Оглавление

1. Постановка задачи

Задачей данной курсовой работы является создание пояснительной записки с пояснением алгоритма – участника второго раунда конкурса национального института стандартов и технологий (NIST) по выбору квантово-устойчивых криптоалгоритмов для стандартизации. В данной курсовой работе будет рассматриваться алгоритм FrodoKEM.

2. Описание алгоритма

Из-за возможности мощных квантовых компьютеров взламывать ныне широко используемые системы шифрования, основанные на задачах дискретного логарифмирования и задачах факторизации целых чисел, было принято решение стандартизировать постквантовый криптографический алгоритм шифрования.

FrodoKEM разработан как консервативный, но практичный постквантовый алгоритм. Ядром FrodoKEM является алгоритм шифрования на основе открытого ключа FrodoPKE, чья безопасность проистекает из тонкой параметризации хорошо изученной проблемы обучения с ошибками (LWE), которая, в свою очередь, имеет тесные связи с криптографией на решетках – задаче оптимизации на дискретных аддитивных подгруппах, заданных на множестве $\mathbb{R}^n$.

Также существует несколько разновидностей FrodoKEM: FrodoKEM-640, FrodoKEM-976 и FrodoKEM-1344. Различаются они лишь уровнем обеспечения безопасности: соответствующий или превышающий безопасность от брутфорса AES-128, AES-192 и AES-256 соответственно.

3. Основные параметры

Основные параметры FrodoPKE и FrodoKEM:

- χ, равномерное распределение вероятностей на $\mathbb{Z}$;
- $q = 2^D$, целочисленная степень двойки, $D \le 16$;
- $n, \bar{m}, \bar{n}$, целочисленные размерности матрицы, $n \equiv 0 \bmod(8)$;
- $B \le D$, количество бит, закодированных в каждой записи матрицы;
- $\ell = B \cdot \bar{m} \cdot \bar{n}$, длина битовых строк, закодированных как матрицы размера $\bar{m}$ на $\bar{n}$;
- $\text{len}_\mu = \ell$, длина сообщения в битах;
- $M = \{0,1\}^{\text{len}_\mu}$, пространство сообщений;
- $\text{len}_{\text{seed}_A}$, битовая длина начальных значений, используемых для генерации псевдослучайных матриц;

- $\text{len}_{\text{seed}_{\text{SE}}}$, битовая длина начальных значений, используемых для генерации псевдослучайных битов для выборки ошибок;
- Gen, алгоритм генерации матрицы (AES-128 или SHAKE-128);
- T_χ, таблица распределения для выборки;

Также алгоритм FrodoKEM использует дополнительные 5 параметров:

- len_s, длина битового вектора **s**, используемого для генерации псевдослучайного общего секрета в случае сбоя декапсуляции в преобразовании $\text{FO}^{\not\perp}$;
- len_z, битовая длина начального числа, используемого для псевдослучайной генерации $\text{seed}_{\mathbf{A}}$;
- len_k, битовая длина промежуточного общего секрета **k** в преобразовании $\text{FO}^{\not\perp}$;
- len_{pkh}, разрядность хеша $G_1(pk)$ открытого ключа в преобразовании $\text{FO}^{\not\perp}$;
- len_{ss}, битовая длина ключа шифрования **ss** в преобразовании $\text{FO}^{\not\perp}$;

