Предложение к стандартизации пост-квантовой цифровой подписи: схема “Крыжовник”

Елена Киршанова, Никита Колесников, Екатерина Малыгина, and Семён Новосёлов

I. Kant Baltic Federal University, Kaliningrad, Russia
{EKirshanova, NiKolesnikov1, EMalygina, SNovoselov}@kantiana.ru

Аннотация В этой статье мы предлагаем цифровую подпись, безопасность которой основана на алгебраических решетках. Дизайн схемы следует из парадигмы Фиата-Шамира. Мы доказываем безопасность схемы в квантовой модели безопасности и описываем конкретные параметры схемы, предлагаем три различных уровня безопасности. Благодаря модульной структуре решеток, уровень безопасности легко изменить в большую или меньшую стороны. Наше предложение может служить основой проекта по стандартизации пост-квантовых примитивов на решётках.

Аннотация In this paper we propose an algebraic lattice-based signature scheme. The design of the proposal follows the Fiat-Shamir paradigm. Our scheme is proved secure in the quantum random oracle model and achieves security against $\mathsf{UF}-\mathsf{sCMA}$ adversaries. We propose three concrete parameter sets to instantiate the scheme different level of security. Thanks to the algebraic structure of the construction, the scheme is flexible in security levels so that we can achieve trade-offs between speed and security. Our proposal may serve as the basis for a standard of lattice-based schemes.

1 Введение

Криптографические примитивы на решётках – одно из самых обещающих направлений современной криптографии не только ввиду стойкости этих примитивов к атакам на квантовом компьютере, но и вследствие большого спектра конструкций (гомоморфное шифрование, электронные голосования, различные типы подписей), а также их надежности к *классическим* атакам. Криптографические конструкции на решетках не только элегантны в теории, но значимы на практике, и поэтому в достаточно скором будущем будут стандартизированы.[1].

В этой статье мы предлагаем схему цифровой подписи, основанную на алгебраических решётках. Предлагаемая конструкция удовлетворяет следующими основным свойствам:

1. безопасность схемы основана на задачах “в среднем”, а именно, на задачах LWR (Learning With Rounding) и SIS (Shortest Integer Solution) – классические трудные задачи на решётках, определения которым даны в Разделе 2),
2. для эффективности схемы мы используем, так называемый, *модульный* вариант задач, а именно, module-LWR, module-SIS [LS15], что позволяет не только уменьшить размеры параметров схемы и время операций, но и даёт возможность легко варьировать уровни безопасности схемы,
3. стойкость схемы доказана в квантовой модели QROM (Quantum Random Oracle Model) для “сильного” атакующего, а именно, для атаки вида $\mathsf{UF}-\mathsf{sCMA}$ (см. Раздел 2),
4. в процессе генерации ключей и подписи вместо нормального распределения мы используем равномерное распределение из интервала, что уменьшает риск сторонних атак,

[1] Пробные версии обмена ключами New Hope были уже тестированы в TLS соединениях для браузера Google Chrome [ADPS16]. Процесс стандартизации пост-квантовых схем доступен по адресу https://csrc.nist.gov/projects/post-quantum-cryptography

5. мы предлагаем конкретный набор параметров схемы с битовой оценкой сложности атак на предложенные параметры (см. Раздел 5).

Представленная здесь схема основана на парадигме Фиата-Шамира [FS87,Lyu09] и по идеологии продолжает серию работ по предложению конкретных схем подписи [BG14,DKL$^+$18,ABB$^+$17]. Основное отличие нашей схемы от ранее предложенных заключается в том, что безопасность ключей основана на задаче LWR (а не на задаче LWE, Learning With Erros). Мы считаем, что наш подход упрощает описание и потенциально ускоряет вычисления.

2 Предварительные сведения

2.1 Обозначения

Будем обозначать $\mathbb{Z}/q\mathbb{Z}-$ кольцо целых по чётному модулю q, результат $z \bmod q$ представляем в интервале $[0, q-1]$. Далее обозначим за R, R_q и R_p кольца многочленов $\mathbb{Z}[x]/(x^n+1)$, $\mathbb{Z}/q\mathbb{Z}[x]/(x^n+1)$ и $\mathbb{Z}/p\mathbb{Z}[x]/(x^n+1)$ соответственно. Вектора будем обозначать жирными строчными буквами (например, $\mathbf{x}$), матрицы – прописными (например, $\mathbf{A}$), константы - обычными строчными. Обозначаем за $\mathbb{I}$ единичную матрицу. Элементы кольца $\mathbb{Z}[x]/(x^n+1)$ будем понимать как вектора-коэффициенты многочленов. Вектора по умолчанию являются векторами-столбцами. Евклидова (или ℓ_2) норма вектора $\mathbf{x}$ определяется как $\|\mathbf{x}\| = \|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$, а ℓ_∞-норма как $\|\mathbf{x}\|_\infty = \max_i |x_i|$.

Многочленам из кольца R мы ставим в соответствие вектора-коэффициенты длины n, поэтому произведение векторов $\mathbf{x}\cdot\mathbf{y}$ надо понимать как произведение соответствующих многочленов. Элементу $\mathbf{a} \in R_q$ ставим в соответствие матрицу $\mathrm{rot}(\mathbf{a}) \in (\mathbb{Z}/q\mathbb{Z})^{n\times n}$, i-ая строка которой – коэффициенты многочлена $x^{i-1}\cdot\mathbf{a}$. Такая матрица задает произведение любого элемента из R_q на многочлен $\mathbf{a}$.

Для конечного множества S запись $s \leftarrow S$ обозначает, что s выбрано в соответствии со случайным равномерным распределением на S. За S_β^ℓ обозначим множество векторов длины ℓ, каждый коэффициент которого взят в соответствии с равномерным распределение из множества $[-\beta, \beta]$.

Для любого $x \in \mathbb{Q}$ запись $\mathsf{Round}(x) \in \mathbb{Z}$ означает взятие ближайшего целого, где 1/2 округляется до 1. Для целого $x \in \mathbb{Z}/q\mathbb{Z}$, в бинарном представлении которого $\log q$ бит, функция $\mathsf{MSB}(x, d)$ (соответственно, $\mathsf{LSB}(x, d)$) означает взятие d наибольших (соответственно, наименьших) бит. Все операции распространяются на вектора и матрицы по-коэффициентно. Будем обозначать за $\mathsf{LSB}(x, d)^\pm-$ d младших бит x, взятых их интервала $(-2^{d-1}, 2^{d-1}]$.

