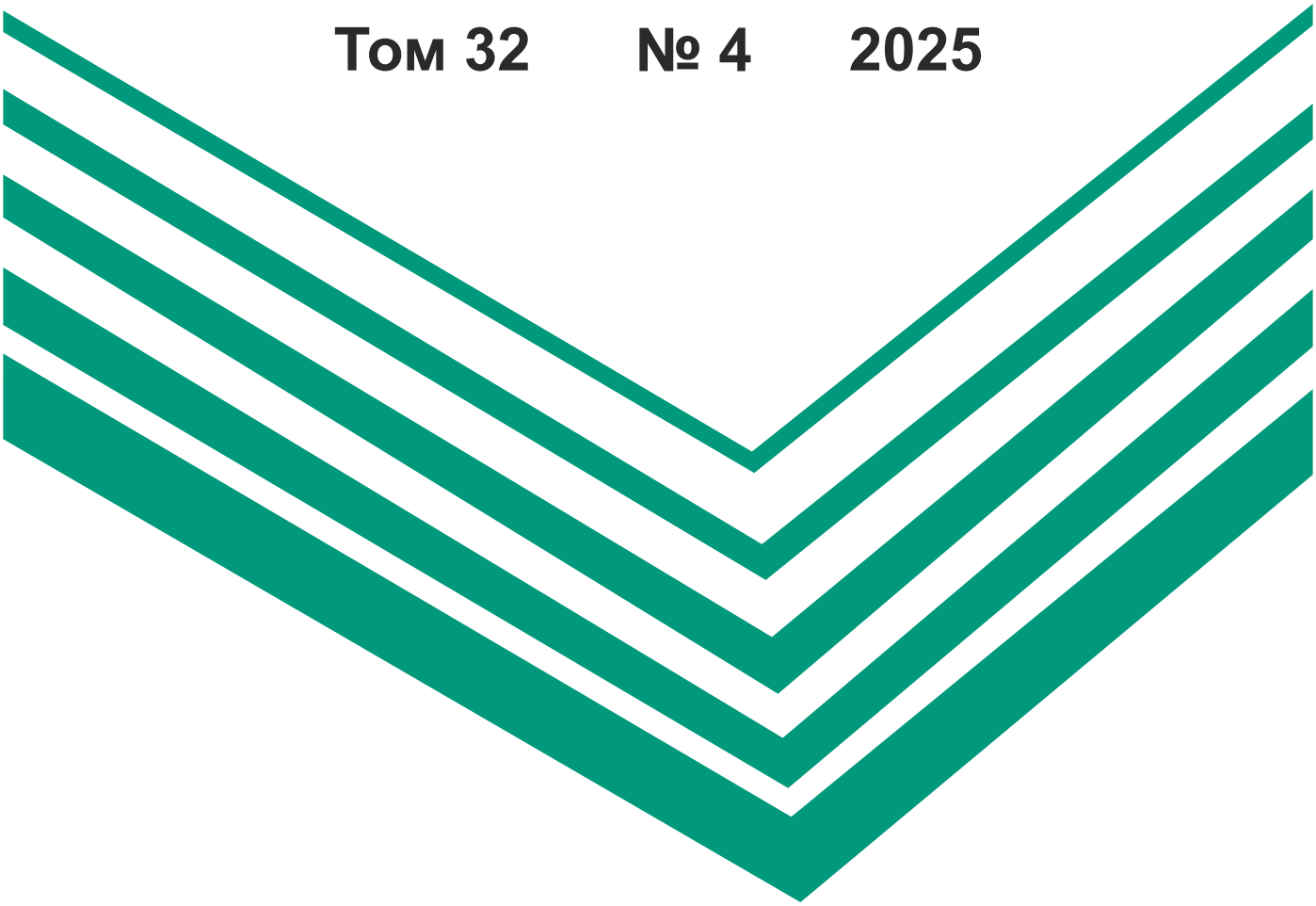
ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 32 № 4 2025



Новосибирск
Издательство Института математики

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь–декабрь 2025. Т. 32, № 4. С. 5–68

УДК 519.7 DOI: 10.33048/daio.2025.32.837

АТАКИ ПО ПОБОЧНЫМ КАНАЛАМ НА ТЕОРЕТИКО-КОДОВЫЕ ПОСТКВАНТОВЫЕ КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ: ОБЗОР. ЧАСТЬ 1

А. О. Бахарев[1,2,a], *Д. М. Воронов*[1,2,b], *Н. А. Коломеец*[1,2,c], *Н. Н. Токарева*[1,2,3,d], *И. С. Хильчук*[1,2,e], *А. С. Шапоренко*[1,2,f]

[1] Национальный технологический центр цифровой криптографии,
Раменский б-р, 1, 119607 Москва, Россия
[2] Новосибирский гос. университет,
ул. Пирогова, 2, 630090 Новосибирск, Россия
[3] Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия
E-mail: [a] a.bakharev@g.nsu.ru, [b] d.voronov2@g.nsu.ru,
[c] n.kolomeets@g.nsu.ru, [d] crypto1127@mail.ru,
[e] i.khilchuk@g.nsu.ru, [f] a.shaporenko@g.nsu.ru

Аннотация. В работе, состоящей из двух частей, приводится структурированный аналитический обзор, посвящённый атакам, использующим информацию, полученную по побочным каналам, на постквантовые криптосистемы, основанные на методах и конструкциях теории помехоустойчивого кодирования. В первой части обзора представлено описание основных криптографических примитивов и алгоритмов, применяемых в теоретико-кодовых криптосистемах, а также приведены описания наиболее значимых современных теоретико-кодовых схем: Classic McEliece, «Кодиеум», «Шиповник», BIKE и HQC. Представленное исследование выполнено в рамках НИР «Кульминация», проведённой в АНО «Национальный технологический центр цифровой криптографии». Табл. 5, ил. 14, библиогр. 111.

Ключевые слова: постквантовая криптография, атака по побочным каналам, теоретико-кодовая криптографическая система.

Введение

Криптография с открытым ключом является важнейшей компонентой современной цифровой связи. Протоколы с открытым ключом применяются для выполнения трёх основных криптографических функций:

© А. О. Бахарев, Д. М. Воронов, Н. А. Коломеец, Н. Н. Токарева, И. С. Хильчук, А. С. Шапоренко, 2025

шифрование с открытым ключом, цифровые подписи и выработка общего секретного ключа. Большинство современных протоколов с открытым ключом основаны на сложности задач факторизации и дискретного логарифмирования. В работе [1] 1997 г. Шор показал, что квантовые компьютеры могут эффективно решать каждую из них, потенциально делая все криптосистемы с открытым ключом, основанные на сложности этих двух задач, небезопасными.

Постквантовая криптография основана на других сложных задачах, для которых неизвестны эффективные алгоритмы решения с помощью квантового компьютера. Одним из основных типов таких задач являются задачи, относящиеся к области помехоустойчивого кодирования. Криптосистемы, основанные на задачах такого типа, называют *теоретико-кодовыми*.

При этом важную роль для оценки защищённости криптографических систем на практике играет учёт методов криптоанализа, которые основаны на особенностях реализаций теоретико-кодовых систем и используют информацию, получаемую злоумышленником по побочным каналам, например время выполнения криптосистемой тех или иных операций, электромагнитное излучение или энергопотребление. Таким образом, оценка устойчивости потенциальных реализаций теоретико-кодовых криптосистем относительно методов этого типа криптоанализа является актуальной задачей.

В работе, состоящей из двух частей, приводится структурированный аналитический обзор, посвящённый атакам по побочным каналам, опубликованным в открытых источниках, на современные постквантовые теоретико-кодовые криптосистемы. В разд. 1 ч. 1 приводится терминология и основные определения, которые используются в данном обзоре. Разд. 2 посвящён описанию криптографических примитивов и алгоритмов, применяемых в теоретико-кодовых криптосистемах. Введению в теоретико-кодовые криптосистемы посвящён разд. 3. В разд. 4–7 приводится описание наиболее значимых современных теоретико-кодовых схем: Classic McEliece, «Кодиеум», «Шиповник», BIKE и HQC. Разд. 8 посвящён основам атак по побочным каналам. В разд. 9 приводится обзор общей представленности работ по рассматриваемой тематике в трудах ведущих конференций.

В ч. 2 обзора проводится детальный анализ наиболее значимых работ последних лет, посвящённых атакам по побочным каналам на современные теоретико-кодовые криптосистемы, а также применимости рассмотренных атак к криптосистемам Classic McEliece, «Кодиеум» и «Шиповник».

1. Терминология и основные определения

1.1. Обозначения. Будем использовать следующие обозначения:

- $\mathbb{F}_q$ — конечное поле из $q = p^n$ элементов для простого p и натурального n; обычно $p = 2$, т. е. рассматривается $\mathbb{F}_{2^n}$;
- $\mathbb{F}_q^n$ — векторное пространство размерности n над $\mathbb{F}_q$, наиболее часто используется $\mathbb{F}_2^n$;
- координаты q-ичного вектора $x \in \mathbb{F}_q^n$ могут нумероваться как с нуля, так и с единицы в зависимости от контекста;
- $\langle A \rangle_{\mathbb{F}_q} = \langle \alpha_1, \ldots, \alpha_m \rangle_{\mathbb{F}_q}$ — линейная оболочка над полем $\mathbb{F}_q$ множества векторов $A = \{\alpha_1, \ldots, \alpha_m\} \subseteq \mathbb{F}_q^n$;
- $\mathbb{F}_q[x]$ — кольцо многочленов переменной x над полем $\mathbb{F}_q$;
- $\operatorname{rk} H$ — ранг матрицы H;
- $\operatorname{wt}(x)$ — вес Хэмминга вектора $x \in \mathbb{F}_q^n$, равный числу его ненулевых координат;
- $d(x, y)$ — расстояние Хэмминга между векторами $x, y \in \mathbb{F}_q^n$, равное числу координат, в которых x и y различаются;
- $[n, k, d]_q$-код $\mathcal{C}$ — линейное подпространство $\mathcal{C} \leqslant \mathbb{F}_q^n$ размерности k такое, что $\min\limits_{x \in \mathcal{C} \setminus \{0\}} \operatorname{wt}(x) = d$.

Обозначения в алгоритмах и протоколах:

- $\perp$ — символ ошибки;
- $[\,]$ — пустой массив;
- $x \leftarrow y$ — операция присвоения переменной x значения y;
- $a \xleftarrow{\$} \mathrm{Alg}$ — операция запуска вероятностного алгоритма Alg с последующим присвоением результата работы Alg переменной a;
- $a \xleftarrow{\mathcal{U}} A$ — операция присвоения переменной a случайного значения в соответствии с равномерным распределением на множестве A;
- $a \parallel b$ — результат конкатенации строк a и b.

Основное внимание будет обращено на следующие криптосистемы:

- NIED — криптосистема Нидеррайтера [2];
- ME — оригинальная криптосистема Мак-Элиса [3];
- CME — криптосистема Мак-Элиса, поданная на конкурс NIST под названием Classic McEliece [4];
- CODI — криптосистема «Кодиеум» [5];
- SHIP — криптосистема «Шиповник» [6].

1.2. Базовые определения. Напомним, что любую матрицу H размера $m \times n$, $m \leqslant n$, над полем $\mathbb{F}_2$ можно привести к приведённому ступенчатому виду (reduced row-echelon form) элементарными преобразованиями строк. Обозначим такую матрицу через H', будем также считать, что $\operatorname{rk} H = m$. Привести матрицу H к указанной форме можно с помощью метода Гаусса, при этом H' определяется однозначно.

Далее будем нумеровать строки, столбцы и координаты с нуля. Обозначим через c_i номер столбца, в котором стоит *ведущий элемент* i-й строки матрицы H' — первая единица слева в i-й строке и единственная в столбце c_i в силу вида H'. Ясно, что

$$0 \leqslant c_0 < c_1 < \cdots < c_{m-1} < n.$$

Напомним, что матрица H' называется *систематической*, если $c_i = i$ для всех $i = 0, \ldots, m-1$. Будем называть H' (μ, ν)-*полусистематической*, $\mu \leqslant \nu$, если

- $c_i = i$ для всех $0 \leqslant i < m - \mu$,
- $c_i \leqslant i - \mu + \nu$ для всех $0 \leqslant i < m$.

При $\mu = \nu$ любая (μ, ν)-полусистематическая матрица H' является систематической. Если $\nu > \mu$, то (μ, ν)-полусистематическая форма допускает большую свободу. В спецификации криптосистемы CME используется как $(\mu, \nu) = (0, 0)$, так и $(\mu, \nu) = (32, 64)$ (см. табл. 1: параметры с обозначением f отвечают использованию полусистематической формы).

Теория кодирования. Опишем основные определения теории кодирования. *Весом Хэмминга* $\mathrm{wt}(x)$ вектора $x \in \mathbb{F}_q^n$ называется число ненулевых координат данного вектора. *Расстоянием Хэмминга* $d(x, y)$ для векторов $x, y \in \mathbb{F}_q^n$ будем называть число координат, в которых они отличаются. Вес Хэмминга и расстояние Хэмминга связаны отношением

$$d(x, y) = \mathrm{wt}(x - y).$$

Произвольное подмножество $\mathcal{C} \subseteq \mathbb{F}_q^n$ называется *кодом длины* n, а элементы $\mathcal{C}$ — кодовыми словами. Отметим, что мы рассматриваем элементы пространства $\mathbb{F}_q^n$ как вектор-строки. *Кодовым расстоянием* d кода $\mathcal{C}$ называется величина, равная минимальному расстоянию Хэмминга между двумя различными кодовыми словами, т. е.

$$d = \min\{d(x, y) \mid x, y \in \mathcal{C},\ x \neq y\}.$$

Код *линейный*, если он образует подпространство $\mathbb{F}_q^n$. Запись $[n, k, d]_q$-код $\mathcal{C}$ означает, что $\mathcal{C} \leqslant \mathbb{F}_q^n$ — линейный код размерности k длины n с кодовым расстоянием d. Отметим, что для линейного кода справедливо равенство

$$d = \min_{x \in \mathcal{C} \setminus \{0\}} \mathrm{wt}(x).$$

Порождающей матрицей $G \in \mathbb{F}_q^{k \times n}$ линейного кода $\mathcal{C}$ называется матрица, строки которой образуют базис $\mathcal{C}$. Матрица $H \in \mathbb{F}_q^{(n-k) \times n}$ называется *проверочной*, если $x \in \mathcal{C}$ в том и только в том случае, когда

$$Hx^\top = 0.$$

Синдромом вектора $x \in \mathbb{F}_q^n$ относительно линейного кода $\mathcal{C}$ называется вектор $Hx^\top$, где H — проверочная матрица $\mathcal{C}$. *Ортогональным кодом* для $[n, k, d]_q$-кода $\mathcal{C}$ называется множество $\mathcal{C}^\perp \subseteq \mathbb{F}_q^n$, для элементов $x \in \mathcal{C}^\perp$ которого выполняется равенство

$$Gx^\top = 0.$$

Если $t \in \mathbb{N}$ — максимальное такое число, что для любых $e \in \mathbb{F}_q^n$, $\mathrm{wt}(e) \leqslant t$, и кодовых слов $x, y \in \mathcal{C}$, $x \neq y$, выполнено неравенство

$$d(x, x + e) < d(y, x + e),$$

то будем говорить, что код $\mathcal{C}$ *исправляет* t *ошибок*. Отметим, что для кода с кодовым расстоянием d имеет место равенство

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor.$$

Определим задачу синдромного декодирования, на которой основывается стойкость криптосистем, построенных на кодах, исправляющих ошибки. В общем случае эта задача NP-трудна [7].

Задача 1 (задача синдромного декодирования при ровно t ошибках). *Даны проверочная матрица $H \in \mathbb{F}_2^{(n-k)\times n}$ линейного кода, целое $t > 0$ и синдром $s = He^\top \in \mathbb{F}_2^{n-k} \setminus \{0\}$, где $e \in \mathbb{F}_2^n$ и $wt(e) = t$. Найти вектор $e' \in \mathbb{F}_2^n$ такой, что $wt(e') = t$ и $He'^\top = s$.*

Пусть $g(x)$ — многочлен степени t над $\mathbb{F}_{q^m}$ и $\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{F}_{q^m}$ — различные элементы, не являющиеся корнями g. *Кодом Гоппы* длины n над полем $\mathbb{F}_q$ называется код, предложенный в [8]:

$$\Gamma(g, \alpha_0, \ldots, \alpha_{n-1}) = \left\{ (u_0, \ldots, u_{n-1}) \in \mathbb{F}_q^n \mid \sum_{i=0}^{n-1} \frac{u_i}{x - \alpha_i} \equiv 0 \pmod{g(x)} \right\}.$$

Код Гоппы представляет собой $[n, k, d]_q$-код, для которого справедливы оценки

$$k \geqslant n - m \deg g, \quad d \geqslant \deg g + 1.$$

Такие коды можно задать также с помощью обобщённых кодов Рида — Соломона. Отметим, что на практике в криптосистемах используются коды размерности $k = n - m \deg g$, (см., например, п. 4.4 с описанием процедуры CME.Kgen(): если код не удовлетворяет нужным параметрам, ключи генерируются заново).

Если выбран неприводимый многочлен, то используют термин *неприводимый* код Гоппы. Важным частным случаем таких кодов является двоичный код Гоппы ($q = 2$). Двоичный код Гоппы называется *сепарабельным*, если выбранный многочлен Гоппы не имеет кратных корней.

Кодовое расстояние двоичного сепарабельного кода Гоппы удовлетворяет неравенству

$$d \geqslant 2 \deg g + 1,$$

т. е. указанный код исправляет не менее $\deg g$ ошибок.

Шифрование с открытым ключом и инкапсуляция ключа. Здесь рассматриваются схемы шифрования с открытым ключом, схемы инкапсуляции ключа и то, как получить одно из другого.

Определение 1. Для заданного пространства сообщений $\mathcal{M}$ *схемой шифрования с открытым ключом* будем называть тройку полиномиальных (вероятностных) алгоритмов $\mathsf{PKE} = (\mathsf{Kgen}, \mathsf{Enc}, \mathsf{Dec})$:

- $(pk, sk) \xleftarrow{\$} \mathsf{PKE.Kgen}()$ — алгоритм генерации ключевой пары, возвращает открытый ключ pk и секретный ключ sk;
- $c \xleftarrow{\$} \mathsf{PKE.Enc}(pk, m)$ — алгоритм шифрования, принимает на вход сообщение m и открытый ключ pk и возвращает шифртекст c;
- $b \leftarrow \mathsf{PKE.Dec}(sk, c)$ — алгоритм расшифрования, принимает на вход секретный ключ sk и шифртекст c и возвращает m, если c — корректный шифртекст сообщения m, и $\perp$ иначе.

Для схемы шифрования с открытым ключом PKE должно выполняться стандартное требование корректности зашифрования и расшифрования сообщения:

$$(pk, sk) \xleftarrow{\$} \mathsf{PKE.Kgen}() \Rightarrow \mathsf{PKE.Dec}(sk, \mathsf{PKE.Enc}(pk, m)) = m.$$

Здесь и далее требование корректности может не выполняться с пренебрежимо малой вероятностью вне зависимости от входных и генерируемых данных алгоритма.