4. Алгоритм FrodoPKE

Генерация ключей

1. Выбор случайного начального числа $\text{seed}_{\mathbf{A}}$, где последовательность бит реализована по закону равномерного распределения $\text{seed}_{\mathbf{A}} \leftarrow \$U(\{0,1\}^{\text{len}_{\text{seed}_{\mathbf{A}}}})$
2. Генерация матрицы $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{n}}$ на основе генератора псевдослучайных чисел со стартовым значением $\text{seed}_{\mathbf{A}}$, $\mathbf{A} \leftarrow \text{Frodo.Gen}(\text{seed}_{\mathbf{A}})$
3. Выбор случайного начального числа seed_{SE}, где последовательность бит реализована по закону равномерного распределения $\text{seed}_{\mathbf{SE}} \leftarrow \$U(\{0,1\}^{\text{len}_{\text{seed}_{\mathbf{SE}}}})$
4. Генерация псевдослучайной битовой строки $(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, ..., \mathbf{r}^{(2n\bar{n}-1)}) \leftarrow \text{SHAKE}(0\text{x}5\text{F} \| \text{seed}_{\mathbf{SE}}, 2n\bar{n} \cdot \text{len}_\chi)$
5. Матрица ошибок $\mathbf{S} \leftarrow \text{Frodo.SampleMatrix}((\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, ..., \mathbf{r}^{(n\bar{n}-1)})), n, \bar{n}, T_\chi)$
6. Матрица ошибок $\mathbf{E} \leftarrow \text{Frodo.SampleMatrix}((\mathbf{r}^{(n\bar{n})}, \mathbf{r}^{(n\bar{n}+1)}, ..., \mathbf{r}^{(2n\bar{n}-1)})), n, \bar{n}, T_\chi)$
7. Вычислить **B = AS + E**
8. Выходные данные: открытый ключ $pk \leftarrow (\text{seed}_{\mathbf{A}}, \mathbf{B})$ и секретный ключ $sk \leftarrow \mathbf{S}$

Шифрование

Входные данные: сообщение $\mu \in M$ и открытый ключ $pk = (\text{seed}_{\mathbf{A}}, \mathbf{B})$

1. Генерация $\mathbf{A} \leftarrow \text{Frodo.Gen}(\text{seed}_{\mathbf{A}})$
2. Выбор случайного начального числа seed_{SE}, где последовательность бит реализована по закону равномерного распределения $\text{seed}_{\mathbf{SE}} \leftarrow {}^{\$}U(\{0,1\}^{\text{len}_{\text{seed}_{\mathbf{SE}}}})$
3. Генерация псевдослучайной битовой строки $(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, ..., \mathbf{r}^{(2\bar{m}n+\bar{m}\bar{n}-1)}) \leftarrow \text{SHAKE}(0\text{x}96 || \text{seed}_{\mathbf{SE}}, 2\bar{m}n+\bar{m}\bar{n} \cdot \text{len}_{\chi})$
4. Матрица ошибок $\mathbf{S}' \leftarrow \text{Frodo.SampleMatrix}((\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, ..., \mathbf{r}^{(\bar{m}n-1)})), \bar{m}, n, T_{\chi})$
5. Матрица ошибок $\mathbf{E}' \leftarrow \text{Frodo.SampleMatrix}((\mathbf{r}^{(\bar{m}n)}, \mathbf{r}^{(\bar{m}n+1)}, ..., \mathbf{r}^{(2\bar{m}n-1)})), \bar{m}, n, T_{\chi})$
6. Матрица ошибок $\mathbf{E}'' \leftarrow \text{Frodo.SampleMatrix}((\mathbf{r}^{(2\bar{m}n)}, \mathbf{r}^{(2\bar{m}n+1)}, ..., \mathbf{r}^{(2\bar{m}n+\bar{m}\bar{n}-1)})), \bar{m}, \bar{n}, T_{\chi})$
7. Вычислить $\mathbf{B}' = \mathbf{S}'\mathbf{A} + \mathbf{E}'$ и $\mathbf{V} = \mathbf{S}'\mathbf{B} + \mathbf{E}''$
8. Выходные данные: шифртекст $c \leftarrow (\mathbf{C}_1, \mathbf{C}_2) = (\mathbf{B}', \mathbf{V} + \text{Frodo.Encode}(\mu))$

Дешифрование

Входные данные: шифртекст $c = (\mathbf{C}_1, \mathbf{C}_2)$ и секретный ключ $sk = \mathbf{S}$

1. Вычислить $\mathbf{M} = \mathbf{C}_2 - \mathbf{C}_1\mathbf{S}$
2. Выходные данные: сообщение $\mu' \leftarrow \text{Frodo.Decode}(\mathbf{M})$