В нашей схеме мы будет использовать два модуля $q = 2^\nu$ и $p = 2^\mu$. "Конвертирование" элемента $x \in \mathbb{Z}/q\mathbb{Z}$ в $x' \in \mathbb{Z}/p\mathbb{Z}$ происходит по правилу $x' = \mathsf{Round}(x \cdot \frac{p}{q})$. Так как наши модули – степени двойки, этот же результат можно получить, добавив к x константу $h = 2^{\nu-\mu-1}$ и взяв μ значимых бит: $x' = \mathsf{MSB}(x+h, \mu)$. Такое представление операции Round использовано, например, в [DKSRV18]. Вектор, каждая координата которого равна h, обозначим за $\mathbf{h}$.

Для всякого целого $w > 0$ положим $B_w = \{\mathbf{x} \in R \,|\, \|\mathbf{x}\|_\infty = 1, \|\mathbf{x}\| = \sqrt{w}\} \subseteq R$.

2.2 Синтаксис и модели безопасности цифровых подписей

Определение 1 *Цифровая подпись – примитив, состоящий из трёх алгоритмов:*

- *вероятностный алгоритм генерации ключевой пары* $\mathsf{KeyGen}(\mathsf{par})$*, возвращающий секретный ключ* sk *и ключ верификации* vk*,*
- *вероятностный алгоритм генерации подписи* $\mathsf{Sign}(\mathsf{sk}, m)$*, который для сообщения* $m \in \mathcal{M}$ *возвращает подпись* σ*,*
- *детерминированный алгоритм* $\mathsf{Verify}(m, \sigma, \mathsf{vk})$*, который возвращает либо "Accept" (подпись* σ *корректна для* (m, vk)*), либо "Reject" (подпись* σ *не корректна для* (m, vk)*).*

Цифровая подпись корректна с долей ошибки ε*, если для всех пар* $(\mathsf{sk}, \mathsf{vk}) \in \mathsf{KeyGen}(\mathsf{par})$ *и всех сообщений* $m \in \mathcal{M}$ *имеем* $\Pr[\mathsf{Verify}(m, \mathsf{Sign}(\mathsf{sk}, m), \mathsf{vk}) = \text{"}Accept''] \geq 1 - \varepsilon$*.*

Для доказательства безопасности схемы подписи нам понадобятся три модели: "слабая" модель $\mathsf{UF}-\mathsf{NMA}$ (unforgeability against no-message attack), в которой атакующий не имеет доступа к подписывающему оракулу; модель$\mathsf{UF}-\mathsf{CMA}$ (unforgeability against chosen-message attack), где атакующий, имея доступ по подписывающему оракулу, должен сформировать подпись к новому сообщению; "сильная" модель $\mathsf{UF}-\mathsf{sCMA}$ (strong unforgeability against chosen-message attack), где атакой считается корректно сформированная подпись сообщения, подпись для которого атакующий уже знает.

2.3 Сложные задачи на решётках

Безопасность нашей подписи основывается на двух "сложных в среднем" задачах. Первая – задача Обучения с Округлением (от англ. Learning with Rounding, LWR) [BPR12] – детерминированная версия задачи Обучения с Ошибками (Learning with Errors, LWE[Reg05]). В основе безопасности ключей подписи лежит трудность этой модульной версии задачи над факторкольцом R_q [LS15]. Все вычисления производятся в факторкольце R_q, матрица $\mathbf{A}$ формируется как блочная матрица из $k \cdot \ell$ элементов из R_q, где каждый блок – матрица $\mathrm{rot}(\mathbf{a})$.

Для нашей схемы, в отличие от классических задач LWR и LWE, где матрица $\mathbf{A}$ берётся случайным образом из $R_q^{k\times\ell}$, мы будем требовать, чтобы хотя бы один из $k \cdot \ell$ многочленов был обратим в R_q. Случайный элемент из R_q для $q = 2^\nu$ будет обратимым с пренебрежимо малой вероятностью, поэтому один из блоков матрицы $\mathbf{A}$ строится не из случайного многочлена, а с помощью многочлена специального вида, обратимость которого следует из следующего факта, доказанного в [Wat07]. Мы полагаем, что специальный вид одного блока матрицы $\mathbf{A}$ не противоречит предположениям сложности.

Факт 1 (Обратимость элемента в фактор-кольце, [Wat07, Theorem 11.1]) *Пусть $\mathbb{Z}/q\mathbb{Z}$ – кольцо. Тогда $f = a_0 + a_1 x + \ldots + a_{n-1}x^{n-1}$ – единица в R_q тогда и только тогда, когда a_0 – обратим в $\mathbb{Z}/q\mathbb{Z}$, а $a_1, \ldots, a_{n-1}$ – нильпотенты в $\mathbb{Z}/q\mathbb{Z}$.*

В кольце $\mathbb{Z}/q\mathbb{Z}$, где $q = 2^\nu$, нильпотентными являются четные элементы, отличные от 0, а обратимыми - нечетные. Пользуясь Фактом 1, мы можем легко построить обратимый многочлен в R_q. Более того, обратимых многочленов в R_q, достаточно много, порядка $2^{\Omega(n)}$.

Определение 2 *Задача обучения с округлением (MLWR)*

Пусть $q \geq p \geq 1$, $k, \ell \geq 1$ – целые числа. MLWR-распределение для вектора $\mathbf{s} \leftarrow R_q^\ell$ есть множество пар вида $\left(\mathbf{A}, \mathsf{Round}\left(\frac{p}{q} \cdot \mathbf{A} \cdot \mathbf{s}\right)\right)$, где $\widetilde{\mathbf{A}} \leftarrow R_q^{k\times\ell}$.

Задача поиска: Для заданного произвольным образом большого числа выборок из MLWR-распределения для вектора $\mathbf{s} \leftarrow R_q^\ell$ восстановить $\mathbf{s}$.

Задача различения распределений: Для заданного произвольным образом большого числа выборок из $\widetilde{R}_q^{k\times\ell} \times R_p^k$ определить, являются ли они равномерно распределенными или же MLWR-распределенными для вектора $\mathbf{s} \leftarrow R_q^\ell$.

Вероятность успеха алгоритма, решающего задачу MLWR *с введёнными выше параметрами, будем обозначать $\mathsf{Adv}_{k,\ell,p,q}^{\mathrm{MLWR}}$.*

Обе версии задачи эквивалентны (то есть, имея оракул, решающий одну задачу, можно решить другую за полиномиальное от n время), [BGM+16]. В доказательстве безопасности схемы подписи нам понадобится вторая версия.