Определение 2. Для заданного пространства ключей $\mathcal{K}$ *схемой инкапсуляции ключа* будем называть тройку полиномиальных (вероятностных) алгоритмов $\mathsf{KEM} = (\mathsf{Kgen}, \mathsf{Encaps}, \mathsf{Decaps})$:

- $(pk, sk) \xleftarrow{\$} \mathsf{KEM.Kgen}()$ — алгоритм генерации ключевой пары, возвращает открытый ключ pk и секретный ключ sk;
- $(K, c) \xleftarrow{\$} \mathsf{KEM.Encaps}(pk)$ — алгоритм инкапсуляции, на вход принимает открытый ключ pk и возвращает ключ K и его инкапсуляцию c;
- $K \leftarrow \mathsf{KEM.Decaps}(sk, c)$ — алгоритм декапсуляции, на вход принимает секретный ключ sk и инкапсуляцию c и возвращает K, если c — корректная инкапсуляция ключа K, и $\perp$ иначе.

Для схемы инкапсуляции ключа KEM должно выполняться стандартное требование корректности алгоритмов инкапсуляции и декапсуляции:

$$(pk, sk) \xleftarrow{\$} \mathsf{Kgen}(),\ (K, c) \leftarrow \mathsf{Encaps}(pk) \Rightarrow K = \mathsf{Decaps}(sk, c).$$

После небольшого введения в терминологию теоретической (доказуемой) стойкости определим модели угроз, относительно которых принято рассматривать стойкость представленных криптосистем.

Противник $\mathcal{A}$ — некоторый вероятностный алгоритм (вероятностная машина Тьюринга). Под вычислительными ресурсами противника будем понимать величину, ограничивающую сумму времени работы противника (например число тактов вычислений) и размера его программы (эта оговорка необходима для исключения ситуаций, в которых в код противника прописываются некоторые предвычисленные таблицы, упрощающие перебор). Запись $\mathcal{A}^{\mathcal{O}}$ означает противника (вероятностный алгоритм) $\mathcal{A}$, имеющего доступ к оракулу $\mathcal{O}$. Под записью $\mathbb{P}[\mathcal{A} \to a]$ понимаем вероятность того, что вероятностный алгоритм $\mathcal{A}$ выдаст a. Противник взаимодействует с окружением (экспериментом) посредством обращения к набору оракулов, которые формализуют возможности противника по взаимодействию с некоторой реальной системой.

Через $\mathrm{Exp}^{M}_{\mathcal{K}}$ обозначим эксперимент — вероятностный алгоритм, моделирующий для противника условия (модель) M, в рамках которых он взаимодействует с криптосистемой $\mathcal{K}$. Будем рассматривать два типа экспериментов.

- В экспериментах первого типа противнику необходимо реализовать угрозу, определив некоторую искомую величину, например найти дискретный логарифм или подделать подпись (задача поиска).
- В экспериментах второго типа перед началом взаимодействия с противником случайно равновероятно выбирается один из двух сценариев воспроизведения, а противнику необходимо определить, по какому сценарию воспроизводится эксперимент (задача различения). В этом случае для модели используются обозначения M-0 или M-1 в зависимости от номера сценария воспроизведения. Примером эксперимента второго типа может служить задача различения генерируемых криптосистемой ключей с одной стороны и случайных строк с другой, или задача различения Диффи — Хеллмана.

Для противника $\mathcal{A}$ определим его *преимущество* $\mathrm{Adv}^{M}_{\mathcal{K}}(\mathcal{A})$ в модели M для криптосистемы $\mathcal{K}$ как вероятность успешного прохождения эксперимента:

$$\mathrm{Adv}^{M}_{\mathcal{K}}(\mathcal{A}) = \mathbb{P}\big[\mathrm{Exp}^{M}_{\mathcal{K}}(\mathcal{A}) \to 1\big]$$

для экспериментов первого типа и

$$\mathrm{Adv}^{M}_{\mathcal{K}}(\mathcal{A}) = \big|\mathbb{P}\big[\mathrm{Exp}^{M\text{-}1}_{\mathcal{K}}(\mathcal{A}) \to 1\big] - \mathbb{P}\big[\mathrm{Exp}^{M\text{-}0}_{\mathcal{K}}(\mathcal{A}) \to 1\big]\big|$$

для экспериментов второго типа. В настоящей работе через $\mathrm{InSec}(\mathrm{par})$ будем обозначать максимальное преимущество среди всех противников

с ограничением par на вычислительные и иные ресурсы (количество запросов к оракулам, их максимальные и суммарные длины и т. д.), т. е.

$$\mathrm{InSec}(\mathrm{par}) = \max_{\mathcal{A} \in A(\mathrm{par})} \mathrm{Adv}^{M}_{\mathcal{K}}(\mathcal{A}),$$

где $A(\mathrm{par})$ — множество всех противников с ограничением par на вычислительные и иные ресурсы.

Определение 3. *Преимуществом* противника $\mathcal{A}$ в модели OW-CPA для схемы шифрования PKE назовём величину

$$\mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) = \mathbb{P}\big[\mathrm{Exp}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) \to 1\big],$$

где псевдокод эксперимента $\mathrm{Exp}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}$ определён на рис. 1.

$\mathrm{Exp}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A})$

1 : $(pk, sk) \xleftarrow{\$} \mathsf{Kgen}()$
2 : $m \xleftarrow{\mathcal{U}} \mathcal{M}$
3 : $c^* \leftarrow \mathsf{Enc}(pk, m)$
4 : $m' \leftarrow \mathcal{A}(pk, c^*)$
5 : **if** $m' = m$ **then**
6 : **return** 1
7 : **else**
8 : **return** 0

Рис. 1

$\mathrm{Exp}^{\mathsf{IND\text{-}CPA}\text{-}b}_{\mathsf{PKE}}(\mathcal{A})$

1 : $(pk, sk) \xleftarrow{\$} \mathsf{Kgen}()$
2 : $(m_0, m_1) \leftarrow \mathcal{A}(pk)$
3 : $c \leftarrow \mathsf{Enc}(pk, m_b)$
4 : $b' \leftarrow \mathcal{A}(pk, c)$
5 : **if** $b' = b$ **then**
6 : **return** 1
7 : **else**
8 : **return** 0

Рис. 2

Через $\mathrm{InSec}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(t)$ будем обозначать максимум среди преимуществ вида $\mathrm{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A})$, где максимум берётся по всем противникам $\mathcal{A}$ с ограничением t на вычислительные ресурсы.

Определение 4. *Преимуществом* противника $\mathcal{A}$ в модели IND-CPA для схемы шифрования PKE назовём величину

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) = \big|\mathbb{P}\big[\mathrm{Exp}^{\mathsf{IND\text{-}CPA\text{-}1}}_{\mathsf{PKE}}(\mathcal{A}) \to 1\big] - \mathbb{P}\big[\mathrm{Exp}^{\mathsf{IND\text{-}CPA\text{-}0}}_{\mathsf{PKE}}(\mathcal{A}) \to 1\big]\big|,$$

где псевдокод эксперимента $\mathrm{Exp}^{\mathsf{IND\text{-}CPA}\text{-}b}_{\mathsf{PKE}}$, $b \in \{0, 1\}$, определён на рис. 2.

Определение 5. *Преимуществом* противника в модели IND-CCA для механизма инкапсуляции ключа KEM назовём величину

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A}) = \big|\mathbb{P}\big[\mathrm{Exp}^{\mathsf{IND\text{-}CCA\text{-}1}}_{\mathsf{KEM}}(\mathcal{A}) \to 1\big] - \mathbb{P}\big[\mathrm{Exp}^{\mathsf{IND\text{-}CCA\text{-}0}}_{\mathsf{KEM}}(\mathcal{A}) \to 1\big]\big|,$$

где псевдокод эксперимента $\mathrm{Exp}^{\mathsf{IND\text{-}CCA}\text{-}b}_{\mathsf{KEM}}, b \in \{0, 1\}$, определён на рис. 3.

$\mathrm{Exp}^{\mathsf{IND\text{-}CCA}\text{-}b}_{\mathsf{KEM}}(\mathcal{A})$

1 : $(pk, sk) \xleftarrow{\$} \mathsf{Kgen}()$
2 : $(K_0^*, c^*) \xleftarrow{\$} \mathsf{Encaps}(pk)$
3 : $K_1^* \xleftarrow{\mathcal{U}} \mathcal{K}$
4 : $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{decaps}}}(c^*, K_b^*)$
5 : **return** b'

$\mathcal{O}_{\mathrm{decaps}}(c)$

1 : **if** $c = c^*$ **then**
2 : **return** $\perp$
3 : **else**
4 : $K \leftarrow \mathsf{Decaps}(sk, c)$
5 : **return** K

Рис. 3

Через $\mathrm{InSec}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(t, q_{\mathrm{decaps}})$ будем обозначать максимум среди преимуществ вида $\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathcal{A})$, где максимум берётся по всем противникам $\mathcal{A}$ с ограничением t на вычислительные ресурсы, выполняющим не более q_{decaps} запросов к оракулу $\mathcal{O}_{\mathrm{decaps}}$.

Используя преобразование Фуджисаки — Окамото [9] (FO-преобразование), из стойкой в модели OW-CPA или в модели IND-CPA схемы шифрования с открытым ключом PKE можно получить стойкую в модели IND-CCA схему инкапсуляции ключа KEM. В общем случае модель стойкости рассматриваемого шифрования с открытым ключом может быть другой, но в случае кодовых криптосистем интересны модели OW-CPA и IND-CPA. В настоящей работе будем рассматривать два вида FO-преобразования: $\mathsf{U}^{\not\perp}$ и $\mathsf{U}^{\not\perp}_m$. Пусть схема шифрования с открытым ключом

KEM.Kgen()

1 : $(pk, sk') \xleftarrow{\$} \mathsf{PKE.Kgen}()$
2 : $s \xleftarrow{\mathcal{U}} \mathcal{M}$
3 : $sk \leftarrow (sk', s)$
4 : **return** (sk, pk)

KEM.Encaps(pk)

1 : $m \xleftarrow{\mathcal{U}} \mathcal{M}$
2 : $c \xleftarrow{\$} \mathsf{PKE.Enc}(pk, m)$
3 : $K \leftarrow H(m, c)$ // $\mathsf{U}^{\not\perp}$
4 : $K \leftarrow H(m)$ // $\mathsf{U}^{\not\perp}_m$
5 : **return** (K, c)

KEM.Decaps(sk, c)

1 : $(sk', s) \leftarrow sk$
2 : $m' \leftarrow \mathsf{PKE.Dec}(sk', c)$
3 : **if** $m' = \perp$ **then**
4 : $K \leftarrow H(s, c)$
5 : **else**
6 : $K \leftarrow H(m', c)$ // $\mathsf{U}^{\not\perp}$
7 : $K \leftarrow H(m')$ // $\mathsf{U}^{\not\perp}_m$
8 : **return** K

Рис. 4. Преобразования $\mathsf{U}^{\not\perp}$ и $\mathsf{U}^{\not\perp}_m$

PKE стойкая в модели OW-CPA или IND-CPA. Тогда для некоторой хэш-функции H, действующей в пространство ключей $\mathcal{K}$, способ получения стойкой в модели IND-CCA схемы инкапсуляции ключа KEM с использованием преобразований $\mathsf{U}^{\not\perp}$ и $\mathsf{U}^{\not\perp}_m$ представлен на рис. 4.

Отметим, что в [10] показана эквивалентность преобразований $\mathsf{U}^{\not\perp}$ и $\mathsf{U}^{\not\perp}_m$ в модели QROM, представляющей собой модель безопасности над моделью M. В модели QROM в качетсве противника выступает полиномиальный квантовый алгоритм, имеющий классический доступ к оракулам модели M и квантовый доступ к случайной функции, моделирующей хэш-функцию в рассматриваемом криптомеханизме.

Схемы подписи. Здесь рассматриваются схемы подписи, их модели угроз и один из способов их построения.

Определение 6. *Схемой подписи* будем называть тройку полиномиальных (вероятностных) алгоритмов $\mathsf{SS} = (\mathsf{Kgen}, \mathsf{Sign}, \mathsf{Verify})$:

• $(sk, vk) \xleftarrow{\$} \mathsf{SS.Kgen}()$ — алгоритм генерации ключевой пары, возвращает ключ подписи sk и ключ проверки подписи vk;

• $\sigma \xleftarrow{\$} \mathsf{SS.Sign}(sk, m)$ — алгоритм формирования подписи, принимает на вход ключ подписи sk и сообщение m и возвращает подпись σ для сообщения m;

• $b \xleftarrow{\$} \mathsf{SS.Verify}(vk, m, \sigma)$ — алгоритм проверки подписи, принимает на вход ключ проверки подписи vk, сообщение m и подпись σ и возвращает 1, если подпись верна, и 0 иначе.

Для схемы подписи SS должно выполняться стандартное требование корректности формирования и проверки подписи:

$$(sk, vk) \xleftarrow{\$} \mathsf{SS.Kgen}() \Rightarrow \mathsf{SS.Verify}(vk, m, \mathsf{SS.Sign}(sk, m)) = 1.$$

Определение 7. *Преимуществом* противника $\mathcal{A}$ в модели SUF-CMA для схемы подписи SS назовём величину

$$\mathrm{Adv}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{SS}}(\mathcal{A}) = \mathbb{P}\big[\mathrm{Exp}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{SS}}(\mathcal{A}) \to 1\big],$$

где псевдокод эксперимента $\mathrm{Exp}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{SS}}$ определён на рис. 5.

Через $\mathrm{InSec}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{SS}}(t, q_{\mathrm{sign}})$ будем обозначать максимум среди всех преимуществ вида $\mathrm{Adv}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{SS}}(\mathcal{A})$, где максимум берётся по всем противникам $\mathcal{A}$ с ограничением t на вычислительные ресурсы и делающим не более q_{sign} запросов к оракулу $\mathcal{O}_{\mathrm{sig}}$.

Аналогичным образом можно определить модель UF-CMA, единственное отличие которой заключается в том, что в списке L хранятся только

$\mathrm{Exp}^{\mathsf{SUF\text{-}CMA}}_{\mathsf{SS}}(\mathcal{A})$

1 : $(pk, sk) \xleftarrow{\$} \mathsf{Kgen}()$
2 : $L \leftarrow [\,]$
3 : $(m, \sigma) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathrm{sig}}}(pk)$
4 : **if** $(m, \sigma) \notin L \wedge \mathsf{Verify}(vk, m, \sigma)$ **then**
5 : **return** 1
6 : **else**
7 : **return** 0

$\mathcal{O}_{\mathrm{sig}}(m)$

1 : $\sigma \xleftarrow{\$} \mathsf{Sign}(sk, m)$
2 : $L \leftarrow L \cup (m, \sigma)$
3 : **return** σ

Рис. 5

сообщения m. Иными словами, даже верная новая подпись σ для сообщения m, которое уже было подано на вход оракулу $\mathcal{O}_{\mathrm{sig}}$, будет не принята экспериментом.

В настоящей работе будут рассматриваться схемы подписи, полученные из Σ-протоколов. Σ-*протоколом* называется схема идентификации, в которой один из участников (доказывающий) должен доказать другой стороне (проверяющему) знание некоторого секрета без его раскрытия. Σ-протокол состоит из трёх сообщений — обязательство, вызов, ответ — и имеет вид, представленный на рис. 6.

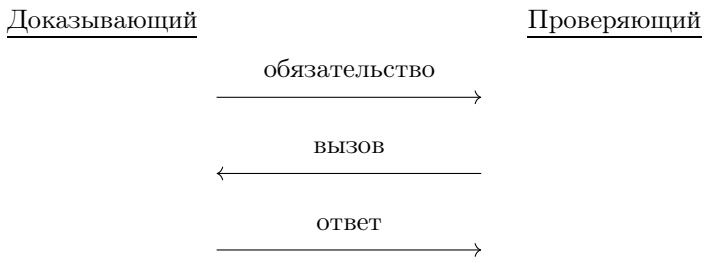


Рис. 6. Σ-протокол

В сообщении «обязательство» доказывающий отправляет результат некоторой (односторонней) функции от выбранных значений, что позволяет проверяющей стороне убедиться в том, что соответствующие значения доказывающего были выбраны до сообщения «вызов», не зная соответствующих значений до сообщения «ответ». В этом сообщении проверяющий выбирает некоторые значения, в соответствии с которыми доказывающий должен сформировать сообщение «ответ».

Используя преобразование Фиата — Шамира [11], можно получить схему подписи из Σ-протокола. Основная идея преобразования Фиата — Шамира состоит в отсутствии отправки первых двух сообщений и замене сообщения «вызов» значением $H(\text{«обязательство»} \parallel m)$, где H — это некоторая криптографическая хэш-функция, действующая в множество сообщений «вызов», а m — подписываемое сообщение.

2. Криптографические примитивы и базовые определения

2.1. Аддитивное быстрое преобразование Фурье. В криптосистемах, рассматриваемых в настоящей работе, используются многочлены над полем $\mathbb{F}_{2^m}$ характеристики 2. При этом часто требуется найти значения таких многочленов на множестве точек или во всех точках поля $\mathbb{F}_{2^m}$ (см., например, [4]).

Другими словами, для многочлена $f \in \mathbb{F}_{2^m}[x]$ требуется вычислить значения $f(\alpha_1), \ldots, f(\alpha_{2^m})$, где $\{\alpha_1, \ldots, \alpha_{2^m}\} = \mathbb{F}_{2^m}$. Для этого можно использовать алгоритм из [12], который называется *аддитивным быстрым преобразованием Фурье* и обозначается FFT (алгоритм 1). Стоит отметить, что в литературе имеются и другие вариации алгоритма FFT.

Алгоритм 1. Аддитивное быстрое преобразование Фурье (FFT)

Вход: многочлен $f \in \mathbb{F}_{2^m}[x]$, $\deg f < 2^m$, $m \geqslant 1$ целое, линейно независимые над $\mathbb{F}_2$ элементы $\beta_1, \ldots, \beta_m \in \mathbb{F}_{2^m}$, массив $B = \langle \beta_1, \ldots, \beta_m \rangle_{\mathbb{F}_2}$.

Выход: $\mathrm{FFT}(f, m, B) = (f(B[0]), \ldots, f(B[2^m - 1]))$, где $B[i] = c_1\beta_1 + \cdots + c_m\beta_m$ при $c_1 2^0 + c_2 2^1 + \cdots + c_m 2^{m-1} = i$.

if $m = 1$ **then**
  return $(f(0), f(\beta_1))$
$g(x) \leftarrow f(\beta_m x)$
найти $g_{0,i}, g_{1,i} \in \mathbb{F}_{2^m}$: $g(x) = \sum\limits_{i=0}^{2^{m-1}-1} (g_{0,i} + g_{1,i}x)(x^2 - x)^i$ ▷ [12, алгоритм 1]
$g_0(x) \leftarrow \sum\limits_{i=0}^{2^{m-1}-1} g_{0,i}x^i$, $g_1(x) \leftarrow \sum\limits_{i=0}^{2^{m-1}-1} g_{1,i}x^i$
for $i = 1, \ldots, m-1$ **do**
  $\gamma_i \leftarrow \beta_i\beta_m^{-1}$, $\delta_i \leftarrow \gamma_i^2 - \gamma_i$
$\Gamma \leftarrow \langle \gamma_1, \ldots, \gamma_{m-1} \rangle_{\mathbb{F}_2}$
$\Delta \leftarrow \langle \delta_1, \ldots, \delta_{m-1} \rangle_{\mathbb{F}_2}$
$(u_0, \ldots, u_{2^{m-1}-1}) \leftarrow \mathrm{FFT}(g_0, m-1, \Delta)$
$(v_0, \ldots, v_{2^{m-1}-1}) \leftarrow \mathrm{FFT}(g_1, m-1, \Delta)$
for $i = 0, \ldots, 2^{m-1} - 1$ **do**
  $w_i = u_i + v_i\Gamma[i]$
  $w_{i+2^{m-1}} = w_i + v_i$
return $(w_0, \ldots, w_{2^m-1})$

2.2. Методы декодирования кодов Гоппы.