5. Фудзисаки-Окамото преобразование из PKE в KEM

Пусть PKE=(KeyGen,Enc,Dec) - алгоритм ассиметричного шифрования с пространством сообщений *M* и пространством шифртекстов *C*, где *R* – это вероятностное пространство Enc. Пусть len_{s}, len_{k}, len_{pkh}, len_{ss} - это параметры, а $G_1 : \{0,1\}^* \to \{0,1\}^{\text{len}_{\text{pkh}}}$, $G_2 : \{0,1\}^* \to R \times \{0,1\}^{\text{len}_{\text{k}}}$ и $F : \{0,1\}^* \to \{0,1\}^{\text{len}_{\text{ss}}}$ - хеш-функции. Обозначим $\text{KEM}^{\not\perp}=\text{FO}^{\not\perp}[\text{PKE}, G_1, G_2, F]$ как алгоритм инкапсуляции ключа.

$\mathsf{KEM}^{\not\perp\prime}$.KeyGen():

1: $(pk, sk) \leftarrow_{\$} \mathsf{PKE.KeyGen}()$
2: $\mathbf{s} \leftarrow_{\$} \{0,1\}^{\mathsf{len_s}}$
3: $\mathbf{pkh} \leftarrow G_1(pk)$
4: $sk' \leftarrow (sk, \mathbf{s}, pk, \mathbf{pkh})$
5: **return** (pk, sk')

$\mathsf{KEM}^{\not\perp\prime}$.Encaps($pk$):

1: $\mu \leftarrow_{\$} \mathcal{M}$
2: $(\mathbf{r}, \mathbf{k}) \leftarrow G_2(G_1(pk)\|\mu)$
3: $c \leftarrow \mathsf{PKE.Enc}(\mu, pk; \mathbf{r})$
4: $\mathbf{ss} \leftarrow F(c\|\mathbf{k})$
5: **return** $(c, \mathbf{ss})$

$\mathsf{KEM}^{\not\perp\prime}$.Decaps($c, (sk, \mathbf{s}, pk, \mathbf{pkh})$):

1: $\mu' \leftarrow \mathsf{PKE.Dec}(c, sk)$
2: $(\mathbf{r}', \mathbf{k}') \leftarrow G_2(\mathbf{pkh}\|\mu')$
3: **if** $c = \mathsf{PKE.Enc}(\mu', pk; \mathbf{r}')$ **then**
4: **return** $\mathbf{ss}' \leftarrow F(c\|\mathbf{k}')$
5: **else**
6: **return** $\mathbf{ss}' \leftarrow F(c\|\mathbf{s})$

Рисунок 1. Конструкция алгоритма инкапсуляции KEM из PKE и хеш-функций G_1, G_2, F