Безопасность подписи основана на Задаче нахождения Короткого Целочисленного Решения (Short Integer Solution Problem, SIS), [Ajt96]. Нам потребуется модульная версия этой задачи.

Определение 3 *Задача нахождения Короткого Целочисленного Решения (*MSIS*)*

Зафиксируем $b \in \mathbb{N}$ и пусть $\mathbf{A} \leftarrow R_q^{k\times\ell}$. Модульная задача нахождения короткого целочисленного решения, параметризованная посредством $b > 0$, заключается в нахождении "короткого" ненулевого прообраза $\mathbf{y} \leftarrow R_q^{k+\ell}$ в решетке, определяемой $\mathbf{A}$, т.е.

$$\mathbf{y} \neq 0, \quad [\mathbb{I}|\mathbf{A}] \cdot \mathbf{y} = 0 \quad u \quad \|\mathbf{y}\|_\infty \leq b.$$

Вероятность успеха алгоритма, решающего задачу MSIS *с введёнными выше параметрами, будем обозначать* $\mathsf{Adv}^{\mathrm{MSIS}}_{k,\ell,q,b}$.

Для доказательства безопасности нашей схемы нам потребуется вариант задачи SIS, так называемый SelfTargetSIS, предложенный в [KLS18]. В этой же работе описана редукция от SIS к SelfTargetSIS.

Определение 4 *Задача SelfTargetSIS*

Пусть $\mathcal{H} : \{0,1\}^* \to B_w$ *– криптографическая хэш-функция. Зададим случайным образом* $\mathbf{A} \leftarrow R_q^{k\times\ell}$ *и доступ к квантовому случайному оракулу* $\mathcal{H}(\cdot)$. *Тогда задача SelfTargetSIS сводится к нахождению*

$$\mathbf{y} = [\mathbf{r}, \mathbf{c}]^T, \quad \text{где} \quad 0 \le \|\mathbf{y}\|_\infty \le \gamma, \quad \mathcal{H}([\mathbf{A}|\mathbb{I}] \cdot \mathbf{y}, M) = \mathbf{c},$$

здесь $M \in \{0,1\}^*$ *– исходное сообщение. Вероятность успеха алгоритма, решающего задачу SelfTargetSIS с введёнными выше параметрами, будем обозначать* $\mathsf{Adv}^{SelfTargetSIS}_{k,\ell,q,b}$.

3 Описание схемы

Цифровая подпись будет зависеть от следующих параметров: $q = 2^\nu$, $p = 2^\mu$, $\nu > \mu$. Мы будем использовать криптографическую хэш-функцию $\mathcal{H} : \{0,1\}^* \to B_w$ (см. [DKL$^+$18] для построения хэш-функции с областью значений B_w). Для стойкости к квантовым атакам [BHMT02], должно выполнятся $|B_w| \ge 2^{256}$, что достигается при $w \ge 60$. Параметры k, ℓ отвечают за размерности ключей. Параметры s, γ задают интервалы для коэффициентов многочленов в процессе генерации ключей или подписи, параметры d, β отвечают за корректность и безопасность схемы. Подпись формируется для сообщений $M \in \{0,1\}^*$. Конкретные значения параметров заданы в Разделе 5

Алгоритм 3.1 Генерация ключей

Вход: $k > \ell > 1$, $q > p$, s
Выход: $\mathbf{A}$, $\mathbf{t}$

1: $\mathbf{A} \leftarrow R_q^{k\times\ell}$
2: $\mathbf{s} \leftarrow S_s^\ell$
3: $\mathbf{t} = \mathsf{Round}\left(\frac{p}{q} \cdot \mathbf{As}\right)$, $\mathbf{t}_0 = \mathsf{LSB}(\mathbf{As} + \mathbf{h}, \nu - \mu)$ ▷ $\mathbf{As} + \mathbf{h} = 2^{\nu-\mu}\mathbf{t} + \mathbf{t}_0, \|\mathbf{t}_0\|_\infty < 2^{\nu-\mu}$
4: **return** $\mathsf{sk} = (\mathbf{s}, \mathbf{t}_0), \mathsf{vk} = (\mathbf{A}, \mathbf{t})$

3.1 Корректность

Докажем корректность нашей схемы. Убедимся в том, что значения $\mathbf{w}$, полученные на шаге 2 Алгоритма 3.2 и шаге 1 Алгоритма 3.3 равны. Это обеспечивается первым условием на шаге 5. Обратимся к алгоритму верификации подписи, в процессе которого мы вычисляем

$$\mathbf{w} = \mathbf{Az} - 2^{\nu-\mu} \cdot \mathbf{t} \cdot \mathbf{c} = \mathbf{A} \cdot (\mathbf{y} + \mathbf{s} \cdot \mathbf{c}) - \mathbf{c} \cdot 2^{\nu-\mu} \cdot \mathsf{Round}\left(\frac{p}{q} \cdot \mathbf{As}\right) = \mathbf{Ay} + \mathbf{Asc} - \mathbf{c} \cdot 2^{\nu-\mu} \cdot \mathsf{Round}\left(\frac{p}{q} \cdot \mathbf{As}\right),$$

Из равенства $\mathsf{Round}\left(\frac{p}{q} \cdot \mathbf{As}\right) = \mathsf{MSB}(\mathbf{As} + \mathbf{h}, \mu)$, где $\mathbf{h}_i = 2^{\nu-\mu-1}$, следует

$$\mathbf{w} = \mathbf{Ay} + \mathbf{Asc} - \mathbf{c} \cdot 2^{\nu-\mu} \cdot \mathsf{MSB}(\mathbf{As} + \mathbf{h}, \mu) = \mathbf{Ay} + \mathbf{Asc} - \mathbf{c} \cdot (\mathbf{As} + \mathbf{h} + \mathsf{LSB}(\mathbf{As} + \mathbf{h}, \nu - \mu)).$$

Заметив, что $\mathbf{t}_0 = \mathsf{LSB}(\mathbf{As} + \mathbf{h}, \nu - \mu)$ окончательно получаем в процессе верификации $\mathbf{w} = \mathbf{Ay} - \mathbf{c} \cdot (\mathbf{h} + \mathbf{t}_0)$. Для вектора "ошибки" $\mathbf{c} \cdot (\mathbf{h} - \mathbf{t}_0)$ справедливо $\|\mathbf{c} \cdot (\mathbf{h} - \mathbf{t}_0)\|_\infty < w \cdot 2^{\nu-\mu+1}$, поскольку $\mathbf{c} \in B_w$ и $\|\mathbf{h} - \mathbf{t}_0\|_\infty < 2^{\nu-\mu+1}$. В процедуре генерации подписи на шаге 5 мы убеждаемся в том, что добавление вектора $\mathbf{c} \cdot (\mathbf{h} - \mathbf{t}_0)$ к $\mathbf{Ay} - \mathbf{c} \cdot (\mathbf{h} - \mathbf{t}_0)$ не изменит магнитуду его коэффициентов, что проверяется функцией $\mathsf{LSB}(,)^{\pm}$. Получаем в итоге $\mathsf{MSB}(\mathbf{Az} - 2^{\nu-\mu} \cdot \mathbf{t} \cdot \mathbf{c}, d) = \mathsf{MSB}(\mathbf{Ay}, d)$.