Алгоритм Берлекэмпа — Мэсси. Будем использовать интерпретацию алгоритма Берлекэмпа — Мэсси, представленную здесь в виде алгоритма 2.

Алгоритм 2. Алгоритм Берлекэмпа — Мэсси

Вход: последовательность $a_1, \ldots, a_n \in \mathbb{F}_q$.
Выход: многочлен $f(x) = 1 + f_1 x + \cdots + f_L x^L$ минимальной степени такой, что $a_j + f_1 a_{j-1} + \cdots + f_L a_{j-L} = 0$, $j = L+1, L+2, \ldots, n$.

$f(x) \leftarrow 1$, $b(x) \leftarrow 1$, $L \leftarrow 0$
for $r = 1, \ldots, n$ **do**
  $\Delta \leftarrow a_r + f_1 a_{r-1} + \cdots + f_L a_{r-L}$
  if $\Delta = 0$ **then**
    $b(x) \leftarrow xb(x)$.
  else
    $b'(x) \leftarrow f(x) - \Delta x b(x)$
    if $2L < r$ **then**
      $b(x) \leftarrow \Delta^{-1} f(x)$
      $f(x) \leftarrow b'(x)$
      $L \leftarrow r - L$
    else
      $b(x) \leftarrow xb(x)$
      $f(x) \leftarrow b'(x)$
return $f(x)$

Существуют вариации алгоритма, минимизирующие число обращений элементов поля, а также вариации, использующие альтернативное описание. Возможно сделать реализацию для декодирования кодов, ориентированную на операцию расшифрования в кодовых криптосистемах, в которой искомый многочлен находится за постоянное время.

Алгоритм Берлекэмпа — Мэсси применим для решения системы уравнений

$$\begin{bmatrix} a_n & a_{n-1} & \ldots & a_1 \\ a_{n+1} & a_n & \ldots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{2n-1} & a_{2n-2} & \ldots & a_n \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} -a_{n+1} \\ -a_{n+2} \\ \vdots \\ -a_{2n} \end{bmatrix}.$$

Декодирование двоичного кода Гоппы на основе алгоритма Берлекэмпа — Мэсси.

Вход: синдром $s = Hy^\top \in \mathbb{F}_2^t$ принятого сообщения $y \in \mathbb{F}_2^n$, многочлен $g \in \mathbb{F}_{2^m}[x]$, $\deg g = t$, и элементы $\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{F}_{2^m}$, определяющие код Гоппы $\Gamma(g, \alpha_0, \ldots, \alpha_{n-1}) \subseteq \mathbb{F}_2^n$.

Выход: вектор $e \in \mathbb{F}_2^n$ такой, что $He^\top = 0$ и $\mathrm{wt}(y-e) \leqslant t$, при этом $H(y-e)^\top = s$

Шаг 1. Вычислить проверочную матрицу двойного размера $2t \times n$:

$$H_{(2)} = \begin{bmatrix} \frac{1}{g^2(\alpha_0)} & \frac{1}{g^2(\alpha_1)} & \cdots & \frac{1}{g^2(\alpha_{n-1})} \\ \frac{\alpha_0}{g^2(\alpha_0)} & \frac{\alpha_1}{g^2(\alpha_1)} & \cdots & \frac{\alpha_{n-1}}{g^2(\alpha_{n-1})} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\alpha_0^{2t-1}}{g^2(\alpha_0)} & \frac{\alpha_1^{2t-1}}{g^2(\alpha_1)} & \cdots & \frac{\alpha_{n-1}^{2t-1}}{g^2(\alpha_{n-1})} \end{bmatrix}.$$

Шаг 2. Привести матрицу $H_{(2)}$ к двоичному виду $H'_{(2)}$, заменив каждый элемент двоичным столбцом длины m согласно используемому представлению поля (неприводимый многочлен задан в спецификациях криптосистем).

Шаг 3. Вычислить синдром $s_{(2)} = H'_{(2)}(s, 0, \ldots, 0)^\top$ длины $2t$.

Шаг 4. С помощью алгоритма Берлекэмпа — Мэсси по синдрому $s_{(2)}$ найти многочлен — локатор ошибок $\sigma(x)$ такой, что $y_i - e_i = 1$ тогда и только тогда, когда $\sigma(\alpha_i) = 0$ при $i = 0, \ldots, n-1$.

Шаг 5. Найти $\deg\sigma$ корней многочлена $\sigma(x)$, например, последовательной подстановкой в него ненулевых элементов поля $\mathbb{F}_{2^m}$.

Шаг 6. Учитывая систематическую форму используемых в криптосистемах в качестве открытого ключа матриц, корней многочлена $\sigma(x)$ достаточно, чтобы подсчитать e, так как в качестве y подходит вектор $(s, 0, \ldots, 0)$ размерности n.

Заметим, что в криптосистеме CME шаг 5 выполняется с использованием аддитивного быстрого преобразования Фурье (см. алгоритм 1).

2.3. Расширенный алгоритм Евклида. Представляет собой естественное обобщение обычного алгоритма Евклида и изложен в виде алгоритма 3.

2.4. Метод Гаусса. Используется для приведения матрицы к систематическому виду, что требуется, например, для определения открытых ключей в системе CME. Соответствующий алгоритм представлен в виде алгоритмов 4 и 5: исключение столбца и собственно метод Гаусса.

2.5. Алгоритм расщепления носителя. Предложенный Сандрие в работе [13] алгоритм позволяет для линейного $[n, k, d]_q$-кода $\mathcal{C}$ с тривиальной группой автоморфизмов и перестановочно эквивалентного ему кода $\mathcal{D}$ эффективно определять перестановку $\pi \in S_n$, переводящую $\mathcal{C}$ в $\mathcal{D}$. Хорошо работает на линейных кодах, похожих на случайные. Многие рассматриваемые далее атаки используют этот алгоритм как одну

Алгоритм 3. Расширенный алгоритм Евклида (EEA)

Вход: многочлены $a(z)$, $b(z)$: $\deg a \geqslant \deg b$, d_{fin}.
Выход: многочлены $u(z)$, $r(z)$: $r(z) = b(z)u(z) \bmod a(z)$, $\deg r \leqslant d_{\text{fin}}$.

$r_{-1}(z) \leftarrow a(z)$, $r_0(z) \leftarrow b(z)$
$u_{-1}(z) \leftarrow 1$, $u_0(z) \leftarrow 0$
$i \leftarrow 0$
while $\deg r_i(z) > d_{\text{fin}}$ **do**
    $i \leftarrow i + 1$
    $q_i(z) \leftarrow r_{i-2}(z) \operatorname{div} r_{i-1}(z)$
    $r_i(z) \leftarrow r_{i-2}(z) \bmod r_{i-1}(z)$ ▷ $r_{i-2}(z) = q_i(z)r_{i-1}(z) + r_i(z)$
    $u_i(z) \leftarrow u_{i-2}(z) - q_i(z)u_{i-1}(z)$
$N \leftarrow i$
return $u_N(z)$, $r_N(z)$

Алгоритм 4. Исключение столбца EliminateColumn(H_j, j)

Вход: матрица $H_j \in \mathbb{F}_2^{tm \times n}$, номер столбца $j \in \{1, \ldots, tm\}$.
Выход: матрица $H_{j+1} \in \mathbb{F}_2^{tm \times n}$: $H_{j+1}[:, j] = e_j^\top$, $H_{j+1} = G_j H_j$ для некоторой обратимой матрицы $G_j \in \mathbb{F}_2^{tm \times tm}$, или $\perp$.

$H_{j+1} \leftarrow H_j$
if $H_{j+1}[j, j] \neq 1$ **then** ▷ убедиться, что $H_{j+1}[j, j] = 1$
    $k \leftarrow \min\{j + 1 \leqslant r \leqslant tm \mid H_{j+1}[r, j] = 1\}$
    if такое k не существует **then**
        return $\perp$
    $H_{j+1}[j, :] \leftarrow H_{j+1}[j, :] + H_{j+1}[k, :]$ ▷ прибавить k-ю строку к j-й
for $i \in \{1, \ldots, tm\} \setminus \{j\}$ **do**
    if $H_{j+1}[i, j] = 1$ **then** ▷ убедиться, что $H_{j+1}[i, j] = 0$ при $i \neq j$
        $H_{j+1}[i, :] \leftarrow H_{j+1}[i, :] + H_{j+1}[j, :]$ ▷ прибавить j-ю строку к i-й
return H_{j+1}

Алгоритм 5. Метод Гаусса

Вход: матрица $H \in \mathbb{F}_2^{tm \times n}$: $\operatorname{rk} H[1{:}tm, 1{:}tm] = tm$.
Выход: систематическая форма $H' = (I_{tm} \mid A) \in \mathbb{F}_2^{tm \times n}$ матрицы H.

$H_1 \leftarrow H$
for $j = 1, \ldots, tm$ **do**
    $H_{j+1} \leftarrow \text{EliminateColumn}(H_j, j)$
return $H' \leftarrow H_{tm+1}$

из своих составных частей. В рамках настоящей работы нет необходимости в его детальном описании.

2.6. Декодирование по информационным совокупностям. Алгоритм, обозначаемый ISD и представленный в виде алгоритма 6 в соответствии с работой Штерна [14], находит кодовое слово веса w в $[n,k]$-коде $\mathcal{C}$ с порождающей матрицей $G \in \mathbb{F}_2^{k\times n}$. В дополнение на вход алгоритма подаются параметры $p \in \{0,1,\ldots,w\}$ и $\ell \in \{0,1,\ldots,n-k\}$. Также фиксируется маскирующая функция $\varphi(x) = x_{k+1}x_{k+2}\cdots x_{k+\ell}$.

Алгоритм 6. Алгоритм Штерна

Вход: порождающая матрица $G \in \mathbb{F}_2^{k\times n}$, параметры $w, p, \ell \in \mathbb{N}$.
Выход: кодовое слово $c \in \mathcal{C}$ веса w.

$L \leftarrow \varnothing$
repeat
  применить случайную перестановку столбцов π к матрице G
  привести полученную матрицу $\pi(G)$ к виду $G' = (I_k \mid A)$
  $L_1 \leftarrow \varnothing$, $L_2 \leftarrow \varnothing$
  for $u \in \{v \in \mathbb{F}_2^{k/2} \mid \mathrm{wt}(v) = p\}$ **do**
    добавить $x = (u \parallel 0)G'$ в список L_1
    добавить $x' = (0 \parallel u)G'$ в список L_2
  отсортировать список L_1 согласно функции $\varphi(x)$
  отсортировать список L_2 согласно функции $\varphi(x)$
  for $x \in L_1$ **do**
    for $x' \in L_2$ **do**
      if $\varphi(x) = \varphi(x')$ **then**
        добавить вектор $(x \parallel x')$ в список L
until существует $(x \parallel x') \in L$ такой, что $\mathrm{wt}(x) = w - 2p$
if $x = (u \parallel 0 \parallel x_{k+1}\ldots x_n) \wedge x' = (0 \parallel u' \parallel x'_{k+1}\ldots x'_n)$ **then**
  return $c \leftarrow (u \parallel u')G$

3. Теоретико-кодовые криптосистемы

Теоретико-кодовыми криптосистемами обычно называют системы, которые основаны на задачах помехоустойчивого кодирования. В настоящее время криптосистемы такого типа представляют повышенный интерес в связи с развитием постквантовой криптографии.

Первой криптосистемой кодового типа является криптосистема Мак-Элиса [3], разработанная в 1978 г. Основная идея данной криптосистемы состоит в маскировке некоторого линейного кода под код, не обладающий видимой алгебраической и комбинаторной структурой. Как известно, общая задача декодирования линейного кода NP-трудна [7], однако, зная структуру такого кода, можно легко расшифровать сообщение. В оригинальном варианте криптосистемы в качестве кодов, исправляющих ошибки, используются двоичные коды Гоппы [8].

Далее, Нидеррайтер в 1986 г. предложил свой вариант кодовой криптосистемы [2], основанный на обобщённых кодах Рида — Соломона. Этот вариант оказался нестойким: в 1992 г. В. М. Сидельников и С. О. Шестаков опубликовали атаку на криптосистему Нидеррайтера [15]. Авторы полагали, что их атака применима и к двоичным кодам Гоппы, но это оказалось не так, и вариант криптосистемы Нидеррайтера на двоичных кодах Гоппы всё ещё не взломан [16].

Криптосистема Мак-Элиса, как и многие другие кодовые криптосистемы, обладает значимым преимуществом — высокой скоростью зашифрования и расшифрования. Однако её основной недостаток — большой размер открытого ключа. В связи с этим предпринято множество попыток модифицировать криптосистему с применением разных семейств кодов. Абсолютное большинство таких попыток не увенчалось успехом. Такие системы обычно взламывают, опираясь на структуру используемого кода, или их стойкость не превосходит стойкости криптосистем на кодах Гоппы. Так, криптосистема Сидельникова на основе кодов Рида — Маллера, предложенная в 1994 г. [17], взломана в 2007 г. [18]. Различные вскрытые модификации системы Мак-Элиса описаны в монографии [19].

В настоящее время Национальный институт стандартов и технологий США (NIST) проводит ряд конкурсов по выбору квантово-устойчивых алгоритмов с целью стандартизации набора схем инкапсуляции ключа и цифровой подписи. В четвёртом раунде конкурса NIST [21] участвовали четыре криптосистемы KEM, три из которых кодовые — CME, BIKE и HQC, а последняя признана ненадёжной. CME использует идею Нидеррайтера, при этом названа в честь Мак-Элиса в силу использования кодов Гоппы; BIKE основан на квазициклических кодах (MDPC); в различных же модификациях HQC могут использоваться также квазициклические коды. В работе [20] приведён анализ производительности данных криптосистем, из которого становится ясно, что наиболее производительной с точки зрения реализации является HQC, демонстрирующая наименьшее время выполнения и наименьший размер ключа среди трёх систем. CME наименее производительная, вместе с тем наиболее изученная. Однако, вопреки ожиданиям многих специалистов, в качестве стандарта была выбрана криптосистема HQC. Обратим внимание, что обзор третьего раунда конкурса NIST представлен в [22], а в обзоре второго раунда [23] отмечается особая важность исследования стойкости систем к атакам по побочным каналам.

Отметим, что атаки по побочным каналам на теоретико-кодовые криптосистемы принципиально не отличаются от атак на любые другие. Идея заключается в выявлении алгоритма, допускающего утечки по побочным каналам, и исследовании возможности использования полученной информации для восстановления секрета.

ME.Kgen()

$g(x) \xleftarrow{\mathcal{U}} \{g \in \mathbb{F}_{2^m}[x] \mid g \text{ неприводимый}, \deg g = t\}$

$(\alpha_0, \ldots, \alpha_{n-1}) \xleftarrow{\mathcal{U}} \{\alpha \in \mathbb{F}_{2^m}^n \mid \alpha_i \neq \alpha_j \text{ при } i \neq j\}$

$G \leftarrow$ порождающая матрица кода $\Gamma(g, \alpha_0, \ldots, \alpha_{n-1})$

$k \leftarrow \dim \Gamma =$ длина сообщения m

$S \xleftarrow{\mathcal{U}} \mathrm{GL}_k(2)$

$P \xleftarrow{\mathcal{U}}$ множество перестановочных матриц порядка n

$G_{\mathrm{pub}} \leftarrow SGP$

return $(pk = G_{\mathrm{pub}},\ sk = (g, \alpha_0, \ldots, \alpha_{n-1}, S, P))$

ME.Enc(pk, m)

$e \xleftarrow{\mathcal{U}} \{e \in \mathbb{F}_2^n \mid \mathrm{wt}(e) = t\}$

$c \leftarrow mG_{\mathrm{pub}} + e$

return c

ME.Dec(sk, c)

$c' \leftarrow cP^{-1}$

найти m': $m'G + e' = c', \mathrm{wt}(e') \leqslant t$

// алгоритм декодирования кода Гоппы Γ

if m' найдено $\wedge\ \mathrm{wt}(e') = t$ **then**

  $m \leftarrow m'S^{-1}$

  return m

else

  return $\perp$

Рис. 7. Схема шифрования Мак-Элиса [3]

4. Криптосистема Classic McEliece

4.1. Обозначения:

- n — длина кода, количество точек для кода Гоппы;
- $q = 2^m$ — размер поля;
- через m в криптосистемах также обозначено сообщение, но из контекста всегда ясно, что подразумевается;
- t — степень многочлена Гоппы и вес вектора ошибок; необходимо, чтобы $mt < n$;
- $k = n - mt$ — размерность кода Гоппы для используемых параметров;
- $f(z) \in \mathbb{F}_2[z]$ — неприводимый нормированный многочлен степени m, определяющий поле $\mathbb{F}_q$ в виде элементов кольца $\mathbb{F}_2[z]/(f(z))$ и двоичных векторов из $\mathbb{F}_2^m$, образованных коэффициентами остатков от деления на многочлен $f(z)$; в большинстве описаний будет опущен;

NIED.Kgen()

1 : $g(x) \xleftarrow{\mathcal{U}} \{g \in \mathbb{F}_{2^m}[x] \mid g \text{ неприводимый}, \deg g = t\}$
2 : $(\alpha_0, \ldots, \alpha_{n-1}) \xleftarrow{\mathcal{U}} \left\{\alpha \in \mathbb{F}_{2^m}^n \mid \alpha_i \neq \alpha_j \text{ при } i \neq j\right\}$
3 : $h_{i,j} \leftarrow \alpha_j^i / g(\alpha_j)$, $i \in [0, t-1]$
4 : $(\beta_{i,j,0}, \ldots, \beta_{i,j,m-1}) \leftarrow h_{i,j}$ // $\beta_{i,j,l} \in \mathbb{F}_2$
5 : $\widehat{h}_{im+l,j} \leftarrow \beta_{i,j,l}$
6 : $\widehat{H} \leftarrow (\widehat{h}_{i,j}) \in \mathbb{F}_2^{mt \times n}$
7 : представить $\widehat{H}$ в виде $(I_{n-k} \mid H')$, $H' \in \mathbb{F}_2^{mt \times mt}$
8 : **return** $(pk = H', sk = (g, \alpha_0, \ldots, \alpha_{n-1}))$

NIED.Enc(pk, m)

1 : $m \in \mathcal{M}_{n,t}$
2 : $c \leftarrow (I_{n-k} \mid H')m^\top$
3 : **return** $c^\top$

NIED.Dec(sk, c)

1 : $H \leftarrow (I_{n-k} \mid H')$
2 : $c' = c \parallel 0^k$
3 : найти e: $He^\top = 0 \wedge \mathrm{wt}(c' \oplus e) \leqslant t$
4 : // алгоритм Берлекэмпа — Мэсси
5 : **if** e найден $\wedge\ \mathrm{wt}(c' + e) = t$ **then**
6 : **return** $m \leftarrow c' + e$
7 : **else**
8 : **return** $\perp$

Рис. 8. Схема шифрования Нидеррайтера [2]

• $\mathcal{M}_{n,t} = \left\{x \in \mathbb{F}_2^n \mid \mathrm{wt}(x) = t\right\}$ — множество сообщений для криптосистем NIED, CME, CODI.

4.2. Оригинальная криптосистема Мак-Элиса (ME). Криптосистема с открытым ключом (PKE), представленная на рис. 7, предложена Мак-Элисом в 1978 г. [3].