6. Алгоритм FrodoKEM

Генерация ключей

1. Выбор случайных начальных чисел $\mathbf{s}, \text{seed}_{\mathbf{SE}}$ и $\mathbf{z}$ где последовательность бит реализована по закону равномерного распределения $\mathbf{s}\|\text{seed}_{\mathbf{SE}} \| \mathbf{z} \leftarrow_{\$} U(\{0,1\}^{\text{len}_\mathbf{s} + \text{len}_{\text{seed}_\mathbf{SE}} + \text{len}_\mathbf{z}})$
2. Генерация псевдослучайного начального числа $\text{seed}_\mathbf{A} \leftarrow \text{SHAKE}(\mathbf{z}, \text{len}_{\text{seed}_\mathbf{A}})$
3. Генерация матрицы $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{n}}$ на основе генератора псевдослучайных чисел со стартовым значением $\text{seed}_\mathbf{A}$, $\mathbf{A} \leftarrow \text{Frodo.Gen}(\text{seed}_\mathbf{A})$
4. Генерация псевдослучайной битовой строки $(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, ..., \mathbf{r}^{(2n\bar{n}-1)}) \leftarrow \text{SHAKE}(0\text{x}5\text{F}\|\text{seed}_{\mathbf{SE}}, 2n\bar{n} \cdot \text{len}_\chi)$
5. Матрица ошибок $\mathbf{S} \leftarrow \text{Frodo.SampleMatrix}((\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, ..., \mathbf{r}^{(n\bar{n}-1)})), n, \bar{n}, T_\chi)$
6. Матрица ошибок $\mathbf{E} \leftarrow \text{Frodo.SampleMatrix}((\mathbf{r}^{(n\bar{n})}, \mathbf{r}^{(n\bar{n}+1)}, ..., \mathbf{r}^{(2n\bar{n}-1)})), n, \bar{n}, T_\chi)$
7. Вычислить $\mathbf{B} \leftarrow \mathbf{AS} + \mathbf{E}$
8. Вычислить $\mathbf{b} \leftarrow \text{Frodo.Pack}(\mathbf{B})$
9. Вычислить $\mathbf{pkh} \leftarrow \text{SHAKE}(\text{seed}_\mathbf{A}\|\mathbf{b}, \text{len}_\mathbf{pkh})$
10. Выходные данные: открытый ключ $pk \leftarrow \text{seed}_\mathbf{A} \| \mathbf{b}$ и секретный ключ $sk' \leftarrow (\mathbf{s}\|\text{seed}_\mathbf{A}\|\mathbf{b}, \mathbf{S}, \mathbf{pkh})$

Инкапсуляция ключей

Входные данные: открытый ключ $pk = \text{seed}_\mathbf{A} \| \mathbf{b}$

1. Выбор случайного ключа μ где последовательность бит реализована по закону равномерного распределения $\mu \leftarrow_{\$} U(\{0,1\}^{\text{len}_\mu})$
2. Вычислить $\mathbf{pkh} \leftarrow \text{SHAKE}(pk, \text{len}_\mathbf{pkh})$
3. Генерация псевдослучайных $\text{seed}_\mathbf{SE} \| \mathbf{k} \leftarrow \text{SHAKE}(\mathbf{pkh}\|\mu, \text{len}_{\text{seed}_\mathbf{SE}} + \text{len}_\mathbf{k})$

4. Генерация псевдослучайной битовой строки $(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, ..., \mathbf{r}^{(2\bar{m}n+\bar{m}\bar{n}-1)}) \leftarrow \text{SHAKE}(0\text{x}96 || \text{seed}_{\mathbf{SE}}, 2\bar{m}n+\bar{m}\bar{n} \cdot \text{len}_{\chi})$
5. Матрица ошибок $\mathbf{S}' \leftarrow \text{Frodo.SampleMatrix}((\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, ..., \mathbf{r}^{(\bar{m}n-1)})), \bar{m}, n, T_{\chi})$
6. Матрица ошибок $\mathbf{E}' \leftarrow \text{Frodo.SampleMatrix}((\mathbf{r}^{(\bar{m}n)}, \mathbf{r}^{(\bar{m}n+1)}, ..., \mathbf{r}^{(2\bar{m}n-1)})), \bar{m}, n, T_{\chi})$
7. Генерация $\mathbf{A} \leftarrow \text{Frodo.Gen}(\text{seed}_{\mathbf{A}})$
8. Вычислить $\mathbf{B}' = \mathbf{S}'\mathbf{A} + \mathbf{E}'$
9. Вычислить $\mathbf{c}_1 \leftarrow \text{Frodo.Pack}(\mathbf{B}')$
10. Матрица ошибок $\mathbf{E}'' \leftarrow \text{Frodo.SampleMatrix}((\mathbf{r}^{(2\bar{m}n)}, \mathbf{r}^{(2\bar{m}n+1)}, ..., \mathbf{r}^{(2\bar{m}n+\bar{m}\bar{n}-1)})), \bar{m}, \bar{n}, T_{\chi})$
11. Вычислить $\mathbf{B} \leftarrow \text{Frodo.Unpack}(\mathbf{b}, n, \bar{n})$
12. Вычислить $\mathbf{V} = \mathbf{S}'\mathbf{B} + \mathbf{E}''$
13. Вычислить $\mathbf{C} \leftarrow \mathbf{V} + \text{Frodo.Encode}(\mu)$
14. Вычислить $\mathbf{c}_2 \leftarrow \text{Frodo.Pack}(\mathbf{C})$
15. Вычислить $\mathbf{ss} \leftarrow \text{SHAKE}(\mathbf{c}_1 || \mathbf{c}_2 || \mathbf{k}, \text{len}_{\mathbf{ss}})$
16. Выходные данные: шифртекст $\mathbf{c}_1 || \mathbf{c}_2$ и ключ шифрования **ss**

