ as described in §III-A, and vector $\mathbf{s} \in \mathbb{Z}_q^n$ define an LWE oracle $\mathcal{O}_{\mathbf{s},\chi}$ as follows: on receiving a query, it samples $\mathbf{a} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \xleftarrow{\chi} \mathbb{Z}^n$, and outputs $(\mathbf{a}, \mathbf{a}^t\mathbf{s} + \mathbf{e} \bmod (q))$. Define the "Random" oracle $\mathcal{U}$ as returning uniformly sampled $\mathbf{a}$ and $\mathbf{u}$ in $\mathbb{Z}_q^n$ on each query. The LWE assumption states that no efficient adversary $\mathcal{A}$ has non-negligible advantage

$$\begin{aligned}\mathsf{Adv}^{\mathsf{LWE}}_{n,q,\chi}(\mathcal{A}) := \big|\Pr\left[\mathbf{s} \leftarrow \mathbb{Z}_q^n \colon \mathcal{A}^{\mathcal{O}_{\mathbf{s},\chi}}(1^n) = 1\right] \\ -\Pr\left[\mathcal{A}^{\mathcal{U}}(1^n) = 1\right]\big| .\end{aligned}$$

The proof of security of our key exchange protocol relies on a variant of the LWE problem stated below, where secrets $\mathbf{s}$ are drawn from $\chi$. It was shown by Applebaum *et al.* [7] that this variant of LWE has a tight reduction to the LWE problem.

**Lemma IV.1** (Short LWE [7])**.** *For parameters $n$, $q$, $\alpha$, $m = \mathsf{poly}(n)$, and distribution $\chi$, any adversary $\mathcal{A}$ with $\mathsf{Adv}^{\mathsf{SLWE}}_{n,m,q,\chi}(\mathcal{A})$ in distinguishing the following two distributions:*

1) $\mathbf{A} \leftarrow \mathbb{Z}_q^{m\times n}$, $\mathbf{s}, \mathbf{e} \xleftarrow{\chi} \mathbb{Z}^n$, *output* $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$.
2) $\mathbf{A} \leftarrow \mathbb{Z}_q^{m\times n}$, $\mathbf{u} \leftarrow \mathbb{Z}_q^m$ *output* $(\mathbf{A}, \mathbf{u})$,

*can be used to construct an algorithm $\mathcal{B}$ with $\mathsf{Adv}^{\mathsf{LWE}}_{n,q,\chi}(\mathcal{B}) = \mathsf{Adv}^{\mathsf{SLWE}}_{n,m,q,\chi}(\mathcal{A})$ running in roughly the same time as $\mathcal{A}$ and requiring $O(n^2 + m)$ oracle calls to the LWE-or-Random oracle. Therefore, under the LWE assumption, there is no efficient adversary $\mathcal{A}$ distinguishing between distributions 1 and 2 above with non-negligible advantage.*

We also consider the instantiation of the Short LWE problem where the decision oracle outputs $(\mathbf{A}, \mathbf{D})$ where $\mathbf{D} \in \mathbb{Z}_q^{n\times\bar{n}}$ is computed with $\bar{n}$ independent samples of $\mathbf{s}, \mathbf{e}$ or $\mathbf{u}$ respectively. A straightforward hybrid argument for the same set of parameters shows us that any adversary distinguishing these two distributions with advantage $\varepsilon$ can be used to construct an efficient adversary breaking SLWE (and hence LWE) with advantage $\geq \varepsilon/\bar{n}$.