Однако обратим внимание, что криптосистема, поданная на конкурс NIST [4] под названием Classic McEliece, построена на базе криптосистемы Нидеррайтера, при этом она также использует коды Гоппы.

4.3. Криптосистема Нидеррайтера. На рис. 8 приведено описание криптосистемы Нидеррайтера [2], которая, как и оригинальная криптосистема Мак-Элиса, относится к классу PKE. Данное описание согласовано с описанием криптосистемы CME, в том числе в нём используется код Гоппы.

Обратим внимание, что сообщениями в данной криптосистеме являются элементы $\mathcal{M}_{n,t}$, т. е. двоичные векторы размерности n и веса t.

4.4. Криптосистема CME. Перейдём к описанию криптосистемы, получившей название Classic McEliece. Её спецификация [4] датируется 23.10.2022 г. Согласно заявке авторами криптосистемы являются исследователи из более чем 10 организаций.

В режиме PKE её процедуры Enc и Dec совпадают с описанными ранее NIED.Enc и NIED.Dec. Далее опишем возможные параметры криптосистемы, CME.Kgen() и функции KEM.

Заданные в спецификации значения параметров отражены в табл. 1 (их описание см. в п. 4.1). Пустые μ, ν можно интерпретировать как $(\mu, \nu) = (0, 0)$ (или (i, i) для любого $0 \leqslant i < n$).

Таблица 1

Параметры криптосистемы CME

Бит. стой-кость	Название	m	t	n	k	μ, ν	Откр. ключ, МБ	Секр. ключ, КБ	Шифр-текст, Б
128	mceliece348864	12	64	3488	2720	—	0,25	6,34	96
128	mceliece348864f	12	64	3488	2720	32, 64	0,25	6,34	96
192	mceliece460896	13	96	4608	3360	—	0,5	13,29	156
192	mceliece460896f	13	96	4608	3360	32, 64	0,5	13,29	156
256	mceliece6688128	13	128	6688	5024	—	1	13,61	208
256	mceliece6688128f	13	128	6688	5024	32, 64	1	13,61	208
256	mceliece6960119	13	119	6960	5413	—	1	13,63	194
256	mceliece6960119f	13	119	6960	5413	32, 64	1	13,63	194
256	mceliece8192128	13	128	8192	6528	—	1,3	13,79	208
256	mceliece8192128f	13	128	8192	6528	32, 64	1,3	13,79	208

Для описания процедуры генерации ключей CME.Kgen() определим следующие дополнительные параметры.

- H — криптографическая хэш-функция, возвращающая ℓ битов.
- Входы функции H будем записывать следующим образом: $\mathsf{H}(b, v, c)$, где $b \in \mathbb{F}_2$, $v \in \mathbb{F}_2^n$ и $c \in \mathbb{F}_2^{mt}$ — шифртекст. Кодирование параметров в единую строку для хэширования происходит следующим образом: число b представляется 1 байтом, вектор v — $\lceil n/8 \rceil$ байтами, вектор c — $\lceil mt/8 \rceil$ байтами.
- G — криптографический генератор псевдослучайных битов, принимающий на вход двоичную строку длины ℓ (т. е. имеет seed размера ℓ).

В спецификации предлагается использовать $\ell = 256$, а в качестве выхода H брать первые ℓ бит выхода функции SHAKE256. В качестве G также используется SHAKE256: на вход ей подаётся байт, в котором записано число 64, за которым следуют ещё $\lceil \ell/8 \rceil$ байтов, содержащих ℓ-битный вход генератора G (итого 33 байта).

Процедура CME.$\mathsf{Kgen}(\delta)$

Вход: $\delta \in \mathbb{F}_2^\ell$.

Выход: (pk, sk) — ключевая пара.

Шаг 1. Получить $(s, r, \delta') = \mathsf{G}(\delta)$, где $\delta' \in \mathbb{F}_2^\ell$, $s \in \mathbb{F}_2^n$, а длина двоичной строки r достаточна для генерации остальных параметров (конкретное её значение несущественно в контексте данной работы).

Шаг 2. Определить порядок $\alpha_0, \ldots, \alpha_{q-1}$ элементов поля $\mathbb{F}_q$ и неприводимый многочлен g при помощи строки r. Если что-либо не удалось сгенерировать корректно, то вся процедура CME.Kgen перезапускается с $\delta = \delta'$.

Шаг 3. Построить для кода Гоппы $\Gamma(\alpha_0, \ldots, \alpha_{n-1})$ проверочную матрицу H.

Шаг 4. Привести матрицу H к систематическому или (μ, ν)-полусистематическому виду (в зависимости от выбранных параметров криптосистемы). Результатом шага 4 является набор $(T, c_{mt-\mu}, \ldots, c_{mt-1}, \Gamma')$ такой, что

- $c_{mt-\mu}, \ldots, c_{mt-1}$ — номера ведущих столбцов получившейся (μ, ν)-полусистематической матрицы ($c_i = i$ для систематической матрицы);
- Γ' — код Гоппы, полученный из Γ перестановкой столбцов с номерами $c_{mt-\mu}, \ldots, c_{mt-1}$ в позиции $mt-\mu, \ldots, mt-1$, для этого достаточно перенумеровать используемые точки $\alpha_0, \ldots, \alpha_{n-1}$ в $\alpha'_0, \ldots, \alpha'_{n-1}$;
- проверочная матрица кода Γ' представима в систематическом виде $(I_{mt} \mid T)$, где I_{mt} — единичная матрица порядка mt.

Аналогично шагу 2 вся процедура CME.Kgen перезапускается с $\delta = \delta'$, если проверочную матрицу кода Γ' не удалось привести к нужному виду. На шаге 4 используется метод Гаусса (см. п. 2.4), и на его реализацию направлена одна из атак по сторонним каналам, рассматриваемых ниже.

Шаг 5. $pk \leftarrow T$.

Шаг 6. $sk \leftarrow (\delta, c, g, \alpha', s)$, где

- $c = (c_{mt-\mu}, \ldots, c_{mt-1})$,
- $\alpha' = (\alpha'_0, \ldots, \alpha'_{n-1}, \alpha_n, \alpha_{q-1})$ с учётом возможной перенумерации $\alpha_0, \ldots, \alpha_{n-1}$ в $\alpha'_0, \ldots, \alpha'_{n-1}$ на шаге 4.

CME.$\mathsf{Encaps}(pk)$

1 : $m \xleftarrow{\mathcal{U}} \mathcal{M}_{n,t}$
2 : $c \leftarrow \text{NIED.}\mathsf{Enc}(pk, m)$
3 : $K \leftarrow \mathsf{H}(1, m, c)$
4 : **return** (K, c)

CME.$\mathsf{Decaps}(sk, c)$

1 : $(sk_{\text{NIED}}, s) \leftarrow sk$
2 : $m \leftarrow \text{NIED.}\mathsf{Dec}(sk_{\text{NIED}}, c)$
3 : **if** $m \neq \perp$ **then**
4 : **return** $K \leftarrow \mathsf{H}(1, m, c)$
5 : **else**
6 : **return** $K \leftarrow \mathsf{H}(0, s, c)$

Рис. 9. Схема Classic McEliece [4]

Сообщения в криптосистеме CME представляются векторами из $\mathcal{M}_{n,t}$. На рис. 9 приведены её KEM-функции CME.Encaps и CME.Decaps. Отметим, что в актуальной версии криптосистемы [4] эти функции упрощены по сравнению со спецификацией 2020 г., представленной на третий раунд конкурса NIST [24]. Один из аргументов, обозначенных в рамках четвёртого раунда конкурса NIST [25], — сделать отличия от существующего патента U.S. 9912479 ещё более явными.

4.5. Известные реализации.

- Официальная реализация [4] и несколько неофициальных, ссылки на которые размещены на странице [26].
- Наработки в рамках библиотеки CIRCL [27, 28].
- Реализация для OpenSSH, включённая в библиотеку liboqs [29] и основанная на программном коде PQClean [30].
- Реализация для VPN-протокола WireGuard [31].
- Свободные реализации [32–34].
- Много информации представлено в презентации Бернштейна [35].

5. Криптосистема «Кодиеум»

Схема инкапсуляции ключа «Кодиеум» была разработана в рамках деятельности рабочей группы 2.5 «Постквантовые криптографические механизмы» Технического комитета по стандартизации «Криптографическая защита информации» и представлена на XXVI Международной научно-практической конференции «РусКрипто — 2024» [5]. Как и в Classic McEliece (см. рис. 9), в Кодиеуме используется схема шифрования Нидеррайтера (см. рис. 8). Для получения итоговой схемы используется FO-преобразование $\mathsf{U}^{\not\perp}_m$ в отличие от используемого в CME преобразования $\mathsf{U}^{\not\perp}$. Схема криптосистемы «Кодиеум» представлена на рис. 10. В этой криптосистеме применяется хэш-функция $h\colon \{0,1\}^* \to \{0,1\}^{512}$ — Стрибог-512 [36].

CODI.Kgen()

1 : $(pk, sk) \leftarrow \text{NIED.Kgen}()$
2 : $s \xleftarrow{\mathcal{U}} \mathcal{M}$
3 : $sk' \leftarrow (sk, s)$
4 : **return** (pk, sk')

CODI.Encaps(pk)

1 : $m \xleftarrow{\mathcal{U}} \mathcal{M}$
2 : $c \leftarrow \text{NIED.Enc}(pk, m)$
3 : $K \leftarrow h(0 \parallel m)$
4 : **return** (K, c)

CODI.Decaps(sk')

1 : $(sk, c) \leftarrow sk'$
2 : $m' \leftarrow \text{NIED.Dec}(sk, c)$
3 : **if** $m' \neq \perp$ **then**
4 : **return** $K \leftarrow h(0 \parallel m')$
5 : **else**
6 : **return** $K \leftarrow h(1 \parallel s \parallel c)$

Рис. 10. Схема «Кодиеум» [5]

В [37] с помощью результатов из [10] получены оценки стойкости схемы инкапсуляции ключа «Кодиеум». Однако, как и для криптосистемы Classic McEliece, оценки стойкости могут быть улучшены, что следует из недавних результатов [38]. Данная криптосистема находится на начальной стадии стандартизации, из чего следует, что возможны изменения в строении схемы, в том числе в выборе класса используемых кодов. Авторам настоящей работы не удалось найти реализации криптосистемы «Кодиеум», что свойственно для недавно представленных криптосистем.

В табл. 2 представлены параметры схемы «Кодиеум» и соответствующие им размеры ключей и шифртекстов для различных уровней битовой стойкости в модели QROM. Отметим, что как указано ранее, данные параметры могут быть улучшены в свете недавней работы [38].

Таблица 2

Параметры схемы «Кодиеум»

Битовая стойкость	m	n	k	t	Открытый ключ, МБ	Секретный ключ, КБ	Шифртекст, Б
128	13	16960	14230	210	4,63	29,32	341
192	13	31620	27902	286	12,37	54,49	465
256	13	51980	47404	352	25,75	95,12	574

Секретная информация: $s \xleftarrow{\mathcal{U}} \mathcal{M}_{n,t}$.
Открытая информация: $H \xleftarrow{\mathcal{U}} \mathbb{F}_2^{(n-k)\times n}$, $y^\top \leftarrow Hs^\top$.

	Доказывающий		Проверяющий
1 :	$u \xleftarrow{\mathcal{U}} \mathbb{F}_2^n$, $\sigma \xleftarrow{\mathcal{U}} S_n$		
2 :	$c_0 \leftarrow h(\sigma \parallel Hu^\top)$		
3 :	$c_1 \leftarrow h(\sigma(u))$		
4 :	$c_2 \leftarrow h(\sigma(u \oplus s))$		
5 :		$\xrightarrow{c_0, c_1, c_2}$	
6 :			$b \xleftarrow{\mathcal{U}} \{0, 1, 2\}$
7 :		$\xleftarrow{b}$	
8 :	**if** $b = 0$ **then**		
9 :	$r_0 \leftarrow \sigma$, $r_1 \leftarrow u$		
10 :	**if** $b = 1$ **then**		
11 :	$r_0 \leftarrow \sigma$, $r_1 \leftarrow u \oplus s$		
12 :	**if** $b = 2$ **then**		
13 :	$r_0 \leftarrow \sigma(u)$, $r_1 \leftarrow \sigma(s)$		
14 :		$\xrightarrow{r_0, r_1}$	
15 :			**if** $b = 0$ **then**
16 :			// проверить
17 :			$c_0 = h(r_0 \parallel Hr_1^\top)$
18 :			$c_1 = h(r_0(r_1))$
19 :			**if** $b = 1$ **then**
20 :			// проверить
21 :			$c_0 = h(r_0 \parallel (Hr_1^\top \oplus y))$
22 :			$c_2 = h(r_0(r_1))$
23 :			**if** $b = 2$ **then**
24 :			// проверить
25 :			$c_1 = h(r_0)$
26 :			$c_2 = h(r_0 \oplus r_1)$
27 :			$\mathrm{wt}(r_1) = t$

Рис. 11. Схема идентификации Штерна [39]

6. Криптосистема «Шиповник»

Так же, как и схема инкапсуляции ключа «Кодиеум», схема подписи «Шиповник» была разработана в рамках деятельности рабочей группы 2.5 «Постквантовые криптографические механизмы» Технического комитета по стандартизации «Криптографическая защита информации». Впервые эта схема представлена без названия на XXIII Международной научно-практической конференции «РусКрипто — 2021» [6]. В основе схемы подписи «Шиповник» лежит схема идентификации Штерна [39] (рис. 11), в которой в качестве хэш-функции $h\colon \{0,1\}^* \to \{0,1\}^{512}$ применяется хэш-функция Стрибог-512 [36].

В [39] представлена стратегия, при которой противник, не знающий секретную информацию, может успешно пройти схему идентификации с вероятностью $2/3$. Тогда для уменьшения вероятности успешной атаки

SHIP.Kgen()

$s \xleftarrow{\mathcal{U}} \mathcal{M}_{n,t}$
$y \leftarrow Hs^\top$
return $(pk = y,\ sk = s)$

SHIP.Verify$(pk, m, (c \parallel r))$

$b \leftarrow h'(m \parallel c)$
for $i = 0, \ldots, \delta - 1$ **do**
  if $b_i = 0 \wedge$
    $\wedge\, (c_{i,0} \neq h(r_{i,0} \parallel Hr_{i,1}^\top) \vee$
    $\vee\, c_{i,1} \neq h(r_{i,0}(r_{i,1})))$ **then**
    return 0
  if $b_i = 1 \wedge$
    $\wedge\, (c_{i,0} \neq h(r_{i,0} \parallel (Hr_{i,1}^\top \oplus y)) \vee$
    $\vee\, c_{i,2} \neq h(r_{i,0}(r_{i,1})))$ **then**
    return 0
  if $b_i = 2 \wedge$
    $\wedge\, (c_{i,1} \neq h(r_{i,0}) \vee$
    $\vee\, c_{i,2} \neq h(r_{i,0} \oplus r_{i,1}) \vee$
    $\vee\, \mathrm{wt}(r_{i,1}) \neq t)$ **then**
    return 0
return 1

SHIP.Sign(sk, m)

for $i = 0, \ldots, \delta - 1$ **do**
  $u_i \xleftarrow{\mathcal{U}} \mathbb{F}_2^n$
  $\sigma_i \xleftarrow{\mathcal{U}} S_n$
  $c_{i,0} \leftarrow h(\sigma_i \parallel Hu_i^\top)$
  $c_{i,1} \leftarrow h(\sigma(u_i))$
  $c_{i,2} \leftarrow h(\sigma(u_i \oplus s))$
  $c_i \leftarrow c_{i,0} \parallel c_{i,1} \parallel c_{i,2}$
$c \leftarrow c_0 \parallel \cdots \parallel c_{\delta-1}$
$b \leftarrow h'(m \parallel c)$
for $i = 0, \ldots, \delta - 1$ **do**
  if $b_i = 0$ **then**
    $r_i \leftarrow \sigma_i \parallel u_i$
  if $b_i = 1$ **then**
    $r_i \leftarrow \sigma_i \parallel (u_i \oplus s)$
  if $b_i = 2$ **then**
    $r_i \leftarrow \sigma_i(u_i) \parallel \sigma_i(s)$
$r \leftarrow r_0 \parallel \cdots \parallel r_{\delta-1}$
return $c \parallel r$

Рис. 12. Схема «Шиповник» [40]

следует повторить протокол δ раз. Для перехода от схемы идентификации к схеме подписи используется преобразование Фиата — Шамира (см. п. 1.2). Для этого необходима троичная хэш-функция $h'\colon \{0,1\}^* \to \{0,1,2\}^\delta$, определённая следующим образом:

$$h'(x) = \left\lfloor \frac{h(x) \cdot 3^\delta}{2^{512}} \right\rfloor,$$

где двоичные и троичные строки естественным образом отождествляются с натуральными числами. Схема подписи «Шиповник» представлена на рис. 12.

Стойкость данной криптосистемы анализируется в [40–42]. В табл. 3 приведены параметры схемы «Шиповник» и соответствующие им размеры открытого ключа и подписи для различных уровней битовой стойкости с учётом результатов доказуемой стойкости согласно работам [6, 40].

Таблица 3

Параметры схемы «Шиповник»

Битовая стойкость	n	k	t	δ	Открытый ключ, МБ	Подпись, МБ
80	2896	1448	318	137	0,25	0,62
128	4841	2421	533	219	0,70	1,75
256	8841	4421	973	438	2,33	6,78
512	16818	8409	1850	876	8,43	27,43

6.1. Известные реализации. Открытая реализация [43] отечественного постквантового алгоритма «Шиповник» компании «Криптонит» выполнена компанией QApp в рамках деятельности в составе рабочей группы «Постквантовые криптографические механизмы» Технического комитета 26 Росстандарта. Проект написан на языке C с оптимизацией под наборы команд SSE4.1, SSE2 и MMX.

7. Криптосистемы HQC и BIKE

Положим $\mathcal{R} = \mathbb{F}_2[X]/(X^r - 1)$ — кольцо циклических многочленов степени не выше $r-1$ с коэффициентами из поля $\mathbb{F}_2$. Простое p называется *примитивным*, если многочлен $(X^p - 1)/(X - 1)$ неприводим над $\mathbb{F}_2$.

Линейный код $\mathcal{C} \leqslant \mathbb{F}_2^n$ *квазициклический* (n_0-квазициклический), если циклический сдвиг произвольного кодового слова из $\mathcal{C}$ на n_0 позиций также является кодовым словом. Проверочная матрица $H \in \mathbb{F}_2^{r \times n}$ такого кода имеет блочный вид

$$H = (H_0 \mid H_1 \mid \cdots \mid H_{n_0-1}),$$

где r простое, $n = n_0 r$. Каждый блок $H_i \in \mathbb{F}_2^{r\times r}$ — циркулянтная матрица, в которой каждая строка получается циклическим сдвигом предыдущей строки на 1 позицию вправо (сдвиг последней строки приводит к первой). Благодаря изоморфизму между кольцом циркулянтных матриц размера $r \times r$ и кольцом многочленов $\mathcal{R}$ сложение и умножение матриц может выполняться в кольце многочленов. Под QC-MDPC-кодом понимается квазициклический код с проверочной матрицей, все строки которой имеют фиксированный вес $w = O(\sqrt{n})$.

Схемы инкапсуляции ключа BIKE и HQC, безопасные в модели IND-CCA, получаются из схем шифрования с открытым ключом, безопасных в модели IND-CPA. Приведём описание задач, на сложности решения которых основана безопасность схем HQC-PKE (задача QCSD) и BIKE-PKE (задачи QCSD и QCCF) в модели IND-CPA.