Декапсуляция ключей

Входные данные: шифртекст $\mathbf{c}_1 || \mathbf{c}_2$ и секретный ключ $sk' \leftarrow (\mathbf{s} || \text{seed}_{\mathbf{A}} || \mathbf{b}, \mathbf{S}, \mathbf{pkh})$

1. Вычислить $\mathbf{B}' \leftarrow \text{Frodo.Unpack}(\mathbf{c}_1)$
2. Вычислить $\mathbf{C} \leftarrow \text{Frodo.Unpack}(\mathbf{c}_2)$
3. Вычислить $\mathbf{M} \leftarrow \mathbf{C} - \mathbf{B}'\mathbf{S}$
4. Вычислить $\mu' \leftarrow \text{Frodo.Decode}(\mathbf{M})$
5. Преобразовать $pk \leftarrow \text{seed}_{\mathbf{A}} || \mathbf{b}$
6. Генерация псевдослучайных $\text{seed}_{\mathbf{SE}}' || \mathbf{k}' \leftarrow \text{SHAKE}(\mathbf{pkh} || \mu', \text{len}_{\text{seed}_{\mathbf{SE}}} + \text{len}_{\mathbf{k}})$
7. Генерация псевдослучайной битовой строки $(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, ..., \mathbf{r}^{(2\bar{m}n+\bar{m}\bar{n}-1)}) \leftarrow \text{SHAKE}(0\text{x}96 || \text{seed}_{\mathbf{SE}}, 2\bar{m}n+\bar{m}\bar{n} \cdot \text{len}_{\chi})$
8. Матрица ошибок $\mathbf{S}' \leftarrow \text{Frodo.SampleMatrix}((\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, ..., \mathbf{r}^{(\bar{m}n-1)})), \bar{m}, n, T_{\chi})$
9. Матрица ошибок $\mathbf{E}' \leftarrow \text{Frodo.SampleMatrix}((\mathbf{r}^{(\bar{m}n)}, \mathbf{r}^{(\bar{m}n+1)}, ..., \mathbf{r}^{(2\bar{m}n-1)})), \bar{m}, n, T_{\chi})$
10. Генерация $\mathbf{A} \leftarrow \text{Frodo.Gen}(\text{seed}_{\mathbf{A}})$
11. Вычислить $\mathbf{B}'' \leftarrow \mathbf{S}'\mathbf{A} + \mathbf{E}'$
12. Матрица ошибок $\mathbf{E}'' \leftarrow \text{Frodo.SampleMatrix}((\mathbf{r}^{(2\bar{m}n)}, \mathbf{r}^{(2\bar{m}n+1)}, ..., \mathbf{r}^{(2\bar{m}n+\bar{m}\bar{n}-1)})), \bar{m}, \bar{n}, T_{\chi})$
13. Вычислить $\mathbf{B} \leftarrow \text{Frodo.Unpack}(\mathbf{b}, n, \bar{n})$
14. Вычислить $\mathbf{V} \leftarrow \mathbf{S}'\mathbf{B} + \mathbf{E}''$

15. Вычислить $\mathbf{C}' \leftarrow \mathbf{V} + \text{Frodo.Encode}(\mu')$
16. Если $\mathbf{B}' \,||\, \mathbf{C} = \mathbf{B}'' \,||\, \mathbf{C}'$, то
17. Выходные данные: ключ шифрования $\mathbf{ss} \leftarrow \text{SHAKE}(\mathbf{c_1}||\mathbf{c_2}||\mathbf{k}', \text{len}_{\mathbf{ss}})$
18. Иначе
19. Выходные данные: ключ шифрования $\mathbf{ss} \leftarrow \text{SHAKE}(\mathbf{c_1}||\mathbf{c_2}||\mathbf{s}, \text{len}_{\mathbf{ss}})$