Алгоритм 3.2 Генерация подписи

Вход:
$q = 2^\nu$, $p = 2^\mu$, $\ell > 1 \in \mathbb{Z}$, $d < \nu \in \mathbb{Z}$, $\beta \in \mathbb{Z}$, $\gamma \in \mathbb{Z}$ – параметры схемы,
$M \in \{0,1\}^*$ – сообщение,
$\mathbf{A} \in R_q^{k \times \ell}$ – часть открытого ключа,
$\mathsf{sk} = (\mathbf{t}, \mathbf{s})$ – секретный ключ,
$\mathcal{H}\{0,1\}^* \to B_w$ – хэш-функция
Выход: $(\mathbf{z}, \mathbf{c})$ – подпись для сообщения M.

1: $\mathbf{y} \leftarrow S^\ell_{\gamma-1}$
2: $\mathbf{w} = \mathbf{A} \cdot \mathbf{y}$
3: $\mathbf{c} = \mathcal{H}\left(\mathsf{MSB}(\mathbf{w}, d), M\right)$
4: $\mathbf{z} = \mathbf{y} + \mathbf{sc}$
5: **if** $(\left\|\mathsf{LSB}(\mathbf{w} - \mathbf{c}(\mathbf{h} - \mathbf{t}_0), \nu - d)^{\pm}\right\|_\infty \geq 2^{\nu-d-1} - w \cdot 2^{\nu-\mu+1})$ **or** $(\|\mathbf{z}\|_\infty \geq \gamma - \beta)$ **then**
6: **restart**
7: **return** $(\mathbf{z}, \mathbf{c})$

Алгоритм 3.3 Проверка подписи

Вход: M, $\mathbf{z}$, $\mathbf{c}$, $\mathbf{A}$, $\mathbf{t}$, d, $\mathcal{H}$, β, γ
Выход: Accept or Reject

1: $\mathbf{w} = \mathbf{Az} - \mathbf{t} \cdot 2^{\nu-\mu} \cdot \mathbf{c}$
2: $\mathbf{c}' = \mathcal{H}(\mathsf{MSB}(\mathbf{w}, d)), M)$
3: **if** $\mathbf{c}' == \mathbf{c}$ and $\|\mathbf{z}\|_\infty \leq \gamma - \beta$ **then**
4: **return** "Accept"
5: **else**
6: **return** "Reject"

3.2 О числе итераций в процедуре Sign

В процессе вычисления подписи Алгоритм 3.2 на шаге 5, проверяет, попадают ли коэффициенты вектора $\mathbf{z}$ в интервал $[-(\gamma - \beta - 1), \gamma - \beta - 1]$. Для фиксированного ключа $\mathbf{s}$ вероятность этого события зависит от $\|\mathbf{y}\|_\infty$, выбранного на шаге 1. Вычислим эту вероятность.

Пусть $\mathbf{z} = \mathbf{y} + \mathbf{v}$ такой, что $\mathbf{z} \in S^\ell_{\gamma-\beta-1}$. Обозначим $\beta = \|\mathbf{cs}\|_\infty$. Так как $\|\mathbf{s}\|_\infty \leq s$ и $\mathbf{c} \in B_w$, то $\beta < ws$. Отсюда $\|\mathbf{v}\|_\infty \leq \beta$. Для каждого коэффициента $\mathbf{v}_i$ вектора $\mathbf{v}$ соответствующий коэффициент $\mathbf{z}_i$ лежит в интервале $[-(\gamma - \beta - 1), \gamma - \beta - 1]$. Поскольку $\mathbf{y} = \mathbf{z} - \mathbf{v}$, то $\mathbf{y} \in S^\ell_{\gamma-1}$ и соответствующий коэффициент $\mathbf{y}_i$ лежит в интервале $[-(\gamma - 1), \gamma - 1]$. Следовательно,

$$p_1 = \Pr_{\mathbf{y} \leftarrow S^\ell_{\gamma-1}}[\|\mathbf{z}\|_\infty < \gamma - \beta] = \frac{\left|S^\ell_{\gamma-\beta-1}\right|}{\left|S^\ell_{\gamma-1}\right|} = \left(\frac{2\gamma - 2\beta - 1}{2\gamma - 1}\right)^{n\cdot\ell} = \left(1 - \frac{\beta}{\gamma - \frac{1}{2}}\right)^{n\cdot\ell} \approx \exp\left(-\frac{\beta n \ell}{\gamma}\right).$$

Алгоритм 3.2 на шаге 5 также проверяет, когда коэффициенты вектора $\mathsf{LSB}(\mathbf{w}, \nu - d)^{\pm}$ не попадают в интервал $\left(-2^{\nu-d-1} + w \cdot 2^{\nu-\mu+1}, 2^{\nu-d-1} - w \cdot 2^{\nu-\mu+1}\right]$, в то время как в общем случае $\mathsf{LSB}(\mathbf{x}, \nu - d)^{\pm} \in \left(-2^{\nu-d-1}, 2^{\nu-d-1}\right]$ Учитывая (эвристически) равномерный характер распределений, получаем

$$p_2 = \Pr_{\mathbf{w} \in S^k_{2^{\nu-d-1}-1}}\left[\left\|\mathsf{LSB}(\mathbf{w} - \mathbf{c}(\mathbf{h} - \mathbf{t}_0), \nu - d)^{\pm}\right\|_\infty < 2^{\nu-d-1} - w \cdot 2^{\nu-\mu+1}\right] = \left(\frac{2^{\nu-d} - w \cdot 2^{\nu-\mu+2} - 1}{2^{\nu-d} - 1}\right)^{n\cdot k} =$$

$$= \left(1 - \frac{w 2^{\nu-\mu+2}}{2^{\nu-d} - 1}\right)^{n\cdot k} \approx \exp\left(-nk \cdot \frac{w 2^{\nu-\mu+2}}{2^{\nu-d} - 1}\right).$$

Таким образом, ожидаемое число повторений функции Sign Алгоритма 3.2

$$\mathbb{E}[\#\text{итераций}] = (p_1 \cdot p_2)^{-1} \tag{1}$$

4 Доказательство безопасности

Доказательство безопасности нашей схемы – адаптация доказательства из [DKL+18,KLS18] для задачи MLWR и для другого выбора модуля (в схеме Dilithium используется $q \equiv 1 \bmod n$, в то время как здесь q – степень двойки). Наша подпись построена на парадигме Фиата-Шамира[2] (точнее на парадигме Фиата-Шамира с прерываниями [Lyu09]). Доказательство безопасности в модели QROM (Quantum Random Oracle Model, или квантовая модель случайного оракула)[3] состоит из двух этапов.