Задача 2 (задача синдромного декодирования квазициклического кода, QCSD). *Даны проверочная матрица квазициклического кода* $H \in \mathbb{F}_2^{(n-k)\times n}$, *синдром* $s \in \mathbb{F}_2^{n-k}$ *и целое* $t > 0$. *Найти вектор* $e \in \mathbb{F}_2^n$ *такой, что* $wt(e) \leqslant t$ *и* $He^\top = s$.

Задача 3 (задача поиска кодового слова квазициклического кода, QCCF). *Даны проверочная матрица* $H \in \mathbb{F}_2^{(n-k)\times n}$ *квазициклического кода и целое* $t > 0$. *Найти вектор* $c \in \mathbb{F}_2^n$ *такой, что* $wt(c) = t$ *и* $Hc^\top = 0$.

Основные виды атак, направленных на кодовые криптосистемы на основе квазициклических кодов, — декодирование на основе информационных совокупностей [44] (см. также п. 2.6) и его модификации, а также атаки, направленные на структуру кода [45] или вид многочлена, порождающего циклическую структуру [46, 47]. Помимо этого схемы могут быть уязвимы для GJS-атаки, представленной в работе [48] 2016 г., в случае повторного использования пары ключей. GJS, или реакционная атака, использует корреляцию между секретным ключом и паттернами ошибок, вызывающими отказ декодирования. Нескольких таких паттернов достаточно для проведения успешной атаки. В работе [49] показано, что основную вычислительную сложность имеет задача обнаружения первого такого паттерна. Атаки по побочным каналам, направленные на HQC и BIKE, описаны в разд. 9 ниже и в ч. 2 обзора.

7.1. Схема HQC. Hamming Quasi-Cyclic — схема инкапсуляции ключа на основе квазициклических кодов без скрытой структуры, которая была представлена на конкурс NIST [50]. Её основными преимуществами являются малый размер открытого ключа и точная оценка вероятности ошибки расшифрования. Кроме того, в отличие от большинства существующих кодовых криптосистем, основанных на схеме шифрования Мак-Элиса, стойкость схемы HQC опирается на сложность решения

HQC.KeyGen()

1 : $h \xleftarrow{\mathcal{U}} \mathcal{R}$
2 : $G \leftarrow$ порождающая матрица кода $\mathcal{C}$ // $G \in \mathbb{F}_2^{k \times n}$
3 : $sk \leftarrow (x, y) \xleftarrow{\$} \mathcal{R}_w \times \mathcal{R}_w$
4 : $pk \leftarrow (h, s = x + h \cdot y)$
5 : **return** (pk, sk)

HQC.Encrypt(pk, m)

1 : $m \xleftarrow{\mathcal{U}} \mathbb{F}_2^k$
2 : $e \xleftarrow{\mathcal{U}} \mathcal{R}_{w_e}$
3 : $r \leftarrow (r_1, r_2) \xleftarrow{\$} \mathcal{R}_{w_r} \times \mathcal{R}_{w_r}$
4 : $u \leftarrow r_1 + h \cdot r_2$
5 : $v \leftarrow \text{truncate}(mG + s \cdot r_2 + e, \ell)$
6 : $c \leftarrow (u, v)$
7 : **return** c

HQC.Decrypt(sk, c)

1 : $m' \leftarrow \mathcal{C}.\mathsf{Decode}(v - u \cdot y)$
2 : **return** m'

Рис. 13. Схема HQC-PKE

задачи QCSD, а не на скрытую структуру кода, исправляющего ошибки. Актуальная на данный момент спецификация обновлена 19.02.2025 г. [51].

В схеме HQC используются два вида кодов. Первый — открытый известный $[n, k]$-код $\mathcal{C} \in \mathbb{F}_2^n$, исправляющий Δ ошибок и являющийся конкатенацией кодов Рида — Маллера и Рида — Соломона, для которого имеются эффективные алгоритмы кодирования $\mathcal{C}.\mathsf{Encode}$ и декодирования $\mathcal{C}.\mathsf{Decode}$. Второй — случайный квазициклический код длины $2n$ и размерности n с проверочной матрицей $(I, \operatorname{rot} h) \in \mathbb{F}_2^{2n \times n}$, где $\operatorname{rot} h$ — циркулянтная матрица, индуцированная вектором h. В отличие от $\mathcal{C}$, предполагается, что никто не знает эффективного алгоритма декодирования для этого кода. Отметим, что его декодирование не требуется ни для шифрования, ни для расшифрования при работе схемы.

Параметры криптосистемы отражены в табл. 4 и представляют собой набор $(n, k, \Delta, w, w_r, w_e, \ell)$, где n — наименьшее примитивное число, большее $n_1 n_2$ (произведение длин кодов Рида — Соломона и Рида — Маллера соответственно), которое позволяет избежать алгебраической атаки

Таблица 4

Параметры схемы HQC

Битовая стойкость	n_1	n_2	n	w	$w_r = w_e$	Открытый ключ, Б	Секретный ключ, Б	Шифр-текст, Б
128	46	384	17,669	66	75	2,249	56	4,497
192	56	640	35,851	100	114	4,522	64	9,042
256	90	640	57,637	131	149	7,245	72	14,485

факторизацией многочлена. Лишние $\ell = n - n_1 n_2$ битов, появляющиеся в результате, отсекаются операцией truncate$(v, \text{num-bits})$. Далее k — кодовое расстояние кода $\mathcal{C}$, w — вес Хэмминга векторов x, y, а w_r, w_e — веса Хэмминга векторов r_1, r_2 и e.

Безопасность схемы шифрования с открытым ключом, представленной на рис. 13, в модели IND-CPA зависит от сложности решения задачи QCSD. Итоговая схема инкапсуляции ключа получается из схемы HQC-PKE с помощью FO-преобразования, что обеспечивает безопасность в модели IND-CCA.

Создателями приведены как официальная реализация [52], так и оптимизированная для работы за постоянное время [53].

7.2. Схема BIKE. Bit-Flipping Key Encapsulation — схема инкапсуляции ключа на основе двоичных квазициклических кодов, также представленная на конкурс NIST. Актуальная на данный момент спецификация обновлена 10.10.2024 г. [54].

В основе BIKE лежит криптосистема Нидеррайтера на квазициклических кодах [55], безопасность которой в модели IND-CPA зависит от сложности решения задач QCSD и QCCF. Открытый текст представлен вектором (e_0, e_1) веса t, а шифртекст — его синдромом. Итоговая схема инкапсуляции ключа, безопасная в модели IND-CCA, получается с помощью FO-преобразования.

BIKE.Kgen()

1 : $(h_0, h_1) \xleftarrow{\mathcal{U}} \mathcal{R}_w$
2 : $h \leftarrow h_1 h_0^{-1}$
3 : $\mu \leftarrow \pi_\ell(h)$
4 : $\sigma \xleftarrow{\$} \mathcal{M}$
5 : **return** $(pk = h,\ sk = (h_0, h_1, \mu, \sigma))$

BIKE.Encaps(h)

1 : $m \xleftarrow{\mathcal{U}} \mathcal{M}$
2 : $(e_0, e_1) \leftarrow H(m, \pi_\ell(h))$
3 : $c \leftarrow (e_0 + e_1 h, m \oplus L(e_0, e_1))$
4 : $K \leftarrow K(m, c)$
5 : **return** (K, c)

BIKE.Decaps(h_0, h_1, μ, σ, c)

1 : $e' \leftarrow \mathsf{decoder}(c_0 h_0, h_0, h_1)$,
2 : $e' \in \mathcal{R}^2 \cup \{\perp\}$
3 : $m' \leftarrow c_1 \oplus L(e')$
4 : **if** $e' = H(m', \mu)$ **then**
5 : $\quad K \leftarrow K(m', c)$
6 : **else**
7 : $\quad K \leftarrow K(\sigma, c)$
8 : **return** K

Рис. 14. Схема BIKE-KEM

Таблица 5

Параметры схемы BIKE

Битовая стойкость	r	w	t	Открытый ключ, Б	Секретный ключ, Б	Шифр-текст, Б
128	12,323	142	134	1,540	312	1,572
192	24,659	206	199	3,082	450	3,114
256	40,973	274	264	5,122	612	5,154

В представленной на рис. 14 схеме $\mathcal{M} = \{0,1\}^{\ell}$ — множество сообщений, H, K, L — хэш-функции, а $\pi_{\ell}(h)$ возвращает первые ℓ битов вектора h. Для восстановления вектора e из синдрома используется BIKE-Flip-декодер. Параметры схемы отражены в табл. 5 и представляют собой набор (r, w, t), где r — размер блока, w — вес строки проверочной матрицы, t — вес вектора ошибок.

Создателями приведены две реализации криптосистемы BIKE — официальная [56] и дополнительная для различных архитектур [57].

8. Атаки по побочным каналам

8.1. Основные понятия. Криптографические примитивы — математические преобразования, которые входным данным сопоставляют выходные, возможно, параметризированные ключом. Поскольку криптографические примитивы будут выполняться на некотором устройстве в определённой среде, они будут обладать специальными физическими характеристиками, которые, как и математические характеристики, могут быть подвержены анализу с целью выявления секретных параметров, используемых в вычислениях. Так, например, устройства потребляют энергию и выполняют поставленные задачи за определённое время. Они также обладают электромагнитным полем, рассеивают тепло и создают шум.

Атаками по побочным каналам (side-channel attacks или SCA) называются методы атак на криптографические системы и другие защищённые устройства, при которых помимо уязвимостей в программном обеспечении или алгоритме злоумышленник также использует дополнительную информацию, извлекаемую из физического поведения системы.

По степени влияния атакующего на криптографическую систему различают *пассивные* и *активные* атаки. Первые проводятся путём прослушивания и анализа информации без изменения входных и выходных данных, а вторые — путём анализа информации и входных/выходных данных с изменением данных. К пассивным атакам относятся те, которые не оказывают заметного влияния на работу системы: атакующий получает некоторую информацию о работе системы, но сама система работает

в обычном режиме. С другой стороны, при активной атаке противник оказывает некоторое влияние на поведение системы, например, изменяя входные/выходные данные. Отметим, что различие между активными и пассивными атаками больше связано с влиянием на вычислительные процессы системы, чем с физическим влиянием на устройство.

По характеру воздействия атаки делятся на *инвазивные* — атаки, при которых криптосистема вскрывается и осуществляется прямое воздействие на её компоненты, *полуинвазивные* — атаки, при которых оказывают влияние на элементы криптосистемы, но без непосредственного контакта с ней, а также *неинвазивные* — атаки, не требующие прямого воздействия на саму криптосистему. К полуинвазивным атакам можно отнести, например, использование лазерного луча. Неинвазивные атаки направлены на анализ внешней информации о работе криптосистемы, например замер энергопотребления или времени выполнения алгоритма.

8.2. Измерительная установка и модели утечек. Для практической реализации атаки по побочным каналам первостепенное значение имеет измерительная установка. Её назначение — преобразование физических характеристик наблюдаемого устройства в анализируемые цифровые данные. Далее приведём ряд элементов, из которых обычно состоят такие установки [58]:

- целевое криптографическое устройство, например смарт-карта, программируемая пользователем вентильная матрица или интегральная схема, на которой запущен какой-либо криптографический примитив, например блочный шифр;
- внешний источник питания, тактовый генератор и любые дополнительные схемы, необходимые для правильной работы устройства;
- датчик утечки; так, например, потребление энергии можно контролировать, вставив небольшой резистор в цепь питания целевого устройства; электромагнитное излучение можно улавливать с помощью простых самодельных катушек;
- устройство сбора данных, например цифровой осциллограф, подключённый к компьютеру для статистического анализа набора измерений энергопотребления или электромагнитного излучения. Такие наборы измерений называются *трассами*.

Качество измерительной установки в основном определяется объёмом шума в её трассах. Шум является центральной проблемой для атак по побочным каналам и может влиять на их эффективность. Обычно рассматриваются следующие типы шумов: физический шум, который создаётся транзисторами и их окружением; шум измерения, вызванный процессом измерения и инструментами, которые для него используются; алгоритмический шум, создаваемый некоторыми вычислениями в реализации.

Модель утечки определяет наблюдаемую информацию, которая просачивается через побочные каналы. Моделирование происходит с помощью функции утечки, которая принимает состояние устройства в качестве входных данных и возвращает значение (или набор значений), представляющее информацию, наблюдаемую из побочных каналов. Хорошие модели утечки оказывают сильное влияние на эффективность атаки по побочным каналам. Они используются как для симуляции атак, так и для повышения их эффективности. Примерами самых простых моделей утечки являются *модель расстояния Хэмминга* и *модель веса Хэмминга*. Модель веса Хэмминга предполагает, что когда значение x_0 вычисляется в устройстве, фактические утечки по побочным каналам коррелируют с весом Хэмминга этого значения $\mathrm{wt}(x_0)$. Модель расстояния Хэмминга предполагает, что когда значение x_0 переключается на значение x_1, фактические утечки по побочным каналам коррелируют с расстоянием Хэмминга этих значений, а именно $d(x_0, x_1) = \mathrm{wt}(x_0 \oplus x_1)$. Также модель утечки, используемая для атаки, возможно, не идеально соответствует реальным наблюдениям. В таком случае говорят о шуме соответствия модели. На практике используются более продвинутые модели утечек, о которых можно прочитать подробнее в работах [59–61].

8.3. Простые и разностные атаки. По способу анализа полученных данных различают *простые* атаки, при которых исследуется прямая зависимость между информацией, полученной по побочному каналу, и операциями, выполняемыми устройством, и *разностные* атаки, при которых проводится большое количество измерений с использованием статистических методов для исследования взаимосвязи между входными данными и информацией, полученной по побочному каналу [62].

Напомним, что наборы измерений энергопотребления или электромагнитного излучения называются *трассами*. При выполнении простой атаки с помощью анализа трассы можно определить операции, выполненные устройством, или, например, количество выполненных раундов. Полученная таким образом информация, возможно, не выявляет секретной информации сама по себе. Однако такой анализ трасс утечки может быть предварительным шагом в проведении более мощной атаки. Например, таким образом можно определять части трасс, которые представляют интерес для злоумышленника.

При проведении разностной атаки злоумышленник выполняет ряд следующих шагов. В начале злоумышленник определяет алгоритм и целевую платформу (например смарт-карту), с которой он намерен извлечь секретную информацию. Злоумышленник определяет тип утечки, который он хочет использовать. Также подготавливается измерительная установка. Атаки по побочным каналам обычно основаны на стратегии

«разделяй и властвуй», в которой разные части секрета восстанавливаются по отдельности. Так, например, злоумышленник может выбрать, какая часть секретного ключа является целью его атаки. Если позволено, то злоумышленник случайно выбирает входные данные, которые подаются на целевое устройство. Если нет, то предполагается, что злоумышленник может отслеживать открытые тексты. Для ряда известных входных открытых текстов и возможных значений секрета, например всех возможных заполнений части секретного ключа, злоумышленник предсказывает получаемые в целевом устройстве значения, зависящие от секретного ключа, которые будут вычислены во время выполнения алгоритма. Затем злоумышленник моделирует утечки целевого устройства для тех же наборов возможных заполнений части секретного ключа. Используя измерительную установку, злоумышленник измеряет утечку целевого устройства. Наконец, злоумышленник применяет статистические методы для сравнения предсказанных утечек с проведёнными измерениями. Если атака успешна, то ожидается, что модель, соответствующая правильному ключевому кандидату, даст наилучший результат сравнения. Подробное описание данных атак приведено, например, в [59].

8.4. Профилированные атаки. В отличие от разностных атак, при которых с помощью модели утечки злоумышленник вычисляет гипотетическую потребляемую мощность для предсказанных значений переменных, *профилированные атаки* предполагают, что злоумышленник обладает копией целевого устройства, которую он может использовать для более точной характеризации физических утечек. Наибольшее применение получили *шаблонные атаки*, которые были представлены в [63]. Далее опишем их идею.

Предполагается, что утечки могут быть смоделированы как многомерные гауссовские распределения, называемые шаблонами, которые задаются вектором математических ожиданий и ковариационной матрицей. Эти шаблоны строятся для каждого класса секретной информации на этапе профилирования и используются позже на этапе сопоставления.

Этап профилирования состоит в вычислении параметров многомерных гауссовых распределений для каждого класса секретной информации. Гауссово распределение определяется математическим ожиданием и дисперсией. Для вычисления математического ожидания и дисперсии злоумышленник использует копию целевого устройства, над которым он имеет полный контроль и может отслеживать физические свойства. Затем злоумышленник запускает целевую криптографическую операцию большое число раз с различными значениями секретного параметра, при этом используя случайные значения для других входных данных, классифицируя полученные трассы потребляемой мощности и формируя шаблоны, каждому из которых соответствует трасса энергопотребления.

После того как все шаблоны будут созданы, трасса энергопотребления может быть связана с заданным шаблоном с помощью функции плотности вероятности.

Затем происходит сопоставление зафиксированной утечки и созданных ранее шаблонов.

8.5. Основные виды утечек. Атаки по побочным каналам тесно связаны с существованием физически наблюдаемых явлений, которые вызваны выполнением вычислительных работ электронных устройств. Так, например, микропроцессоры потребляют энергию, требуют некоторого времени вычисления, а также излучают электромагнитные поля, тепло и издают шум.

1. Атаки по времени основаны на анализе времени, которое необходимо для выполнения определённых операций в криптографических системах, с целью извлечения секретной информации. Разные операции могут занимать различное время в зависимости от входных данных, что создаёт уязвимость. Данный вид атак пассивный и неинвазивный и направлен на определение секрета с помощью анализа высокоточных замеров времени выполнения алгоритма при различных входных данных. Например, в алгоритмах шифрования, если выполнение операции отличается по времени в зависимости от значения битов ключа, злоумышленник может попытаться подобрать ключ, анализируя, сколько времени занимает корректная обработка данных.

2. Атаки по потребляемой мощности основаны на том, что во время выполнения криптографических операций потребление энергии может варьироваться в зависимости от выполняемых вычислений и обрабатываемых данных. Так же, как и в случае атаки по времени, выполняется высокоточный замер мощности, потребляемой устройством, после чего проводится анализ полученных данных с целью определения выполняемых в устройстве операций. Относятся к пассивным и неинвазивным атакам.

3. Атаки по электромагнитному излучению представляют собой метод получения секретной информации с использованием анализа электромагнитных сигналов, излучаемых устройством во время его работы. Эти атаки основаны на том факте, что многие электронные устройства испускают электромагнитные волны, которые могут быть зафиксированы и проанализированы злоумышленником. Данный вид атак пассивный и неинвазивный.

4. Акустические атаки представляют собой тип атак, при которых злоумышленник использует звук, генерируемый устройством во время работы, в качестве канала для извлечения секретной информации. Основной

сложностью для проведення данного типа атак является шум. Относятся к пассивным и неинвазивным атакам.

8.6. Методы противодействия атакам по побочным каналам. Приведём общие идеи для защиты криптографических устройств от атак по побочным каналам [59].