Параметры для KEM

Таблица 1. Параметры для KEM и защита и точность для них

Вероятность ошибки	Тип атаки	Обычный компьютер	Квантовый компьютер	Размер pk (в байтах)	Размер sk (в байтах)	Размер шифртекста (в байтах)
FrodoKEM-640: $n = 640$, $q = 2^{15}$, χ =[-12…12], B =2, $\bar{m} \times \bar{n} = 8 \times 8$						
$2^{-138.7}$	первичная двойственная	149 148	108 108	9616	19888	9736
FrodoKEM-976: $n = 976$, $q = 2^{16}$, χ =[-10…10], B =3, $\bar{m} \times \bar{n} = 8 \times 8$						
$2^{-199.6}$	первичная двойственная	214 214	155 154	15632	31296	15768
FrodoKEM-1344: $n = 1344$, $q = 2^{16}$, χ =[-6…6], B =4, $\bar{m} \times \bar{n} = 8 \times 8$						
$2^{-252.5}$	первичная двойственная	281 279	202 201	21632	43088	21664

7. Преимущества и недостатки

Преимущества

- Простота реализации. Реализация x64: 256 строк простого C кода (+ ранее существовавшие примитивы).
- Константное время выполнения. Благодаря отсутствию модульных сокращения, поскольку все числа являются степени двойки, нет явных модульных преобразований, поэтому легко добиться константного времени работы.
- Совместимость с существующими системами и алгоритмами. У FrodoKEM открытый ключ и инкапсуляция больших размеров, чем у привычного RSA или других постквантовых алгоритмов, например, основанных на RLWE. Тем не менее, их коммуникационные размеры достаточно малы, чтобы они все еще были совместимы со многими существующими развертываниями.
- Защита от атак по сторонним каналам. В настоящее время неизвестны ни атаки по сторонним каналам, ни контрмеры для механизмов инкапсуляции ключей LWE. Но общие методы атак, а также контрмеры, которые применяются к RLWE, также применимы к LWE. Однако, поскольку FrodoKEM не использует методы умножения на основе БПФ, то возможности для атаки значительно уменьшены.

Недостатки

- Относительно медленная работа шифрования и дешифрования.
- Функциональность алгоритма ограничена схемой шифрования и KEM. Схема подписи не предусмотрена.

8. Вывод

Результатом данной курсовой работы является описание постквантового криптографического алгоритма FrodoKEM – участника конкурса NIST. В ходе проведения работы были описаны все ключевые моменты алгоритма, а также выявлены преимущества и недостатки.

Список литературы

1. D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thom´e, L. Valenta, B. VanderSloot, E. Wustrow, S. Zanella-B´eguelin, and P. Zimmermann. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In I. Ray, N. Li, and C. Kruegel, editors, ACM CCS 2015: 22nd Conference on Computer and Communications Security, pages 5–17. ACM Press, Oct. 2015.
2. D. Aharonov and O. Regev. Lattice problems in NP ∩ coNP. Journal of the ACM, 52(5):749–765, 2005. Preliminary version in FOCS 2004.
3. M. R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In J. Coron and J. B. Nielsen, editors, Advances in Cryptology – EUROCRYPT 2017, Part II, volume 10211 of Lecture Notes in Computer Science, pages 103–129. Springer, Heidelberg, Apr. / May 2017.
4. M. R. Albrecht, C. Cid, J.-C. Faug`ere, and L. Perret. Algebraic algorithms for LWE. Cryptology ePrint Archive, Report 2014/1018, 2014. http://eprint.iacr.org/2014/1018.
5. M. R. Albrecht, J.-C. Faug`ere, R. Fitzpatrick, and L. Perret. Lazy modulus switching for the BKW algorithm on LWE. In H. Krawczyk, editor, PKC 2014: 17th International Conference on Theory and Practice of Public Key Cryptography, volume 8383 of Lecture Notes in Computer Science, pages 429–445. Springer, Heidelberg, Mar. 2014.
6. M. R. Albrecht, R. Fitzpatrick, and F. G¨opfert. On the efficacy of solving LWE by reduction to uniqueSVP. In H.-S. Lee and D.-G. Han, editors, ICISC 13: 16th International Conference on Information Security and Cryptology, volume 8565 of Lecture Notes in Computer Science, pages 293–310. Springer, Heidelberg, Nov. 2014.
7. M. R. Albrecht, F. G¨opfert, F. Virdia, and T. Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. In T. Takagi and T. Peyrin, editors, Advances in Cryptology – ASIACRYPT 2017, Part I,