На первом этапе мы показываем, что существует алгоритм (Алгоритм 4.1), симулирующий генерацию подписи Sign() так, что результат работы симулятора статистически неотличим от выхода реальной процедуры Sign() Алгоритма 3.2. На языке парадигмы Фиата-Шамира это означает, что в основе подписи лежит протоколом с нулевым разглашением, а именно, Honest Verifier Zero Knowledge (нулевое разглашение с честным проверяющим, см. [KLS18]). Доказательство точь-в-точь соответствует доказательству для схемы Dilithium, для полноты мы его повторим в Лемме 1.

Алгоритм 4.1 Симулятор подписи

Вход: $pk = (\mathbf{A}, \mathbf{t}) \in R_q^{k \times \ell} \times R_p^k$
Выход: $(\mathbf{z}, \mathbf{c}) \in R_q^\ell \times \mathcal{C}$

1: С вероятностью $1 - \frac{|S^\ell_{\gamma-\beta-1}|}{|S^\ell_{\gamma-1}|}$ **restart**
2: $\mathbf{z} \leftarrow S^\ell_{\gamma-\beta-1}$
3: **if** $\left\|\mathsf{LSB}(\mathbf{Az} - 2^{\nu-\mu}\mathbf{tc}, \nu - d)\right\|_\infty > 2^{\nu-d} - c \cdot 2^{\nu-\mu+1}$ **then**
4: **restart**
5: **return** $(\mathbf{z}, \mathbf{c})$

Лемма 1 *Если $\beta \geq \max_{\mathbf{c}\in\mathcal{C},\mathbf{s}\in\mathcal{S}} \|\mathbf{c} \cdot \mathbf{s}\|_\infty$, то выходы Алгоритма 3.2 и Алгоритма 4.1 имеют одинаковое распределение.*

Доказательство. Для любого $\mathbf{z} \in S^\ell_{\gamma-\beta-1}$ и для любого $\mathbf{c} \in \mathcal{C}$ выполняется

$$\Pr_{\mathbf{y}\leftarrow S^\ell_{\gamma-1}}[\mathbf{y} + \mathbf{cs} = \mathbf{z}] = \Pr_{\mathbf{y}\leftarrow S^\ell_{\gamma-1}}[\mathbf{y} = \mathbf{z} - \mathbf{cs}].$$

Так как мы полагаем $\|\mathbf{c} \cdot \mathbf{s}\|_\infty \leq \beta$, то $\mathbf{z} - \mathbf{cs} \in S^\ell_{\gamma-1}$, следовательно, $\Pr_{\mathbf{y}\leftarrow S^\ell_{\gamma-1}}[\mathbf{y} = \mathbf{z} - \mathbf{ct}] = \frac{1}{|S^\ell_{\gamma-1}|}$, а значит, каждый элемент $\mathbf{z} \in S^\ell_{\gamma-\beta-1}$ имеет одинаковый шанс быть сгенерированным в результате работы Алгоритма 3.2, а это означает, что Шаг 2 Алгоритма 4.1 идеально симулирует вектор $\mathbf{z}$ реального алгоритма.

Кроме того, вероятность **restart** в Алгоритме 3.2 на шаге 6 вследствие выполнения $\|\mathbf{z}\|_\infty \geq \gamma - \beta$ равна $1 - \frac{|S^\ell_{\gamma-\beta-1}|}{|S^\ell_{\gamma-1}|}$. Это – в точности вероятность **restart** на первом шаге Алгоритма 4.1. Оставшиеся шаги Алгоритмов 3.2 и 4.1 идентичны. □

Далее мы покажем, что в процессе генерации подписи на вход хэш-функции $\mathcal{H}$ подается достаточное количество энтропии. Подобное доказательство в [KLS18, Appendix C] сводится к получению нижней границы вероятности того, что случайный элемент в R_q обратим. В случае простого q вероятность получения обратимого элемента из случайно выбранного, как минимум, равна $(1 - n/q)$. В

[2] Фиат-Шамир предложили универсальный метод построения схемы подписи из криптографической хэш-функции и схемы идентификации, обладающей некоторым свойствами безопасности.

[3] В QROM, в отличие привычного ROM (классической модели случайного оракула), атакующий имеет квантовый доступ к случайному оракулу, то есть может запрашивать значения оракула для сообщений в квантовой суперпозиции.

случае q – степени двойки эта вероятность достаточно мала. Поэтому для формирования открытого ключа $\mathbf{A} \in R_q^{k \times \ell}$ мы генерируем заведомо обратимый элемент с помощью Факта 1. Это эффективная процедура и количество обратимых элементов в R_q – экспоненциально от n. Далее рассуждения аналогичны доказательству в [KLS18, Appendix C], поэтому мы здесь приведём только утверждение в наших обозначениях.

Лемма 2 *Для* $\mathbf{A} \leftarrow R_q^{k \times \ell}$ *такого, что хотя бы один из* $k \cdot \ell$ *многочленов, формирующих* $\mathbf{A}$, *– обратимый, и для всех* $\mathbf{w} \in R_q^{\ell}$*:*

$$\Pr_{\mathbf{y} \in S_{\gamma-1}^{\ell}} [\mathsf{MSB}(\mathbf{A}\mathbf{y}, d) = \mathbf{w}] < \left(\frac{d+1}{2\gamma - 1} \right)^n.$$

Комбинируя результаты Лемм 1 и 2, мы можем утверждать, что для того, чтобы доказать $\mathsf{UF-CMA}$ стойкость схемы подписи, достаточно показать лишь $\mathsf{UF-NMA}$ стойкость (в $\mathsf{UF-NMA}$ атакующий не имеет доступа к подписывающему оракулу). Это классический результат универсальных редукций [KLS18]. Поэтому на втором шаге доказательства мы покажем, что, используя сложность задач MLWR и SelfTargetMSIS, наша подпись стойкая в модели $\mathsf{UF-NMA}$. Этот шаг опять же является адаптацией доказательства [KLS18, Лемма 4.10] для схемы Dilithium.