Для повышения устойчивости устройства к физическим атакам используют щиты, конформные клеи [64], физически неклонируемые функции [65] и съёмные источники питания [66]. К основным методам противодействия также можно отнести создание постоянных утечек или зависимости утечки от некоторой случайной величины [67], внедрение задержек, а также рандомизация времени [68]. Также для защиты блочных шифров от атак по побочным каналам применяют *маскирование* порядка d — разделение каждой секретной переменной, которая встречается в вычислениях, на $d + 1$ частей [69–71]. Наиболее распространённым является булевое маскирование: представление x в виде $x = x_0 \oplus x_1 \oplus \cdots \oplus x_d$ [69]. При применении маскирования для защиты реализации блочного шифра необходимо разработать схему для работы с масками и замаскированными данными, которая должна гарантировать, что части секрета позволят восстановить ожидаемый шифртекст. Работа [72] посвящена динамической и дифференциальной логике выполнения микросхем для уменьшения зависимости потребления энергии от данных. Для уменьшения количества информации в побочных каналах возможно добавление шума.

8.7. Краткая сводка наиболее значимых работ по теме атак на известные кодовые криптосистемы. Первая кодовая криптосистема с открытым ключом была предложена Мак-Элисом [3] с использованием двоичного кода Гоппы. Наиболее широко используемым алгоритмом декодирования для кодов Гоппы являлся алгоритм Паттерсона [73]. Первая атака по времени против реализации алгоритма Паттерсона на ПК, которая позволяла раскрыть зашифрованное сообщение, была описана в [74]. Атака затем была улучшена в [75, 76] и протестирована на платформах FPGA. Дальнейший анализ алгоритма декодирования Паттерсона привёл к более серьёзным атакам, которые направлены на восстановление секретного ключа [77, 78].

Первая атака по потребляемой мощности на криптосистему Classic McEliece была предложена в [79]. Эта атака могла полностью восстановить секретный ключ. Позже в [80] была предложена ещё одна атака по потребляемой мощности, которая направлена на восстановление зашифрованного сообщения, но не на восстановление секретного ключа. Эта атака была также успешно протестирована против реализации FPGA [81].

Альтернативой кодам Гоппы являются MDPC-коды, которые позволяют использовать открытый ключ меньшего размера [55]. Низкоресурсная реализация криптосистемы Мак-Элиса ME с кодами MDPC была предложена в [82]. Эта реализация была подвержена простым атакам по потребляемой мощности и времени, и в связи с этим в [83] была предложена улучшенная реализация. Реализация на платформах FPGA была предложена в [84]. В работе [85] на эту реализацию была проведена разностная атака по потребляемой мощности.

В [86, 87] представлены атаки по побочным каналам, направленные на систему HQC, но, как отмечается в работе, посвящённой результатам третьего раунда конкурса NIST [22], они не применимы к текущим реализациям системы. В [88, 89] представлены атаки по времени на систему HQC, которые также не применимы к текущей её версии, поскольку они были направлены на класс кодов, которые больше не используются в её построении.

В [90, 91] предложены атаки на алгоритм декапсуляции системы Classic McEliece с восстановлением открытого текста и секретного ключа соответственно. В [92] рассмотрена атака на алгоритм декапсуляции системы Classic McEliece, которая нацелена на шаг вычисления полинома — локатора ошибок с помощью алгоритма Берлекэмпа — Мэсси. Авторы [93] с помощью утечки по потребляемой мощности определяют столбцы проверочной матрицы, которые возможно удалить и таким образом уменьшить длину кода, что влечёт снижение сложности решения задачи синдромного декодирования. В [94] авторы предложили атаку на восстановление ключа системы Classic McEliece, используя утечку по потребляемой мощности во время приведения проверочной матрицы к систематическому виду с помощью метода Гаусса в процессе генерации открытого ключа. В работе [95] авторы представили шаблонную атаку на синдромное декодирование, которую они применили к программной реализации Classic McEliece. В [96] представлена шаблонная атака на алгоритм декапсуляции системы Classic McEliece с восстановлением секретного ключа, а в [97] — атака на механизмы инкапсуляции ключей, основанные на FO-преобразовании и его вариантах, которая использует утечку по побочным каналам во время вычисления псевдослучайной функции при повторном шифровании в алгоритме декапсуляции KEM. Подробное описание этих атак будет приведено во второй части работы.

9. Обзор трудов ведущих конференций

В данном разделе рассмотрены работы, связанные с атаками на кодовые криптосистемы по сторонним каналам, представленные на конференциях PQCrypto, начиная с первой, и CHES с 2000 г. В трудах конференций FSE, IACR PKC, начиная с 2000 г., работ указанной тематики

не обнаружено. Несколько работ конференций ASIACRYPT и CBCrypto не вошли в данный обзор. Отметим, что в ч. 2 данного обзора детально разобраны наиболее значимые работы по теме исследования.

9.1. Международная конференция по постквантовой криптографии. Приведём краткое описание работ по атакам на кодовые криптосистемы, представленных на конференциях PQCrypto.

1. Быстрая атака на криптосистему Мак-Элиса (2008 г.). Авторы статьи [98] отмечают, что наиболее быстрая атака на оригинальную систему Мак-Элиса (из известных на 2008 г.) основана на декодировании по информационным совокупностям. Такая атака реализована в работе Канто и Шабо [99] 1998 г. и подробнее анализировалась в [100].

В [98] авторы возвращаются к исходной атаке Штерна [14] 1988 г., которая предшествовала атаке Канто и Шабо. Авторы модернизируют её и показывают, что их атака самая быстрая из известных. Они отмечают, что для первоначально предложенных параметров криптосистемы Мак-Элиса атаку можно провести на компьютерном кластере средней мощности (1400 дней на одном процессоре Core 2 Quad CPU 2,4 ГГц или 7 дней на кластере с 200 вычислительными модулями). Ранее Канто и Сандрие также указывали на то, что система Мак-Элиса не соответствует современным стандартам безопасности, но реальная атака проведена впервые. Также в статье предлагаются новые параметры для криптосистем Мак-Элиса и Нидеррайтера, которые позволяют повысить их стойкость, в том числе к предложенной авторами атаке.

2. Атаки по побочным каналам на криптосистему Мак-Элиса (2008 г.). В статье [74], по утверждению авторов, предпринята первая попытка применить подобные атаки к криптосистеме Мак-Элиса. Авторы отмечают, что простая реализация криптосистемы Мак-Элиса может иметь слабости относительно нескольких типов атак по побочным каналам. В частности, они рассматривают атаку по времени, которая была успешно применена к программной реализации криптосистемы Мак-Элиса. Предложены некоторые усовершенствования в реализации криптосистемы, чтобы противостоять атакам по энергопотреблению и памяти.

Более детально: атаку по времени авторы предпринимают по отношению к степени полинома — локатора ошибок, который используется на шаге исправления ошибки при декодировании. Проведены теоретические исследования и сама практическая атака. Авторами предложены усовершенствования реализации криптосистемы Мак-Элиса против атаки по энергопотреблению на построение проверочной матрицы кода на этапе генерации ключа, а также относительно атаки по времени доступа к памяти в отношении перестановки кодовых слов во время расшифрования.

3. Практические атаки по мощности на реализации криптосистемы Мак-Элиса (2010 г.). Напомним, что стойкость криптосистемы Мак-Элиса основана на том, что задача о декодировании произвольного линейного двоичного кода NP-трудна. Авторы [79] обращают внимание на то, что интерес к реализации постквантовых криптографических алгоритмов, таких как криптосистема Мак-Элиса, на микропроцессорных платформах существенно возрос из-за увеличения объёма памяти устройств. В связи с этим необходимо изучать их уязвимость и устойчивость к физическим атакам, например к современным атакам по мощности. В работе [79] авторы исследуют две атаки по мощности на различные реализации криптосистемы Мак-Элиса на 8-битном микропроцессоре AVR, при этом они отмечают, что подобные атаки рассмотриваются на практике впервые.

4. Атака по времени на секретную перестановку в криптосистеме Мак-Элиса (2010 г.). В [77] представлена новая атака по времени на криптосистему Мак-Элиса. Автор предлагает использовать уязвимости в алгоритме Паттерсона, которые позволяют злоумышленнику собирать информацию о секретной перестановке по побочному каналу. Как утверждает автор, полученная информация может быть использована для существенного снижения сложности атаки, основанной на полном переборе секретного ключа. Автор также описывает некоторые контрмеры к своей атаке.

5. Декодирование «одного из многих» (2011 г.). Как отмечается в статье [45], одной из самых распространённых атак на кодовые криптосистемы в целом является атака, направленная на декодирование случайного линейного кода, поэтому для выбора секретных параметров кодовой системы необходимо тщательно анализировать и измерять сложность лучших методов декодирования для кодов, которые предполагается в ней использовать. Автор рассматривает ситуацию, в которой злоумышленник имеет доступ к многим шифртекстам, и целью атаки является дешифрование какого-либо одного из них.

6. Атаки по времени на инвертирование синдрома в кодовых криптосистемах (2013 г.). В [78] представлена первая практическая атака по времени на кодовые криптосистемы. Атака основана на уязвимостях, обнаруживающихся при расшифровании, а именно — при инвертировании синдрома с помощью расширенного алгоритма Евклида. При этом для успешной атаки автор комбинирует три типа уязвимостей: восстановление нулевого элемента, уточнение первой уязвимости с получением линейных уравнений, а затем и кубических уравнений. Все подходы вместе позволяют получить дополнительную информацию о носителе — части ключа кодовой криптосистемы.

7. Устойчивые к атакам по побочным каналам реализации криптосистемы QC-MDPC Мак-Элиса на устройствах с ограниченными возможностями (2014 г.). Авторы [83] делают отсылку к работе [55], в которой предложено использование квазициклических кодов (QC-MDPC) для криптосистемы Мак-Элиса. Данные коды могут обеспечивать как относительно малый размер ключа, так и высокую производительность на скоростных вычислительных ресурсах. Однако, как отмечают авторы, для широко распространённых микроконтроллеров ранее были представлены только медленные реализации. Они представляют реализацию криптосистемы QC-MDPC Мак-Элиса, обеспечивающую стойкость на уровне 80 битов (порядка 2^{80} операций) на недорогих микроконтроллерах ARM Cortex-M4 с приемлемой производительностью 42 мс при зашифровании и 251–558 мс при расшифровании. Помимо практических вопросов, таких как генерация случайного вектора ошибок, авторы рассматривают атаки по побочным каналам на простую реализацию предложенной схемы и предлагают контрмеры для её защиты от атак по времени и мощности.

8. QC-MDPC Мак-Элиса: атака по времени и CCA2 KEM (2018 г.). В [101] проводится глубокий разбор первопричин GJS-атаки на криптосистему QC-MDPC Мак-Элиса 2016 г. [48]. Авторы предлагают контрмеры для защиты и отмечают, что вес синдрома является фундаментальной величиной, из-за которой происходит утечка секретной информации. Если по побочному каналу удаётся контролировать вес синдрома, то можно провести атаку с восстановлением ключа.

9. Декодеры QC-MDPC с несколькими «оттенками серого» (2020 г.). Схемы KEM на основе квазициклических кодов задействуют декодеры, имеющие небольшую или пренебрежимо малую частоту отказов при декодировании. Эти декодеры должны быть эффективными и реализуемыми в режиме постоянного времени. Одним из примеров такого подхода является BIKE, кандидат второго раунда конкурса NIST. Авторы [102] продолжают свои исследования по теме Black-Gray декодеров и улучшают предыдущие показатели декапсуляции BIKE.

10. Атака по мощности на реализацию криптосистемы HQC, основанную на комбинации кодов Рида — Маллера и Рида — Соломона (2022 г.). В [103] рассматривается схема HQC, являющаяся кандидатом четвёртого раунда конкурса NIST. Авторы отмечают, что начиная с третьей версии, в алгоритме используется новая комбинация кодов, а именно кода Рида — Маллера и кода Рида — Соломона, которая требует модификации ранее уже опубликованных атак. Авторы утверждают, что атака по мощности, предпринятая Uneo и соавторами на CHES 2021, на практике не работает, поскольку упущен тот факт, что реализованный декодер Рида — Маллера не имеет фиксированной границы декодирования. В своей работе [103]

они предлагают определённую модификацию атаки, что делает её успешной для рассматриваемой версии алгоритма.

11. Новая атака восстановления ключа по сторонним каналам на HQC на основе выбранного шифртекста (2022 г.). Авторы [104] вновь отмечают, что определённые этапы декодирования кодовых криптосистем уязвимы для атак по сторонним каналам, и HQC не является исключением. Авторы предлагают новую атаку по сторонним каналам для восстановления ключа HQC с использованием выбранного шифртекста. Атака опирается на преимущества повторного использования статического секретного ключа на микроконтроллере с физическим доступом. Цель авторов, как они её формулируют, состоит в том, чтобы получить статический секретный ключ, ориентируясь на этап декодирования кода Рида — Маллера при декапсуляции и, более точно, на преобразование Адамара. Информация, полученная через сторонние каналы, используется для построения оракула, который различает несколько схем декодирования кодов Рида — Маллера. Авторы показывают, как сделать запрос к оракулу таким образом, чтобы ответы предоставляли полную информацию о статическом секретном ключе. Авторы провели эксперименты и утверждают, что для извлечения всего статического секретного ключа, используемого для декапсуляции, достаточно менее 20 000 трасс в рамках электромагнитной атаки, при этом они предлагают способы защиты от неё.

9.2. Международная конференция по криптографическому оборудованию и встроенным системам. В этом пункте даётся краткое описание работ по атакам на кодовые криптосистемы, представленных на конференциях CHES.

1. QcBits — кодовая криптосистема с постоянным временем работы (2016 г.). В статье [105] представлена схема QcBits — реализация алгоритма шифрования с открытым ключом на основе схемы Нидеррайтера с квазициклическими кодами, выполняющая соответствующие операции за постоянное время для противостояния атакам по времени.

2. Атака по побочным каналам на криптосистему QcBits (2017 г.). В работе [106] демонстрируется, что QcBits, несмотря на стойкость к атакам по времени, уязвима для разностной атаки по энергопотреблению на вычисление синдрома в алгоритме декодирования. Представленная атака позволила авторам составить систему двоичных линейных уравнений с ошибками. После решения системы был полностью восстановлен ключ. В качестве меры противодействия атаке авторы предложили маскирование кодового слова путём сложения его с другим случайным кодовым словом перед процедурой вычисления синдрома.

3. Расширение ошибки в кодовых криптосистемах (2018 г.). Кодовые криптосистемы с открытым ключом имеют вероятность ошибки декодирования, что позволяет, например, проводить GJS-атаку. В статье [49] авторы значительно усиливают эту реакционную атаку, показывая, что после нахождения всего одного паттерна вектора ошибок, который ведёт к отказу декодирования, время, необходимое для нахождения другого сообщения, которое также приведёт к отказу декодирования, становится очень мало́. Этот результат часто используется в совокупности с атаками по сторонним каналам, позволяющими различать успешное и ошибочное декодирование, так как такая утечка информации по сторонним каналам позволяет значительно ускорить поиск первого трудно расшифровываемого сообщения.

4. Совершенствование атак по побочным каналам на кодовые криптосистемы (2019 г.). В статье [107] авторы улучшают атаку на QcBits, предложенную в [106], и демонстрируют, что с помощью утечки по энергопотреблению возможно восстановить ключ без необходимости решать систему зашумлённых двоичных линейных уравнений. В дополнение делается вывод, что криптосистема BIKE по состоянию на время проведения второго раунда конкурса NIST может быть также уязвима к предложенной атаке.

5. Восстановление секретного ключа атакой по времени на криптосистемы HQC и BIKE (2021 г.). В [108] исследована возможность атаки по времени на схемы BIKE и HQC, актуальные на момент публикации. Несмотря на попытку создать реализацию с постоянным временем работы, в системе HQC для генерации случайного вектора фиксированного веса в повторном шифровании при применении преобразования Фуджисаки — Окамото использована процедура выборки с отклонением, время выполнения которой зависит от начального значения θ, в свою очередь зависящего от сообщения именно при инкапсуляции и декапсуляции ключа. В схеме BIKE при декапсуляции ключа также допущены утечки по времени при генерации кодового слова фиксированного веса, что позволяет различать успешность декодирования. Эта информация впоследствии использована в GJS-атаке, позволяющей восстановить секретный ключ. Авторы предполагают, что для выполнения данной атаки злоумышленник имеет возможность взаимодействовать с системой: выполнять зашифрование с инкапсулированным ключом, подавать полученные шифртексты для декапсуляции и наблюдать за выводом процедуры декапсуляции, а также получать информацию о времени выполнения декапсуляции. Впоследствии авторы схемы HQC заменили алгоритм генерации случайного вектора заданного веса алгоритмом 5 из статьи [109] 2021 г.

6. Атака по времени доступа к памяти на криптосистему HQC (2023 г.). В [110] авторы демонстрируют атаку с выбранным шифртекстом по времени доступа к памяти на официальную реализацию системы HQC. Эта работа во многом вдохновлена атакой по времени, описанной в [108].

На стадии профилирования атакующий использует технику flush-and-reload, которая полагается на использование программами общего кэша: в первой фазе атакующий удаляет из кэша участок памяти, затем дожидается исполнения целевой программы. Наконец, атакующий снова запрашивает тот участок памяти, который удалил на первом этапе. Быстрое получение доступа к памяти означает, что при исполнении этот участок был использован и заново внесён в кэш. На основе полученной таким образом информации атакующий строит оракул PC для проверки того, что определённый шифртекст действительно расшифровывается в определённое сообщение. Уязвимость, позволявшая реализовать данную атаку, заключалась в том, что при выполнении процедуры случайной генерации векторов e и r_1 фиксированного веса в кэш загружались только ненулевые координаты векторов. Заметим, что в актуальной реализации HQC время выполнения постоянно, а указанная процедура получает доступ ко всему вектору.

7. Атака по электромагнитному излучению на криптосистему HQC (2023 г.). В работе [111] авторы предложили атаку для восстановления общего ключа на основе алгоритма распространения доверия на несколько шагов алгоритма декапсуляции схемы HQC-KEM: алгоритм декодирования кодов Рида — Соломона и алгоритм кодирования Рида — Соломона, использующийся для повторного шифрования при применении преобразования Фуджисаки — Окамото. Предполагается, что злоумышленник имеет полный контроль над точной копией устройства и может проводить измерения электромагнитного излучения при выполнении операции умножения в поле Галуа. Авторы показывают, что маскирование и перемешивание являются недостаточно эффективными стратегиями противодействия подобной атаке, и оценивают стратегию полного перемешивания, которая могла бы помешать провести данную атаку. Однако в связи со сложностью применения подобной контрмеры именно для защиты алгоритма кодирования Рида — Соломона авторы предлагают заменить его.

9.3. Азиатская конференция по криптографии ASIACRYPT. Авторы работы [48] 2016 г. предложили новую атаку на схемы шифрования с открытым ключом, использующие квазициклические коды, и назвали её реакционной. Впоследствии эту атаку стали называть атакой Гуо — Йохансона — Станковского или GJS-атакой — по именам авторов. В ходе атаки злоумышленник пытается восстановить секретный ключ,

исходя из статистики ошибок декодирования. На первом шаге он посылает специальные сообщения получателю и наблюдает за реакцией последнего: удалось ли декодировать сообщение или произошла ошибка декодирования. Анализ распределения ошибок декодирования позволяет злоумышленнику построить так называемый спектр расстояний — набор расстояний между парами единиц в секретном ключе. На втором шаге атакующий пытается восстановить секретный ключ на основе спектра расстояний. Авторы также предложили модификацию атаки для схем инкапсуляции ключа и описали контрмеры к данной атаке.