To prove security of the key exchange protocol, consider an LWE key-exchange adversary that tries to distinguish the key $K$ (from Figure I) from a uniformly random key given the transcript of the key exchange protocol. More formally, we define the advantage of such an adversary $\mathcal{A}$ as:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{LWEkex}}_{n,q,\chi}(\mathcal{A}) := |\Pr\left[\mathcal{A}\left(\mathbf{A}, \mathbf{B}, \mathbf{B}', \mathbf{C}, K\right) = 1\right] \\ -\Pr\left[\mathcal{A}\left(\mathbf{A}, \mathbf{B}, \mathbf{B}', \mathbf{C}, K'\right) = 1\right]| ,\end{aligned}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $K$ are as in Figure I with parameters $n$, $q$, and $\alpha$, and $K'$ is a uniform bit string of length $|K|$. The following theorem implies that under the Short LWE assumption (and hence the LWE assumption) for parameters $n$, $q$, and $\alpha$, efficient adversaries have negligible advantage against the key exchange protocol of Section III. Once we show this, dropping the protocol into TLS should satisfy the security notions required of authenticated and confidential channel establishment (ACCE) [32] along the lines of [14, Thm. 2]—although, we do note that this would require some careful analysis.[3]

**Theorem IV.1.** *For parameters $n$, $q$, and $\alpha$, there exists a small explicit constant $\nu$ such that any adversary $\mathcal{A}$ with advantage $\mathsf{Adv}^{\mathsf{LWEkex}}_{n,q,\chi}(\mathcal{A})$ attacking the key exchange protocol can be used to construct an algorithm $\mathcal{B}$ running in time roughly the same as $\mathcal{A}$ such that:*

$$\mathsf{Adv}^{\mathsf{LWE}}_{n,q,\chi}(\mathcal{B}) \geq \frac{1}{\nu} \cdot \mathsf{Adv}^{\mathsf{LWEkex}}_{n,q,\chi}(\mathcal{A}).$$

The constant $\nu$ is bounded by $4\bar{m} \cdot \bar{n}$ where $\bar{m} \cdot \bar{n}$ comes from hybrid arguments across rows of $\mathbf{S}'$ and columns of $\mathbf{S}$ respectively and for practical purposes, we have $\nu \leq 36$.

The security of the key exchange protocol follows from indistinguishability of the Short LWE distributions and then applying Theorem IV.1. The proof is deferred to the full version and we note that it extends to the protocol with truncation requiring only a slight modification in the $\mathsf{ClientKeyExchange}$ step (see steps 3 and 4). This is to ensure that in the proof the view of an adversary derived from truncated components matches its view in the untruncated protocol.

## V. Evaluation

### A. Experimental setup

For this round of experiments we have chosen to focus on the 128-bit security level for direct comparison with prevalent non-quantum-resistant ciphersuites and the available RLWE implementation [14]. (In the full version of the paper we will have a more comprehensive evaluation that includes results for AES-256, which is an appropriate choice against a quantum adversary.)

We integrated four ciphersuites into OpenSSL:

- `LWE_RSA_WITH_AES_128_GCM_SHA256`
- `LWE_ECDSA_WITH_AES_128_GCM_SHA256`
- `LWE_ECDHE_RSA_WITH_AES_128_GCM_SHA256`
- `LWE_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`

In the full version of the paper, we will include a more comprehensive evaluation of the scheme including:

- Benchmarks with 256-bit symmetric primitives (such as AES-256 instead of AES-128 for encryption and generating randomness) which are more appropriate choices against a quantum adversary at a 128-bit security level.
- Benchmarks for the latter two hybrid ciphersuites noting that ECDHE will affect computation overhead slightly and only minimally impact bandwidth which will be dominated by lattice-based components.

### B. Optimizations

We propose a number of protocol- and implementation-level optimizations that are essential for achieving competitive performance of the new LWE-based ciphersuites.

The new reconciliation mechanism from Section III-B drives down both bandwidth and computation costs by extracting more random bits from a single ring element. Compared to the previous

[3] The proof follows a series of hybrids one step of which is replacing the premaster secret key (derived from $K$) with an independent and uniformly sampled one in a black-box manner. The indistinguishability of these two hybrids is what follows from Theorem IV.1 and the LWE assumption.

reconciliation mechanism of Peikert [47] extracting a single bit per element, we extract 16 bits, which reduces the total amount of communication and computation by a factor of 4.