Лемма 3 *Для любого квантового алгоритма* $\mathcal{A}$*, успешно атакующего схему подписи, представленную в Алгоритмах 3.1–3.3, в модели* $\mathsf{UF-NMA}$ *с использование оракула* $\mathcal{H}$ *не более* $Q_{\mathcal{H}}$ *раз, существуют квантовые алгоритмы* $\mathcal{B}, \mathcal{C}$*, такие, что*

$$\mathsf{Adv}^{\mathsf{UF-NMA}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{MLWR}}_{PARAMS}(\mathcal{B}) + \mathsf{Adv}^{\mathrm{SelfTargetMSIS}}_{PARAMS}(\mathcal{C}),$$

с временем работы $T(\mathcal{B}) = T(\mathcal{C}) = T(\mathcal{A}) + Q_{\mathcal{H}}$.

Доказательство. Положим, алгоритму $\mathcal{C}$, атакующему SelfTargetMSIS, на вход подаётся матрица $A' = [\mathbf{A}|\mathbf{t}'|\mathbb{I}] \in R_q^{k \times (\ell+1+k)}$. Далее алгоритм $\mathcal{C}$ полагает $\mathbf{t} = \mathsf{Round}(\mathbf{t}' \frac{p}{q}) = \mathsf{MSB}(\mathbf{t}' + \mathbf{h} \bmod q, \mu)$ и использует пару $(\mathbf{A}, \mathbf{t})$ в качестве открытого ключа для алгоритма $\mathcal{A}$. Относительно предположения сложности задачи MLWR такая пара неотличима от реального отрытого ключа, возвращаемого Алгоритмом 3.1. В таком случае алгоритм $\mathcal{A}$ возвращает подпись $(\mathbf{z}, \mathbf{c})$, которая для некоторого сообщения M проходит алгоритм верификации: $\|\mathbf{z}\|_\infty < \gamma - \beta$ и

$$c = \mathcal{H}(\mathsf{MSB}(\mathbf{w}, d), M),$$

где $\mathbf{w} = \mathbf{A}\mathbf{z} - 2^{\nu-\mu}\mathbf{t} \cdot \mathbf{c}$. Так как $\mathsf{MSB}(\mathbf{w}, d) \cdot 2^{\nu-d} = \mathbf{w} - \mathsf{LSB}(\mathbf{w}, \nu - d)$ и $\mathbf{t}' = 2^{\nu-\mu}\mathbf{t} + \mathsf{LSB}(\mathbf{t}', \nu - \mu)$, уравнение верификации эквивалентно

$$c = \mathcal{H}(\mathbf{A}\mathbf{z} - 2^{\nu-\mu}\mathbf{t} \cdot \mathbf{c} - \mathsf{LSB}(\mathbf{w}, \nu - d), M) = \mathcal{H}(\mathbf{A}\mathbf{z} - \mathbf{t}'\mathbf{c} + \mathbf{e}, M),$$

где $\mathbf{e} = -\mathsf{LSB}(\mathbf{w}, \nu - d) - \mathsf{LSB}(\mathbf{t}', \nu - \mu) \cdot \mathbf{c}$. Кроме того, $\mathbf{e}$: $\|\mathbf{e}\|_\infty < 2^{\nu-d} + w \cdot 2^{\nu-\mu}$. Отсюда получаем решение задачи SelfTargetSIS

$$c = \mathcal{H}\left([\mathbf{A}|\mathbf{t}'|\mathbb{I}] \begin{bmatrix} \mathbf{z} \\ \mathbf{c} \\ \mathbf{e} \end{bmatrix}, M \right),$$

где $\|[\mathbf{z}|\mathbf{c}|\mathbf{e}]\|_\infty < \max\{\gamma - \beta, w, 2^{\nu-d} + w \cdot 2^{\nu-\mu}\}$. □

В итоге стойкость схемы базируется на трех задачах: MLWR, MSIS и SelfTargetMSIS. Интуитивно, безопасность ключа основана на задаче MLWR (т.е. задача восстановления ключа сводится к решению MLWR), получение поддельной подписи сводится к задаче SelfTargetMSIS, а MSIS необходима для достижения безопасности в модели $\mathsf{sUF-CMA}$. Объединяя все предыдущие утверждения, получаем основную теорему безопасности нашей схемы подписи:

Теорема 1 *Для любого квантового алгоритма $\mathcal{A}$, успешно атакующего схему подписи, представленную в Алгоритмах 3.1–3.3, в модели* $\mathsf{sUF-CMA}$ *с использование оракула $\mathcal{H}$ не более $Q_{\mathcal{H}}$ раз, существуют квантовые алгоритмы $\mathcal{B},\mathcal{C},\mathcal{D}$ с временем работы $T(\mathcal{D}) = T(\mathcal{B}) = T(\mathcal{C}) = T(\mathcal{A}) + Q_{\mathcal{H}}$, такие, что*

$$\mathsf{Adv}^{\mathsf{sUF-CMA}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathrm{MLWR}}_{k,\ell,p,q}(\mathcal{B}) + \mathsf{Adv}^{\mathrm{SelfTargetMSIS}}_{k,\ell,q,B}(\mathcal{C}) + \mathsf{Adv}^{\mathrm{MSIS}}_{k,\ell,q,B'}(\mathcal{D}) + \left(\frac{d+1}{2\gamma-1}\right)^n,$$

где $B = \max\{\gamma-\beta, 2^{\nu-d} + w2^{\nu-\mu}\}$, $B' = \max\{2^{\nu-d+1}, 2(\gamma-\beta)\}$.

Доказательство. Доказательство стойкости схемы в модели $\mathsf{UF-CMA}$, как следует из Лемм 1, 2 и 3, полагается на сложность задач MLWR и SelfTargetSIS. Для получения доказательства в более сильной модели, $\mathsf{sUF-CMA}$, мы используем сложность задачи MSIS. Напомним, что разница между двумя моделями состоит в том, что в $\mathsf{sUF-CMA}$ атакующий успешен, если он сгенерирует новую подпись $\sigma' = (\mathbf{z}', \mathbf{c})$ сообщения M, для которого атакующий уже мог знать другую корректную подпись $\sigma \neq \sigma'$.