Финансирование работы

Исследование выполнено в рамках государственного задания Института математики им. С. Л. Соболева (проект № FWNF–2022–0019), а также при финансовой поддержке Национального технологического центра цифровой криптографии. Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Shor P. W.** Algorithms for quantum computation: Discrete logarithms and factoring // Proc. 35th Annu. Symp. Foundations of Computer Science (Santa Fe, USA, Nov. 20–22, 1994). Los Alamitos, CA: IEEE Comput. Soc., 1994. P. 124–134. DOI: 10.1109/SFCS.1994.365700.
2. **Niederreiter H.** Knapsack-type cryptosystems and algebraic coding theory // Prob. Control Inf. Theory. 1986. V. 15, No. 2. P. 157–166.
3. **McEliece R. J.** A public-key cryptosystem based on algebraic coding theory // DSN Progress Rep. 1978. V. 42–44. P. 114–116.
4. **Bernstein D. J., Chou T., Cid C.** [et al.]. Classic McEliece. Specification. Chicago: Univ. Ill. Chic., 2022. 16 p. URL: classic.mceliece.org/spec.html (accessed: 6.03.2026).
5. **Высоцкая В. В., Чижов И. В.** Постквантовая схема инкапсуляции ключа «Кодиеум» // Докл. XXVI Междунар. науч.-практ. конф. «РусКрипто» (Москва, Россия, 19–22 марта 2024 г.). М.: РусКрипто, 2024. 16 p. URL: ruscrypto.ru/resource/archive/rc2024/files/05_vysotskaya_chizhov.pdf (дата обращения: 6.03.2026).
6. **Высоцкая В. В., Чижов И. В.** Схема постквантовой электронной подписи на основе протокола идентификации Штерна // Докл. XXIII Междунар. науч.-практ. конф. «РусКрипто» (Москва, Россия, 23–26 марта 2021 г.). М.: РусКрипто, 2021. 27 p. URL: ruscrypto.ru/resource/archive/rc2021/files/02_vysotskaya_chizhov.pdf (дата обращения: 6.03.2026).

7. **Berlekamp E., McEliece R., Van Tilborg H.** On the inherent intractability of certain coding problems (corresp.) // IEEE Trans. Inf. Theory. 1978. V. 24, No. 3. P. 384–386. DOI: 10.1109/TIT.1978.1055873.
8. **Гоппа В. Д.** Рациональное представление кодов и (L, g)-коды // Пробл. передачи информации. 1971. Т. 7, № 3. С. 41–49.
9. **Hofheinz D., Hövelmanns K., Kiltz E.** A modular analysis of the Fujisaki–Okamoto transformation // Theory of cryptography. Proc. 15th Int. Conf. (Baltimore, MD, USA, Nov. 12–15, 2017). Pt. I. Cham: Springer, 2017. P. 341–371. (Lect. Notes Comput. Sci.; V. 10677). DOI: 10.1007/978-3-319-70500-2_12.
10. **Bindel N., Hamburg M., Hövelmanns K.** [et al.]. Tighter proofs of CCA security in the quantum random oracle model // Theory of cryptography. Proc. 17th Int. Conf. (Nuremberg, Germany, Dec. 1–5, 2019). Pt. II. Cham: Springer, 2019. P. 61–90. (Lect. Notes Comput. Sci.; V. 11892). DOI: 10.1007/978-3-030-36033-7_3.
11. **Fiat A., Shamir A.** How to prove yourself: Practical solutions to identification and signature problems // Advances in cryptology — CRYPTO'86. Proc. Conf. Theory and Applications of Cryptographic Techniques (Santa Barbara, USA, Aug. 11–15, 1986). Heidelberg: Springer, 1987. P. 186–194. (Lect. Notes Comput. Sci.; V. 263). DOI: 10.1007/3-540-47721-7_12.
12. **Gao S., Mateer T.** Additive fast Fourier transforms over finite fields // IEEE Trans. Inf. Theory. 2010. V. 56, No. 12. P. 6265–6272.
13. **Sendrier N.** Finding the permutation between equivalent linear codes: The support splitting algorithm // IEEE Trans. Inf. Theory. 2000. V. 46, No. 4. P. 1193–1203. DOI: 10.1109/18.850662.
14. **Stern J.** A method for finding codewords of small weight // Coding theory and applications. Proc. 3rd Int. Colloq. (Toulon, France, Nov. 2–4, 1988). Heidelberg: Springer, 1988. P. 106–113. (Lect. Notes Comput. Sci.; V. 388). DOI: 10.1007/BFb0019850.
15. **Сидельников В. М., Шестаков С. О.** О системе шифрования, построенной на основе обобщённых кодов Рида — Соломона // Дискрет. математика. 1992. Т. 4, № 3. С. 57–63.
16. **Davydov V. V., Beliaev V. V., Kustov E. F.** [et al.]. Modern variations of McEliece and Niederreiter cryptosystems // J. Sci. Tech. Inf. Technol. Mech. Opt. 2022. V. 22, No. 2. P. 324–331. DOI: 10.17586/2226-1494-2022-22-2-324-331.
17. **Сидельников В. М.** Открытое шифрование на основе двоичных кодов Рида — Маллера // Дискрет. математика. 1994. Т. 6, № 2. С. 3–20.
18. **Minder L., Shokrollahi A.** Cryptanalysis of the Sidelnikov cryptosystem // Advances in cryptology — EUROCRYPT 2007. Proc. 26th Annu. Int. Conf. Theory and Applications of Cryptographic Techniques (Barcelona, Spain, May 20–24, 2007). Heidelberg: Springer, 2007. P. 347–360. (Lect. Notes Comput. Sci.; V. 4515). DOI: 10.1007/978-3-540-72540-4_20.
19. **Overbeck R., Sendrier N.** Code-based cryptography // Post-quantum cryptography. Heidelberg: Springer, 2009. P. 95–145.

20. **González de la Torre M. A., Hernández Encinas L., Sánchez García J. I.** Structural analysis of code-based algorithms of the NIST post-quantum call // Logic J. IGPL. 2024. V. 33, No. 5. Article ID jzae071. 12 p. DOI: 10.1093/jigpal/jzae071.
21. **Alagic G., Bros M., Ciadoux P.** [et al.]. Status report on the fourth round of the NIST post-quantum cryptography standardization process. Gaithersburg, MD: NIST, 2025. DOI: 10.6028/NIST.IR.8545.
22. **Alagic G., Apon D. C., Cooper D.** [et al.]. Status report on the third round of the NIST post-quantum cryptography standardization process. Gaithersburg, MD: NIST, 2022. DOI: 10.6028/NIST.IR.8413-upd1.
23. **Alagic G., Alperin-Sheriff J., Apon D. C.** [et al.]. Status report on the second round of the NIST post-quantum cryptography standardization process. Gaithersburg, MD: NIST, 2020. DOI: 10.6028/NIST.IR.8309.
24. **Albrecht M. R., Bernstein D. J., Chou T.** [et al.]. Classic McEliece // Post-quantum cryptography. Round 3 submissions. Gaithersburg, MD: NIST, 2020. URL: csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions (accessed: 6.03.2026).
25. **Bernstein D. J., Chou T., Cid C.** [et al.]. Classic McEliece // Post-quantum cryptography. Round 4 submissions. Gaithersburg, MD: NIST, 2022. URL: csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-4-submissions (accessed: 6.03.2026).
26. **Bernstein D. J., Chou T., Cid C.** [et al.]. Classic McEliece. Implementation. Chicago: Univ. Ill. Chic., 2022. URL: classic.mceliece.org/impl.html (accessed: 6.03.2026).
27. CIRCL: Cloudflare interoperable reusable cryptographic library. San Francisco: Cloudflare, 2023. URL: github.com/cloudflare/circl (accessed: 6.03.2026).
28. Implement Classic McEliece. San Francisco: Cloudflare, 2022. URL: github.com/cloudflare/circl/pull/378 (accessed: 6.03.2026).
29. Open Quantum Safe liboqs: C library for prototyping and experimenting with quantum-resistant cryptography. 2025. URL: github.com/open-quantum-safe/liboqs/tree/main/src/kem/classic_mceliece (accessed: 6.03.2026).
30. **Wiggers T., Stebila D.** Clean, portable, tested implementations of post-quantum cryptography. 2023. URL: github.com/PQClean/PQClean (accessed: 6.03.2026).
31. **Hülsing A., Ning K.-C., Schwabe P., Weber F., Zimmermann P. R.** Post-quantum WireGuard // Proc. 42nd IEEE Symp. Security and Privacy (San Francisco, USA, May24–27, 2021). Los Alamitos, CA: IEEE Comput. Soc., 2021. P. 304–321. DOI: 10.1109/SP40001.2021.00030.
32. Software co-design acceleration of Classic McEliece key encapsulation mechanism. 2021. URL: github.com/beatsnbytes/classic_mceliece (accessed: 6.03.2026).

33. Discrete math final project for 2018 — Implementation of the McEliece cryptosystem. 2018. URL: github.com/arpanrau/McEliece-Implementation (accessed: 6.03.2026).

34. **Nießen T.** Purely educational PoC design and implementation of a PQC key exchange using Classic McEliece. 2019. URL: github.com/tniessen/node-mceliece-key-exchange-poc (accessed: 6.03.2026).

35. **Bernstein D. J.** The McEliece cryptosystem // Talks 1st Post-Quantum Cryptography Summer School in Universities (Chengdu, China, July 17, 2024). 76 p. URL: cr.yp.to/talks/2024.07.17/slides-djb-20240717-mceliece-4x3.pdf (accessed: 6.03.2026).

36. ГОСТ 34.11—2018. Информационная технология. Криптографическая защита информации. Функция хэширования. Введ. 01.06.2019. М.: Стандартинформ, 2018. 25 с.

37. **Vysotskaya V. V., Chizhov I. V.** Design criteria of a new code-based KEM // J. Comput. Virol. Hacking Tech. 2024. V. 20, No. 3. P. 497–511. DOI: 10.1007/s11416-024-00527-z.

38. **Ge J., Liao H., Xue R.** Measure-rewind-extract: Tighter proofs of one-way to hiding and CCA security in the quantum random oracle model // Advances in cryptology — ASIACRYPT 2024. Proc. 30th Int. Conf. Theory and Application of Cryptology and Information Security (Kolkata, India, Dec. 9–13, 2024). Pt. IV. Singapore: Springer, 2024. P. 3–34. (Lect. Notes Comput. Sci.; V. 15487). DOI: 10.1007/978-981-96-0894-2_1.

39. **Stern J.** A new identification scheme based on syndrome decoding // Advances in cryptology — CRYPTO'93. Proc. 13th Annu. Int. Cryptology Conf. (Santa Barbara, USA, Aug. 22–26, 1993). Heidelberg: Springer, 1994. P. 13–21. (Lect. Notes Comput. Sci.; V. 773). DOI: 10.1007/3-540-48329-2_2.

40. **Vysotskaya V. V., Chizhov I. V.** The security of the code-based signature scheme based on the Stern identification protocol // Прикл. дискрет. математика. 2022. № 57. С. 67–90. DOI: 10.17223/20710410/57/5.

41. **Царегородцев К. Д.** Троичная лемма о разветвлении и её приложение к анализу стойкости одной кодовой схемы подписи // Прикл. дискрет. математика. 2023. № 59. С. 58–71. DOI: 10.17223/20710410/59/3.

42. **Высоцкая В. В., Дас Д. К.** Анализ устойчивости постквантовой электронной подписи «Шиповник» к атакам, нацеленным на хэш-функции // Докл. XXVI Междунар. науч.-практ. конф. «РусКрипто» (Москва, Россия, 19–22 марта 2024 г.). М.: РусКрипто, 2024. 36 р. URL: ruscrypto.ru/resource/archive/rc2024/files/05_vysotskaya_das.pdf (дата обращения: 6.03.2026).

43. Открытая реализация алгоритма электронной цифровой подписи «Шиповник» для ТК26. М.: QApp, 2023. URL: github.com/QAPP-tech/shipovnik_tc26 (дата обращения: 6.03.2026).

44. **Prange E.** The use of information sets in decoding cyclic codes // IRE Trans. Inf. Theory. 1962. V. 8, No. 5. P. 5–9. DOI: 10.1109/TIT.1962.1057777.

45. Sendrier N. Decoding one out of many // Post-quantum cryptography. Proc. 4th Int. Workshop (Taipei, China, Nov. 29 – Dec. 2, 2011). Heidelberg: Springer, 2011. P. 51–67. (Lect. Notes Comput. Sci.; V. 7071). DOI: 10.1007/978-3-642-25405-5_4.

46. Löndahl C., Johansson T., Shooshtari M. K., Ahmadian-Attari M., Aref M. R. Squaring attacks on McEliece public-key cryptosystems using quasi-cyclic codes of even dimension // Des. Codes Cryptogr. 2016. V. 80, No. 2. P. 359–377. DOI: 10.1007/s10623-015-0099-x.

47. Guo Q., Johansson T., Löndahl C. A new algorithm for solving ring-lpn with a reducible polynomial // IEEE Trans. Inf. Theory. 2015. V. 61, No. 11. P. 6204–6212. DOI: 10.1109/TIT.2015.2475738.

48. Guo Q., Johansson T., Stankovski P. A key recovery attack on MDPC with CCA security using decoding errors // Advances in cryptology — ASIACRYPT 2016. Proc. 22nd Int. Conf. Theory and Application of Cryptology and Information Security (Hanoi, Vietnam, Dec. 4–8, 2016). Pt. I. Heidelberg: Springer, 2016. P. 789–815. (Lect. Notes Comput. Sci.; V. 10031). DOI: 10.1007/978-3-662-53887-6_29.

49. Nilsson A., Johansson T., Wagner P. S. Error amplification in code-based cryptography // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2019. V. 2019, No. 1. P. 238–258. DOI: 10.46586/tches.v2019.i1.238-258.

50. Aguilar-Melchor C., Blazy O., Deneuville J.-C. [et al.]. Efficient encryption from random quasi-cyclic codes // IEEE Trans. Inf. Theory. 2018. V. 64, No. 5. P. 3927–3943. DOI: 10.1109/TIT.2018.2804444.

51. Gaborit P., Aguilar-Melchor C., Aragon N. [et al.]. HQC cryptosystem specification. Gaithersburg, MD: NIST, 2025. URL: pqc-hqc.org/doc/hqc_specifications_2025_08_22.pdf (accessed: 6.03.2026).

52. Gaborit P., Aguilar-Melchor C., Aragon N. [et al.]. HQC. NIST submission packages. 2025. URL: pqc-hqc.org/doc/archive_submissions.zip (accessed: 6.03.2026).

53. Gaborit P., Aguilar-Melchor C., Aragon N. [et al.]. HQC. Optimized implementation. 2024. URL: web.archive.org/web/20250712014511/pqc-hqc.org/doc/hqc-optimized-implementation_2024-10-30.zip (accessed: 6.03.2026).

54. Aragon N., Barreto P., Bettaieb S. [et al.]. BIKE cryptosystem specification. Gaithersburg, MD: NIST, 2024. URL: bikesuite.org/files/v5.2/BIKE_Spec.2024.10.10.1.pdf (accessed: 6.03.2026).

55. Misoczki R., Tillich J.-P., Sendrier N. [et al.]. MDPC-McEliece: New McEliece variants from moderate density parity-check codes // Proc. 2013 IEEE Int. Symp. Information Theory (Istanbul, Turkey, July 7–12, 2013). Piscataway: IEEE, 2013. P. 2069–2073. DOI: 10.1109/ISIT.2013.6620590.

56. Aragon N., Barreto P., Bettaieb S. [et al.]. BIKE. Reference implementation. 2024. URL: bikesuite.org/reference.html (accessed: 6.03.2026).

57. Additional implementation of BIKE. Seattle: AWS Labs, 2024. URL: github.com/awslabs/bike-kem (accessed: 6.03.2026).

58. **Mangard S., Oswald E., Popp T.** Power analysis attacks: Revealing the secrets of smart cards. New York: Springer, 2007. 338 p. DOI: 10.1007/978-0-387-38162-6.
59. **Standaert F. X.** Introduction to side-channel attacks // Secure integrated circuits and systems. New York: Springer, 2010. P. 27–42. DOI: 10.1007/978-0-387-71829-3_2.
60. **Peeters E., Standaert F. X., Quisquater J. J.** Power and electromagnetic analysis: Improved model, consequences and comparisons // Integration. 2007. V. 40, No. 1. P. 52–60. DOI: 10.1016/j.vlsi.2005.12.013.
61. **Standaert F. X., Mace F., Peeters E.** [et al.]. Updates on the security of FPGAs against power analysis attacks // Reconfigurable computing: Architectures and applications. Rev. Sel. Pap. 2nd Int. Workshop (Delft, The Netherlands, Mar. 1–3, 2006). Heidelberg: Springer, 2006. P. 335–346. (Lect. Notes Comput. Sci.; V. 3985). DOI: 10.1007/11802839_42.
62. **Жуков А. Е.** Криптоанализ по побочным каналам (side channel attacks) // Защита информации. Инсайд. 2010. № 5. С. 28–33.
63. **Chari S., Rao J. R., Rohatgi P.** Template attacks // Cryptographic hardware and embedded systems — CHES 2002. Rev. Pap. 4th Int. Workshop (Redwood Shores, CA, USA, Aug. 13–15, 2002). Heidelberg: Springer, 2003. P. 13–28. (Lect. Notes Comput. Sci.; V. 2523). DOI: 10.1007/3-540-36400-5_3.
64. **Anderson R., Kuhn M.** Tamper resistance — A cautionary note // Proc. 2nd USENIX Workshop Electronic Commerce (Oakland, CA, USA, Nov. 18–21, 1996). Pittsburgh, PA: Carnegie Mellon Univ., 1996. P. 1–11.
65. **Tuyls P., Schrijen G.-J., Škorić B.** [et al.]. Read-proof hardware from protective coatings // Cryptographic hardware and embedded systems — CHES 2006. Proc. 8th Int. Workshop (Yokohama, Japan, Oct. 10–13, 2006). Heidelberg: Springer, 2006. P. 369–383. (Lect. Notes Comput. Sci.; V. 4249). DOI: 10.1007/11894063_29.
66. **Shamir A.** Protecting smart cards from passive power analysis with detached power supplies // Cryptographic hardware and embedded systems — CHES 2000. Proc. 2nd Int. Workshop (Worcester, MA, USA, Aug. 17–18, 2000). Heidelberg: Springer, 2000. P. 71–77. (Lect. Notes Comput. Sci.; V. 1965). DOI: 10.1007/3-540-44499-8_5.
67. **Goubin L., Patarin J.** DES and differential power analysis. The "Duplication" method // Cryptographic hardware and embedded systems. Proc. 1st Int. Workshop (Worcester, MA, USA, Aug. 12–13, 1999). Heidelberg: Springer, 1999. P. 158–172. (Lect. Notes Comput. Sci.; V. 1717). DOI: 10.1007/3-540-48059-5_15.
68. **May D., Muller H. L., Smart N. P.** Random register renaming to foil DPA // Cryptographic hardware and embedded systems — CHES 2001. Proc. 3rd Int. Workshop (Paris, France, May 14–16, 2001). Heidelberg: Springer, 2001. P. 28–38. (Lect. Notes Comput. Sci.; V. 2162). DOI: 10.1007/3-540-44709-1_4.

69. Chari S., Jutla S. C., Rao R. J. [et al.]. Towards sound approaches to counteract power-analysis attacks // Advances in cryptology — CRYPTO'99. Proc. 19th Annu. Int. Cryptology Conf. (Santa Barbara, USA, Aug. 15–19, 1999). Heidelberg: Springer, 1999. P. 398–412. (Lect. Notes Comput. Sci.; V. 1666). DOI: 10.1007/3-540-48405-1_26.