The novel round-and-truncate technique (Section III-C) saves bandwidth by truncating lower-order bits of the transmitted vectors. These savings must be balanced with an increased probability of protocol failure but can compress handshake by more than 18% even when the probability of failure is negligible.

The computation cost for the LWE-based protocol is split comparably between matrix-vector multiplication over 32-bit integers and noise sampling routines. In turn, matrix algebra operations are mostly memory-bound, and careful arrangement of memory accesses is necessary for optimized performance. Noise sampling is dominated by the cost of running PRNG. Hardware acceleration and algorithmic improvements are possible avenues for improvement.

### C. Results

We compared the communication and computation of two ciphers: our new ciphersuite `LWE_RSA_WITH_AES_128_GCM_SHA256`, the ring version of the cipher `RLWE_RSA_WITH_AES_128_GCM_SHA256` that was developed by Microsoft [14] and an ECDHE analog `ECDHE_RSA_WITH_AES_128_GCM_SHA256` with nistp256 curve. We ran micro-benchmarking experiments on 3GHz Intel Core i7 processor, 16GB, 1600MHz DDR3 laptop computer running OS X Yosemite laptop computer.

| Key Exchange | Server/Client keygen | Server/Client shared | Total |
|---|---|---|---|
| `ECDHE` | 0.7ms | 0.6ms | 2ms |
| `RLWE` | 0.81ms | 0.17ms / 0.51ms | 2.13ms |
| `LWE` | 2.56ms | 0.01ms | 5.13ms |

The "Server/Client keygen" column shows the time for computing Server/Client-KeyExchange messages, the column "Server/-Client shared" shows the time for computing the pre-master-secret, the column "Total" shows the maximum sequential time taken by the computation (Server keygen + Client keygen + max{Server shared, Client shared}). The LWE-based ciphersuite is **2.4**× slower compared to RLWE and 2.6× slower compared to ECDHE-based ciphersuite. The overhead of several milliseconds could be viewed as affordable if compared to the average load time of a web page: 4.9 seconds [30].

The difference in communication for the three ciphers that we worked with shows up only in ServerKeyExchange and ClientKeyExchange messages:

| Key Exch. | Server Key Exch. | Client Key Exch. |
|---|---|---|
| `ECDHE` | 205 B | 70 B |
| `RLWE` | 4 KiB | 4 KiB |
| `LWE` | 12 KiB | 12 KiB |
| `LWE` w/truncation | 10 KiB | 10 KiB |

To evaluate the LWE key exchange with truncation mechanism described in Section III-C we set $B = 15$ and $C = 6$, which results in failure probability of $10^{-32.3}$.

The LWE-based protocol has 2.5× bandwidth overhead compared to RLWE. However, in real world applications, the communication during the full handshake is often outweighed by the certificates that get transferred from the server to the client.[4] Therefore, a more realistic comparison of LWE puts it between **1.7–2.1**× more bandwidth intensive than RLWE and between **2.9–6.4**× as compared to ECDHE. We also note that it is instructive to compare the communication overheads to the size of the average web page which is typically around 320 KB [30].

## Acknowledgment

The authors would like to thank Adam Langley, Eric Grosse, and Úlfar Erlingsson for their inputs. This work was done when the first author was an intern with the Google Security and Privacy Research team.

## References

[1] "The transport layer security (TLS) protocol version 1.2," https://tools.ietf.org/html/rfc5246.

[2] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (H)IBE in the standard model," in *Advances in Cryptology–EUROCRYPT 2010*. Springer, 2010, pp. 553–572.

[3] ——, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE," in *Advances in Cryptology–CRYPTO 2010*. Springer, 2010, pp. 98–115.

[4] M. Ajtai, "Generating hard instances of lattice problems," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM, 1996, pp. 99–108.

[5] M. R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of learning with errors," Cryptology ePrint Archive, Report 2015/046, 2015. http:/eprint. iacr. org/2015/046, Tech. Rep., 2015.