Положим, $\mathcal{D}$ находит такой вектор $\mathbf{z}'$, что алгоритм верификации принимает пару $(\mathbf{z}', \mathbf{c})$ как корректную подпись для некоторого сообщения M. Тогда $\mathsf{MSB}(\mathbf{w}', d) = \mathsf{MSB}(\mathbf{w}, d)$, где $\mathbf{w}' = \mathbf{A}\mathbf{z}' - 2^{\nu-\mu}\mathbf{t}\mathbf{c}$ и $\mathbf{w} = \mathbf{A}\mathbf{z} - 2^{\nu-\mu}\mathbf{t}\mathbf{c}$. Так как $\mathbf{w}' = 2^{\nu-d}\mathsf{MSB}(\mathbf{w}', d) - \mathsf{LSB}(\mathbf{w}', \nu-d)$ и $\|\mathsf{LSB}(\mathbf{w}', \nu-d)\|_\infty \leq 2^{\nu-d} - 1$ (аналогично для $\mathbf{w}$), получаем

$$\left\|\mathbf{A}\mathbf{z} - 2^{\nu-\mu}\mathbf{t}\mathbf{c} - 2^{\nu-d}\mathsf{MSB}(\mathbf{w}, d)\right\|_\infty \leq 2^{\nu-d} - 1$$
$$\left\|\mathbf{A}\mathbf{z}' - 2^{\nu-\mu}\mathbf{t}\mathbf{c} - 2^{\nu-d}\mathsf{MSB}(\mathbf{w}', d)\right\|_\infty \leq 2^{\nu-d} - 1$$

Суммируя оба неравенства, получаем $\mathbf{A}(\mathbf{z}-\mathbf{z}') + \mathbf{e} = 0$, где $\|\mathbf{e}\|_\infty \leq 2^{\nu-d+1} - 2$ и $\|\mathbf{z}-\mathbf{z}'\|_\infty \leq 2(\gamma-\beta)$. Таким образом, мы имеем решение для задачи MSIS. □

5 Атаки и выбор параметров

Безопасность нашей схемы подписи основана на двух классических задачах на решётках - задачах MLWR и MSIS. Определять конкретные параметры схемы мы будем, основываясь на сложности атака на эти задачи.

Мы работаем с модульными решетками, определенным над кольцом целых циклотомического расширения, а именно, $R = \mathbb{Z}[x]/(x^{256}+1)$, то есть мы выбираем $n = 256$. Такое n позволяет осуществлять быструю арифметику в R. Основные параметры, определяющие сложности задач MLWR и MSIS, – это k (задает ранг решеток) и ℓ (задает размер секретного вектора $\mathbf{s}$).

Решение задачи MLWR сводится к нахождению короткого вектора в q-арной решетке ранга d

$$\Lambda_{\mathrm{MLWR}} = \{\mathbf{x} \in \mathbb{Z}^d \mid [\mathrm{rot}(\mathbf{A}) \,|\, \mathbb{I} \,|\, \mathbf{t}]\,\mathbf{x} = \mathbf{0} \bmod q\},$$

где $d \leq n(\ell+k)+1$. Мы используем знак $\leq$, так как оптимальная атака может не использовать некоторые строки матрицы $[\mathrm{rot}(\mathbf{A}) \,|\, \mathbb{I} \,|\, \mathbf{t}]$. Нужный нам вектор $\mathbf{x} \in \Lambda_{\mathrm{MLWR}}$ есть вектор $\mathbf{x}_{\mathrm{short}} = [\mathrm{rot}(\mathbf{s}) | - \mathbf{t}_{\mathrm{low}} | - 1]$, где $\mathbf{t}_{\mathrm{low}} = \mathbf{A}\mathbf{s} - \mathbf{t}$ и $\|\mathbf{t}_{\mathrm{low}}\|_\infty \leq 2^{\nu-\mu}$. Это "короткий" вектор в решётке Λ_{MLWE}, так как он значительно короче $\sqrt{d}q^{\frac{1}{nk}}$ – границы Минковского для Λ_{MLWR}.

Это классическая "примальная" атака на LWR, сложность которой зависит от времени работы алгоритма BKZ для нахождения вектора длины $\|\mathbf{x}_{\mathrm{short}}\|$. Оценить конкретное время работы BKZ – нетривиальная задача. Для получения значения 104 в Таблице 1 – консервативной оценки времени работы BKZ для решения задачи LWR – мы опирались на работу [AGVW17] и программный код [ACD+18]. Мы не приводим оценку для, так называемой, "дуальной" атаки на LWR, так как "примальный" метод для наших параметров оказался значительно эффективнее.

Рассмотрим теперь сложность задачи MSIS (так как задача SelfTargetSIS сводится к MSIS, и для наших параметров атаки именно на MSIS работают эффективнее, определяющим фактором

является сложность MSIS). Наиболее эффективная из всех известных атак на MSIS – нахождение короткого вектора в решётке

$$\Lambda_{\mathrm{MSIS}} = \{\mathbf{x} \in \mathbb{Z}^d \mid [\mathrm{rot}(\mathbf{A}) \,|\, \mathbb{I}]\,\mathbf{x} = \mathbf{0} \bmod q\}.$$

В отличие от атаки на MLWR, оптимальный алгоритм для задачи MSIS может опустить некоторые *столбцы* матрицы $[\mathrm{rot}(\mathbf{A}) \,|\, \mathbb{I}]$. Решением задачи MSIS будет считаться короткий вектор $\mathbf{x} \in \Lambda_{\mathrm{MSIS}}$ нормы $\|\mathbf{x}\|_\infty \leq \max\{2^{\nu-d+1}, 2(\gamma-\beta)\}$. Для параметров, приведённых в Таблице 1, эти два значения примерно совпадают.

Для получения конкретной сложности атаки MSIS мы пользовались стратегией, описанной в [DKL+18, Appendix C].[4].

	Параметры									Классическая сложность		Квантовая сложность	
	k	ℓ	ν	μ	w	s	d	γ	$(p_1 \cdot p_2)^{-1}$	MSIS (BKZ b)	MLWR (BKZ b)	MSIS (BKZ b)	MLWR (BKZ b)
Крыжовничек	3	2	23	19	20	4	3	1048576	2.7	84 (250)	70 (204)	77 (250)	65 (250)
Крыжовник	4	3	23	19	39	4	3	1048576	12.9	120 (370)	116 (357)	110 (370)	106 (370)
Крыжовнище	4	5	23	19	50	4	3	1048576	60.8	164 (520)	161 (500)	150 (520)	147 (510)

Таблица 1: Предлагаемые параметры цифровой подписи и их уровни безопасности. В вычислениях конкретного уровня безопасности для задач MLWR, MSIS мы полагаем (консервативно), что сложность нахождения короткого вектора в решетке размерности d равна $2^{0.292d}$ (для классической атаки), и $2^{0.265d}$ (для квантовой атаки), полагая число вызовов BKZ редукции равным $8d$. Параметр β согласно определению $\beta = \|\mathbf{cs}\|_\infty \leq w \|\mathbf{s}\|_\infty = 240$.