volume 10624 of Lecture Notes in Computer Science, pages 297–322. Springer, Heidelberg, Dec. 2017.
8. M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of Learning with Errors. Journal of Mathematical Cryptology, 9(3):169–203, Nov 2015.
9. E. Alkim, L. Ducas, T. P¨oppelmann, and P. Schwabe. Post-quantum key exchange - A new hope. In T. Holz and S. Savage, editors, USENIX Security 2016: 25th USENIX Security Symposium, pages 327–343. USENIX Association, Aug. 2016.
10. A. Andoni, T. Laarhoven, I. P. Razenshteyn, and E. Waingarten. Optimal hashing-based time-space trade-offs for approximate near neighbors. In P. N. Klein, editor, 28th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 47–66. ACM-SIAM, Jan. 2017.
11. B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circularsecure encryption based on hard learning problems. In S. Halevi, editor, Advances in Cryptology – CRYPTO 2009, volume 5677 of Lecture Notes in Computer Science, pages 595–618. Springer, Heidelberg, Aug. 2009.
12. S. Arora and R. Ge. New algorithms for learning in presence of errors. In L. Aceto, M. Henzinger, and J. Sgall, editors, ICALP 2011: 38th International Colloquium on Automata, Languages and Programming, Part I, volume 6755 of Lecture Notes in Computer Science, pages 403–415. Springer, Heidelberg, July 2011. 4110
13. A. Becker, L. Ducas, N. Gama, and T. Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In R. Krauthgamer, editor, 27th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 10–24. ACM-SIAM, Jan. 2016.
14. A. Becker, N. Gama, and A. Joux. Speeding-up lattice sieving without increasing the memory, using sub-quadratic nearest neighbor search. Cryptology ePrint Archive, Report 2015/522, 2015. http://eprint.iacr.org/2015/522.
15. D. J. Bernstein, T. Chou, C. Chuengsatiansup, A. H¨ulsing, E. Lambooij, T. Lange, R. Niederhagen, and C. van Vredendaal. How to manipulate curve standards: A white paper for the black hat. In L. Chen and S. Matsuo, editors, Security Standardisation Research (SSR) 2015, volume 9497 of Lecture Notes in Computer Science, pages 109–139. Springer, 2015.
16. E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In B. S. Kaliski Jr., editor, Advances in Cryptology – CRYPTO'97, volume 1294 of Lecture Notes in Computer Science, pages 513–525. Springer, Heidelberg, Aug. 1997.
17. D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In W. Fumy, editor, Advances in Cryptology – EUROCRYPT'97, volume 1233 of Lecture Notes in Computer Science, pages 37–51. Springer, Heidelberg, May 1997.

18. D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In P. Q. Nguyen and E. Oswald, editors, Advances in Cryptology – EUROCRYPT 2014, volume 8441 of Lecture Notes in Computer Science, pages 533–556. Springer, Heidelberg, May 2014.
19. J. W. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila. Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE. In E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, editors, ACM CCS 2016: 23rd Conference on Computer and Communications Security, pages 1006–1018. ACM Press, Oct. 2016.
20. 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, pages 553–570. IEEE Computer Society Press, May 2015.