[6] J. Alwen, S. Krenn, K. Pietrzak, and D. Wichs, "Learning with rounding, revisited - new reduction, properties and applications," in *Advances in Cryptology—CRYPTO 2013*, 2013, pp. 57–74. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40041-4_4

[7] B. Applebaum, D. Cash, C. Peikert, and A. Sahai, "Fast cryptographic primitives and circular-secure encryption based on hard learning problems," in *Advances in Cryptology-CRYPTO 2009*. Springer, 2009, pp. 595–618.

[8] D. Augot, L. Batina, D. J. Bernstein, J. Bos, J. Buchmann, W. Castryck, O. Dunkelman, T. Güneysu, S. Gueron, A. Hülsing, T. Lange, M. S. E. Mohamed, C. Rechberger, P. Schwabe, N. Sendrier, F. Vercauteren, and B.-Y. Yang, "Initial recommendations of long-term secure post-quantum systems," 2015, http://pqcrypto.eu.org/docs/initial-recommendations.pdf.

[9] A. Banerjee, C. Peikert, and A. Rosen, "Pseudorandom functions and lattices," in *Advances in Cryptology—EUROCRYPT 2012*, 2012, pp. 719–737. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-29011-4_42

[10] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "Recommendation for key management – part 1: General (rev 3)," 2012, available at http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf.

[11] D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. Schneider, P. Schwabe, and Z. Wilcox-O'Hearn, "SPHINCS: practical stateless hash-based signatures," in *Advances in Cryptology–EUROCRYPT 2015*. Springer, 2015, pp. 368–397.

[12] Business Insider, http://www.businessinsider.com/googles-quantum-computing-milestone-2015-3.

[13] A. Bogdanov, S. Guo, D. Masny, S. Richelson, and A. Rosen, "On the hardness of learning with rounding over small modulus," *IACR Cryptology ePrint Archive*, vol. 2015, p. 769, 2015. [Online]. Available: http://eprint.iacr.org/2015/769

[14] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila, "Post-quantum key exchange for the TLS protocol from the ring learning with errors problem," in *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, 2015, pp. 553–570.

[15] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, 2012, pp. 309–325.

[16] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, "Classical hardness of learning with errors," in *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. ACM, 2013, pp. 575–584.

[4] Often the certificate contains a long chain to the root CA or even multiple chains to different CAs. For example the size of the certificate for https://www.google.com is 3KiB, for https://www.washingtonpost.com – 10KiB.

[17] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from Ring-LWE and security for key dependent messages," in *Advances in Cryptology–CRYPTO 2011*. Springer, 2011, pp. 505–524.

[18] ——, "Efficient fully homomorphic encryption from (standard) LWE," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 831–871, 2014.

[19] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," *Journal of cryptology*, vol. 25, no. 4, pp. 601–639, 2012.

[20] Y. Chen and P. Q. Nguyen, "BKZ 2.0: Better lattice security estimates," in *Advances in Cryptology–ASIACRYPT 2011*. Springer, 2011, pp. 1–20.

[21] R. Cramer, L. Ducas, C. Peikert, and O. Regev, "Recovering short generators of principal ideals in cyclotomic rings," Cryptology ePrint Archive, Report 2015/313, 2015, http://eprint.iacr.org/.

[22] N. S. Dattani and N. Bryans, "Quantum factorization of 56153 with only 4 qubits," *ArXiv e-prints*, 2014.

[23] W. Diffie, P. C. Van Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes and cryptography*, vol. 2, no. 2, pp. 107–125, 1992.

[24] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky, "Lattice signatures and bimodal Gaussians," in *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 40–56.

[25] DWave, http://www.dwavesys.com/blog/2015/08/announcing-d-wave-2x-quantum-computer.

[26] Y. Elias, K. E. Lauter, E. Ozman, and K. E. Stange, "Provably weak instances of Ring-LWE," 2015, available at http://eprint.iacr.org/.