6 Особенности реализации

Экспериментальная реализация схемы цифровой подписи на C++ находится в открытом доступе под лицензией GNU GPL v3.[5] Процесс отладки и тестирования выполнялся на ОС Ubuntu Linux 18.04.4 LTS x64, компилятор g++ из пакета GCC 9.2.1. Реализация использует отдельные части кода (библиотеки) проекта цифровой подписи Crystals-Dilithium, находящиеся в общественном достоянии. В этом разделе мы опишем технические особенности реализации и приведем измерения затрат вычислительных ресурсов по времени.

6.1 Хранение открытого ключа

Хранение открытого ключа в явном виде требует значительного расхода памяти для представления матрицы A. Она состоит из $k \cdot l$ многочленов из R_q. Таким образом, размер занимаемой матрицей A памяти составляет $k \cdot l \cdot n \cdot \nu$ бит, в то время как генерация матрицы выполняется с помощью псевдослучайного генератора из инициализирующего вектора фиксированной длины 512 бит. Поэтому мы храним значение инициализирующего вектора и вычисляем матрицу A перед каждой генерацией или верификацией подписи. Для того, чтобы обеспечить обратимость хотя бы одного из $k \times l$ многочленов в R_q, мы фиксируем один многочлен, например, находящийся на позиции $A[0,0]$, и выполняем над ним следующие действия:

1. Все нечетные коэффициенты, кроме 0-го, уменьшаем на единицу;

[4] Скрипт, с помощью которого можно получить Таблицу 1, доступен по ссылке `https://crypto-kantiana.com/elena.kirshanova/#research`

[5] `https://github.com/ElenaKirshanova/pqc_LWR_signature`

2. Если 0-й коэффициент - четный, добавляем к нему единицу.

Таким образом, на позиции $A[0,0]$ окажется многочлен, удовлетворяющий условиям [Wat07, Theorem 11.1], т.е. обратимый в R_q, причем именно этот многочлен будет получен из вектора инициализации единственным образом.

6.2 Генерация равномерного распределения

Секретные векторы $\mathbf{s} \leftarrow S_s^\ell$ (закрытый ключ) и $\mathbf{y} \leftarrow S_{\gamma-1}^\ell$ (участвует в алгоритме подписи) должны быть сгенерированы случайным образом из равномерного распределения. Для этого мы генерируем псевдослучайную последовательность байт с помощью алгоритма SHAKE-128 (SHA3), затем выделяем из этой последовательности u-битные числа, выбрав в качестве u минимальное значение, при котором $2^u > s$ и $2^u > \gamma - 1$ для векторов $\mathbf{s}$ и $\mathbf{y}$ соответственно. Каждое число проверяется на принадлежность интервалам, из которых мы осуществляем выборку. Таким образом из равномерного распределения на отрезке $[0 \dots 2^u - 1]$ мы получаем равномерное распределение на отрезках $[-s \dots s]$ и $[-(\gamma-1) \dots (\gamma-1)]$ для элементов векторов $\mathbf{s}$ и $\mathbf{y}$ соответственно.

Последовательность байт на выходе SHAKE-128 генерируется из случайного вектора инициализации. Длина выходной последовательности выбирается таким образом, чтобы вероятность успешной выборки из нее была не менее $1 - 10^{-5}$.

6.3 Эффективность по памяти и времени

Тестирование времени генерации ключевой пары, подписи и верификации было выполнено на ЦП Intel Xeon(R) E-2146G 3.50GHz x 12, время работы указано количеством тактов в Таблице 2. Измерение среднего значения времени произведено на количестве тестов $N = 1000$, при таких же условиях протестирована для сравнения реализация подписи Dilithium [DKL+18].

	|sk|	|vk|	|sig|	KeyGen	Sign	Verify
Наша подпись	3k	2.4k	1.9k	2.08M	24.6M	2.6M
Dilithium	2.05k	1.2k	2.0k	133k	700k	150k

Таблица 2: Размеры ключей и подписи в байтах для параметров, взятых из Таблицы 1 в сравнении с одним набором параметров подписи Dilithium [DKL+18], сопоставимых по уровню безопасности. Время работы алгоритмов KeyGen, Sign, Verify описано числом тактов.

Список литературы

ABB+17. Erdem Alkim, Nina Bindel, Johannes Buchmann, Özgür Dagdelen, Edward Eaton, Gus Gutoski, Juliane Krämer, and Filip Pawlega. Revisiting tesla in the quantum random oracle model. In *Post-Quantum Cryptography*, pages 143–162, 2017.

ACD+18. Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the {lwe, ntru} schemes! In *Security and Cryptography for Networks*, pages 351–367, 2018.

ADPS16. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange: A new hope. In *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, page 327–343, 2016.

AGVW17. Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving usvp and applications to lwe. In *Advances in Cryptology – ASIACRYPT 2017*, pages 297–322, 2017.

Ajt96. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 99–108, 1996.

BG14. Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. In *Topics in Cryptology – CT-RSA 2014*, pages 28–47. Springer International Publishing, 2014.

BGM+16. Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon Rosen. On the hardness of learning with rounding over small modulus. In *Theory of Cryptography*, pages 209–224, 2016.

BHMT02. Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. volume 305, pages 53–74, 2002.

BPR12. Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 719–737. Springer, 2012.

DKL+18. Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(1):238–268, Feb. 2018.

DKSRV18. Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Saber: Module-lwr based key exchange, cpa-secure encryption and cca-secure kem. In *Progress in Cryptology – AFRICACRYPT 2018*, pages 282–305, 2018.

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology — CRYPTO' 86*, pages 186–194, 1987.

KLS18. Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 552–586. Springer International Publishing, 2018.

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptography*, 75(3):565–599, June 2015.

Lyu09. Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *Advances in Cryptology – ASIACRYPT 2009*, pages 598–616, 2009.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93, 2005.

Wat07. John J. Watkins. *Topics in commutative ring theory.* Princeton University Press, 2007.