[27] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*. ACM, 2008, pp. 197–206.

[28] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM, 1996, pp. 212–219.

[29] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *Algorithmic number theory*. Springer, 1998, pp. 267–288.

[30] U. Hölzle, "Speed matters," 2010, O'Reilly Velocity Conference.

[31] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4G LTE networks," in *The 10th International Conference on Mobile Systems, Applications, and Services, MobiSys'12*, 2012, pp. 225–238. [Online]. Available: http://doi.acm.org/10.1145/2307636.2307658

[32] T. Jager, F. Kohlar, S. Schäge, and J. Schwenk, "On the security of TLS-DHE in the standard model," in *Advances in Cryptology–CRYPTO 2012*. Springer, 2012, pp. 273–293.

[33] X. L. Jintai Ding, Xiang Xie, "A simple provably secure key exchange scheme based on the learning with errors problem," Cryptology ePrint Archive, Report 2012/688, 2012, http://eprint.iacr.org/.

[34] T. Laarhoven, "Sieving for shortest vectors in lattices using angular locality-sensitive hashing," Tech. Rep., 2015, available at http://eprint.iacr.org/.

[35] R. Lindner and C. Peikert, "Better key sizes (and attacks) for LWE-based encryption," in *Topics in Cryptology–CT-RSA 2011*. Springer, 2011, pp. 319–339.

[36] E. Lucero, R. Barends, Y. Chen, J. Kelly, M. Mariantoni, A. Megrant, P. O'Malley, D. Sank, A. Vainsencher, J. Wenner *et al.*, "Computing prime factors with a Josephson phase qubit quantum processor," *Nature Physics*, vol. 8, no. 10, pp. 719–723, 2012.

[37] V. Lyubashevsky, "Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures," in *Advances in Cryptology–ASIACRYPT 2009*. Springer, 2009, pp. 598–616.

[38] ——, "Lattice signatures without trapdoors," in *Advances in Cryptology–EUROCRYPT 2012*. Springer, 2012, pp. 738–755.

[39] V. Lyubashevsky and D. Micciancio, "Asymptotically efficient lattice-based digital signatures," in *Theory of Cryptography*. Springer, 2008, pp. 37–54.

[40] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *Journal of the ACM (JACM)*, vol. 60, no. 6, p. 43, 2013.

[41] D. Micciancio and C. Peikert, "Hardness of SIS and LWE with small parameters," in *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 21–39.

[42] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*. ACM, 2011, pp. 113–124.

[43] NIST, http://www.nist.gov/itl/csd/ct/post-quantum-crypto-workshop-2015.cfm.

[44] NIST Suite B, https://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml.

[45] New York Times, http://www.nytimes.com/2014/06/24/technology/microsoft-makes-a-bet-on-quantum-computing-research.html?partner=rss&emc=rss&_r=0.

[46] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM, 2009, pp. 333–342.

[47] ——, "Lattice cryptography for the internet," in *Post-Quantum Cryptography*. Springer, 2014, pp. 197–219.

[48] ——, "A decade of lattice cryptography," Cryptology ePrint Archive, Report 2015/939, 2015, http://eprint.iacr.org/.

[49] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, p. 34, 2009.

[50] N. Thiagarajan, G. Aggarwal, A. Nicoara, D. Boneh, and J. P. Singh, "Who killed my battery?: analyzing mobile browser energy consumption," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 41–50.

[51] L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature*, vol. 414, no. 6866, pp. 883–887, 2001.

[52] Washington Post, https://www.washingtonpost.com/world/national-security/nsa-seeks-to-build-quantum-computer-that-could-crack-most-types-of-encryption/2014/01/02/8fff297e-7195-11e3-8def-a33011492df2_story.html.

[53] Wall Street Journal, http://www.wsj.com/articles/intel-to-invest-50-million-in-quantum-computers-1441307006.