70. Ueno R., Homma N., Aoki T. Toward more efficient DPA-resistant AES hardware architecture based on threshold implementation // Constructive side-channel analysis and secure design. Rev. Sel. Pap. 8th Int. Workshop (Paris, France, Apr. 13–14, 2017). Cham: Springer, 2017. P. 50–64. (Lect. Notes Comput. Sci.; V. 10348). DOI: 10.1007/978-3-319-64647-3_4.

71. Schwabe P., Stoffelen K. All the AES you need on Cortex-M3 and M4 // Selected areas in cryptography — SAC 2016. Rev. Sel. Pap. 23rd Int. Conf. (St. John's, NL, Canada, Aug. 10–12, 2016). Cham: Springer, 2016. P. 180–194. (Lect. Notes Comput. Sci.; V. 10532). DOI: 10.1007/978-3-319-69453-5_10.

72. Tiri K., Akmal M., Verbauwhede I. A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards // Proc. 28th European Solid-State Circuits Conf. (Florence, Italy, Sept. 24–26, 2002). Piscataway: IEEE, 2002. P. 403–406.

73. Patterson N. The algebraic decoding of Goppa codes // IEEE Trans. Inf. Theory. 1975. V. 21, No. 2. P. 203–207. DOI: 10.1109/TIT.1975.1055350.

74. Strenzke F., Tews E., Molter G. [et al.]. Side channels in the McEliece PKC // Post-quantum cryptography. Proc. 2nd Int. Workshop (Cincinnati, OH, USA, Oct. 17–19, 2008). Heidelberg: Springer, 2008. P. 216–229. (Lect. Notes Comput. Sci.; V. 5299). DOI: 10.1007/978-3-540-88403-3_15.

75. Avanzi R., Hoerder S., Page D. [et al.]. Side-channel attacks on the McEliece and Niederreiter public-key cryptosystems // J. Cryptogr. Eng. 2011. V. 1, No. 4. P. 271–281. DOI: 10.1007/s13389-011-0024-9.

76. Shoufan A., Strenzke F., Molter H. G. [et al.]. A timing attack against Patterson algorithm in the McEliece PKC // Information security and cryptology — ICISC 2009. Rev. Sel. Pap. 12th Int. Conf. (Seoul, Korea, Dec. 2–4, 2009). Heidelberg: Springer, 2010. P. 161–175. (Lect. Notes Comput. Sci.; V. 5984). DOI: 10.1007/978-3-642-14423-3_12.

77. Strenzke F. A timing attack against the secret permutation in the McEliece PKC // Post-quantum cryptography. Proc. 3rd Int. Workshop (Darmstadt, Germany, May 25–28, 2010). Heidelberg: Springer, 2010. P. 95–107. (Lect. Notes Comput. Sci.; V. 6061). DOI: 10.1007/978-3-642-12929-2_8.

78. Strenzke F. Timing attacks against the syndrome inversion in code-based cryptosystems // Post-quantum cryptography. Proc. 5th Int. Workshop (Limoges, France, June 4–7, 2013). Heidelberg: Springer, 2013. P. 217–230. (Lect. Notes Comput. Sci.; V. 7932). DOI: 10.1007/978-3-642-38616-9_15.

79. **Heyse S., Moradi A., Paar C.** Practical power analysis attacks on software implementations of McEliece // Post-quantum cryptography. Proc. 3rd Int. Workshop (Darmstadt, Germany, May 25–28, 2010). Heidelberg: Springer, 2010. P. 108–125. (Lect. Notes Comput. Sci.; V. 6061). DOI: 10.1007/978-3-642-12929-2_9.
80. **Molter H. G., Stöttinger M., Shoufan A.** [et al.]. A simple power analysis attack on a McEliece cryptoprocessor // J. Cryptogr. Eng. 2011. V. 1, No. 1. P. 29–36. DOI: 10.1007/s13389-011-0001-3.
81. **Shoufan A., Wink T., Molter H. G.** [et al.]. A novel processor architecture for McEliece cryptosystem and FPGA platforms // Proc. 20th IEEE Int. Conf. Application-Specific Systems, Architectures and Processors (Boston, MA, USA, July 7–9, 2009). Los Alamitos, CA: IEEE Comput. Soc., 2009. P. 98–105. DOI: 10.1109/ASAP.2009.29.
82. **Heyse S., Von Maurich I., Güneysu T.** Smaller keys for code-based cryptography: QC-MDPC McEliece implementations on embedded devices // Cryptographic hardware and embedded systems — CHES 2013. Proc. 15th Int. Workshop (Santa Barbara, USA, Aug. 20–23, 2013). Heidelberg: Springer, 2013. P. 273–292. (Lect. Notes Comput. Sci.; V. 8086). DOI: 10.1007/978-3-642-40349-1_16.
83. **Von Maurich I., Güneysu T.** Towards side-channel resistant implementations of QC-MDPC McEliece encryption on constrained devices // Post-quantum cryptography. Proc. 6th Int. Workshop (Waterloo, ON, Canada, Oct. 1–3, 2014). Cham: Springer, 2014. P. 266–282. (Lect. Notes Comput. Sci.; V. 8772). DOI: 10.1007/978-3-319-11659-4_16.
84. **Von Maurich I., Güneysu T.** Lightweight code-based cryptography: QC-MDPC McEliece encryption on reconfigurable devices // Proc. 2014 Design, Automation and Test in Europe Conf. (Dresden, Germany, Mar. 24–28, 2014). Piscataway: IEEE, 2014. P. 1–6. DOI: 10.7873/DATE.2014.051.
85. **Chen C., Eisenbarth T., Von Maurich I.** [et al.]. Differential power analysis of a McEliece cryptosystem // Applied cryptography and network security. Rev. Sel. Pap. 13th Int. Conf. (New York, USA, June 2–5, 2015). Cham: Springer, 2015. P. 538–556. (Lect. Notes Comput. Sci.; V. 9092). DOI: 10.1007/978-3-319-28166-7_26.
86. **Schamberger T., Renner J., Sigl G.** [et al.]. A power side-channel attack on the CCA2-secure HQC KEM // Smart card research and advanced applications. Rev. Sel. Pap. 19th Int. Conf. (Lübeck, Germany, Nov. 18–19, 2020). Cham: Springer, 2020. P. 119–134. (Lect. Notes Comput. Sci.; V. 12609). DOI: 10.1007/978-3-030-68487-7_8.
87. **Hlauschek C., Lahr N., Schröder R. L.** On the timing leakage of the deterministic re-encryption in HQC KEM. San Diego, 2021. 24 p. (Cryptol. ePrint Archive / Univ. California; Pap. 2021/1485/20211115:124514). URL: eprint.iacr.org/archive/2021/1485/20211115:124514 (accessed: 6.03.2026).

88. **Wafo-Tapa G., Bettaieb S., Bidoux L.** [et al.]. A practicable timing attack against HQC and its countermeasure // Adv. Math. Commun. 2022. V. 16, No. 3. P. 621–642. DOI: 10.3934/amc.2020126.
89. **Paiva T. B., Terada R.** A timing attack on the HQC encryption scheme // Selected areas in cryptography — SAC 2019. Rev. Sel. Pap. 26th Int. Conf. (Waterloo, ON, Canada, Aug. 12–16, 2019). Cham: Springer, 2019. P. 551–573. (Lect. Notes Comput. Sci.; V. 11959). DOI: 10.1007/978-3-030-38471-5_22.
90. **Lahr N., Niederhagen R., Petri R.** [et al.]. Side channel information set decoding using iterative chunking: Plaintext recovery from the "Classic McEliece" hardware reference implementation // Advances in cryptology — ASIACRYPT 2020. Proc. 26th Int. Conf. Theory and Application of Cryptology and Information Security (Daejeon, South Korea, Dec. 7–11, 2020). Pt. I. Cham: Springer, 2020. P. 881–910. (Lect. Notes Comput. Sci.; V. 12491). DOI: 10.1007/978-3-030-64837-4_29.
91. **Guo Q., Johansson A., Johansson T.** A key-recovery side-channel attack on Classic McEliece implementations // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2022. V. 2022, No. 4. P. 800–827. DOI: 10.46586/tches.v2022.i4.800-827.
92. **Pircher S., Geier J., Danner J.** [et al.]. Key-recovery fault injection attack on the Classic McEliece KEM // Code-based cryptography. Rev. Sel. Pap. 10th Int. Workshop (Trondheim, Norway, May 29–30, 2022). Cham: Springer, 2022. P. 37–61. (Lect. Notes Comput. Sci.; V. 13839). DOI: 10.1007/978-3-031-29689-5_3.
93. **Grosso V., Cayrel P.-L., Colombier B.** [et al.]. Punctured syndrome decoding problem: Efficient side-channel attacks against Classic McEliece // Constructive side-channel analysis and secure design. Proc. 14th Int. Workshop (Munich, Germany, Apr. 3–4, 2023). Cham: Springer, 2023. P. 170–192. (Lect. Notes Comput. Sci.; V. 13979). DOI: 10.1007/978-3-031-29497-6_9.
94. **Brinkmann M., Chuengsatiansup C., May A.** [et al.]. Leaky McEliece: Secret key recovery from highly erroneous side-channel information // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2025. V. 2025, No. 2. P. 94–125. DOI: 10.46586/tches.v2025.i2.94-125.
95. **Bitzer S., Delvaux J., Kirshanova E.** [et al.]. How to lose some weight: A practical template syndrome decoding attack // Des. Codes Cryptogr. 2025. V. 93, No. 7. P. 2503–2519. DOI: 10.1007/s10623-025-01603-1.
96. **Drăgoi V.-F., Colombier B., Vallet N.** [et al.]. Full key-recovery cubic-time template attack on Classic McEliece decapsulation. San Diego, 2024. 25 p. (Cryptol. ePrint Archive / Univ. California; Pap. 2024/1694). URL: eprint.iacr.org/2024/1694 (accessed: 6.03.2026).
97. **Ueno R., Xagawa K., Tanaka Y.** [et al.]. Curse of re-encryption: A generic power/EM analysis on post-quantum KEMs // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2022. V. 2022, No. 1. P. 296–322. DOI: 10.46586/tches.v2022.i1.296-322.

98. Bernstein D. J., Lange T., Peters C. Attacking and defending the McEliece cryptosystem // Post-quantum cryptography. Proc. 2nd Int. Workshop (Cincinnati, OH, USA Oct. 17–19, 2008). Heidelberg: Springer, 2008. P. 31–46. (Lect. Notes Comput. Sci.; V. 5299). DOI: 10.1007/978-3-540-88403-3_3.

99. Canteaut A., Chabaud F. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511 // IEEE Trans. Inf. Theory. 1998. V. 44, No. 1. P. 367–378. DOI: 10.1109/18.651067.

100. Canteaut A., Sendrier N. Cryptanalysis of the original McEliece cryptosystem // Advances in cryptology — ASIACRYPT'98. Proc. Int. Conf. Theory and Application of Cryptology and Information Security (Beijing, China, Oct. 18–22, 1998). Heidelberg: Springer, 1998. P. 187–199. (Lect. Notes Comput. Sci.; V. 1514). DOI: 10.1007/3-540-49649-1_16.

101. Eaton E., Lequesne M., Parent A. [et al.]. QC-MDPC: A timing attack and a CCA2 KEM // Post-quantum cryptography. Proc. 9th Int. Conf. (Fort Lauderdale, FL, USA, Apr. 9–11, 2018). Cham: Springer, 2018. P. 47–76. (Lect. Notes Comput. Sci.; V. 10786). DOI: 10.1007/978-3-319-79063-3_3.

102. Drucker N., Gueron S., Kostic D. QC-MDPC decoders with several shades of gray // Post-quantum cryptography. Proc. 11th Int. Conf. (Paris, France, Apr. 15–17, 2020). Cham: Springer, 2020. P. 35–50. (Lect. Notes Comput. Sci.; V. 12100). DOI: 10.1007/978-3-030-44223-1_3.

103. Schamberger T., Holzbaur L., Renner J. [et al.]. A power side-channel attack on the Reed–Muller Reed–Solomon version of the HQC cryptosystem // Post-quantum cryptography. Proc. 13th Int. Conf. (Eindhoven, The Netherlands, Sept. 28-30, 2022). Cham: Springer, 2022. P. 327–352. (Lect. Notes Comput. Sci.; V. 13512). DOI: 10.1007/978-3-031-17234-2_16.

104. Goy G., Loiseau A., Gaborit P. A new key recovery side-channel attack on HQC with chosen ciphertext // Post-quantum cryptography. Proc. 13th Int. Conf. (Eindhoven, The Netherlands, Sept. 28–30, 2022). Cham: Springer, 2022. P. 353–371. (Lect. Notes Comput. Sci.; V. 13512). DOI: 10.1007/978-3-031-17234-2_17.

105. Chou T. QcBits: Constant-time small-key code-based cryptography // Cryptographic hardware and embedded systems — CHES 2016. Proc. 18th Int. Conf. (Santa Barbara, USA, Aug. 17–19, 2016). Heidelberg: Springer, 2016. P. 280–300. (Lect. Notes Comput. Sci.; V. 9813). DOI: 10.1007/978-3-662-53140-2_14.

106. Rossi M., Hamburg M., Hutter M. [et al.]. A side-channel assisted cryptanalytic attack against QcBits // Cryptographic hardware and embedded systems — CHES 2017. Proc. 19th Int. Conf. (Taipei, China, Sept. 25–28, 2017). Cham: Springer, 2017. P. 3–23. (Lect. Notes Comput. Sci.; V. 10529). DOI: 10.1007/978-3-319-66787-4_1.

107. Sim B.-Y., Kwon J., Choi K. Y. [et al.]. Novel side-channel attacks on quasi-cyclic code-based cryptography // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2019. V. 2019, No. 1. P. 180–212. DOI: 10.46586/tches.v2019.i4.180-212.

108. Guo Q., Hlauschek C., Johansson T. [et al.]. Don't reject this: Key-recovery timing attacks due to rejection-sampling in HQC and BIKE // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2022. V. 2022, No. 3. P. 223–263. DOI: 10.46586/tches.v2022.i3.223-263.

109. Sendrier N. Secure sampling of constant-weight words — Application to BIKE. San Diego, 2021. 16 p. (Cryptol. ePrint Archive / Univ. California; Pap. 2021/1631). URL: eprint.iacr.org/2021/1631 (accessed: 6.03.2026).

110. Huang S., Sim Q. R., Chuengsatiansup C. [et al.]. Cache-timing attack against HQC. San Diego, 2023. 34 p. (Cryptol. ePrint Archive / Univ. California; Pap. 2023/102). URL: eprint.iacr.org/2023/102 (accessed: 6.03.2026).

111. Goy G., Maillard J., Gaborit P. [et al.]. Single trace HQC shared key recovery with SASCA // IACR Trans. Cryptogr. Hardw. Embed. Syst. 2024. V. 2024, No. 2. P. 64–87. DOI: 10.46586/tches.v2024.i2.64-87.

Бахарев Александр Олегович
Воронов Денис Максимович
Коломеец Николай Александрович
Токарева Наталья Николаевна
Хильчук Ирина Сергеевна
Шапоренко Александр Сергеевич

Статья поступила
18 июня 2025 г.
После доработки —
11 августа 2025 г.
Принята к публикации
22 сентября 2025 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
October–December 2025. Vol. 32, No. 4. P. 5–68

UDC 519.7 DOI: 10.33048/daio.2025.32.837

SIDE-CHANNEL ATTACKS ON CODE-BASED POST-QUANTUM CRYPTOGRAPHIC SYSTEMS: A SURVEY. PART 1

A. O. Bakharev[1, 2, a], *D. M. Voronov*[1, 2, b], *N. A. Kolomeec*[1, 2, c], *N. N. Tokareva*[1, 2, 3, d], *I. S. Khilchuk*[1, 2, e], and *A. S. Shaporenko*[1, 2, f]

[1] National Technology Center for Digital Cryptography,
1 Ramensky Boulevard, 119192 Moscow, Russia
[2] Novosibirsk State University,
2 Pirogov Street, 630090 Novosibirsk, Russia
[3] Sobolev Institute of Mathematics,
4 Acad. Koptyug Avenue, 630090 Novosibirsk, Russia
E-mail: [a] a.bakharev@g.nsu.ru, [b] d.voronov2@g.nsu.ru,
[c] n.kolomeets@g.nsu.ru, [d] crypto1127@mail.ru,
[e] i.khilchuk@g.nsu.ru, [f] a.shaporenko@g.nsu.ru

Abstract. This work of two parts provides a structured analytical review devoted to side-channel attacks on post-quantum code-based cryptosystems. The first part of the review presents a description of the main cryptographic primitives and algorithms used in code-based cryptosystems, as well as description of the most significant modern code-based cryptosystems: Classic McEliece, Codiaeum, Shipovnik, BIKE, and HQC. This survey is carried out within the scientific and research project «Kulminatsiya» of the National Technology Center for Digital Cryptography. Tab. 5, illustr. 14, bibliogr. 111.

Keywords: post-quantum cryptography, side-channel attack, code-based cryptographic system.

References

1. **P. W. Shor,** Algorithms for quantum computation: Discrete logarithms and factoring, in *Proc. 35th Annu. Symp. Foundations of Computer Science* (Santa Fe, USA, Nov. 20–22, 1994) (IEEE Comput. Soc., Los Alamitos, CA, 1994), pp. 124–134, DOI: 10.1109/SFCS.1994.365700.
2. **H. Niederreiter,** Knapsack-type cryptosystems and algebraic coding theory, *Prob. Control Inf. Theory* **15** (2), 157–166 (1986).

English transl.: *Journal of Applied and Industrial Mathematics* **19** (4) (2025).

3. **R. J. McEliece,** A public-key cryptosystem based on algebraic coding theory, *DSN Progress Rep.* **42–44**, 114–116 (1978).
4. **D. J. Bernstein, T. Chou, C. Cid,** [et al.], Classic McEliece. Specification (Univ. Ill. Chic., Chicago, 2022), URL: `classic.mceliece.org/spec.html` (accessed: 6.03.2026).
5. **V. V. Vysotskaya** and **I. V. Chizhov,** Post-quantum key encapsulation mechanism "Kodieum", in *Dokl. XXVI Int. Sci. Pract. Conf. RusCrypto* (Moscow, Russia, Mar. 19–22, 2024) (RusCrypto, Moscow, 2024) [Russian], URL: `ruscrypto.ru/resource/archive/rc2024/files/05_vysotskaya_chizhov.pdf` (accessed: 6.03.2026).
6. **V. V. Vysotskaya** and **I. V. Chizhov,** Post-quantum signature scheme based on the Stern identification protocol, in *Dokl. XXIII Int. Sci. Pract. Conf. RusCrypto* (Moscow, Russia, Mar. 23–26, 2021) (RusCrypto, Moscow, 2021) [Russian], URL: `ruscrypto.ru/resource/archive/rc2021/files/02_vysotskaya_chizhov.pdf` (accessed: 6.03.2026).
7. **E. Berlekamp, R. McEliece,** and **H. Van Tilborg,** On the inherent intractability of certain coding problems (corresp.), *IEEE Trans. Inf. Theory* **24** (3), 384–386 (1978), DOI: `10.1109/TIT.1978.1055873`.
8. **V. D. Goppa,** A rational representation of codes and (L, g)-codes, *Probl. Peredachi Inf.* **7** (3), 41–49 (1971) [Russian] [*Probl. Inf. Transm.* **7** (3), 223–229 (1971)].
9. **D. Hofheinz, K. Hövelmanns,** and **E. Kiltz,** A modular analysis of the Fujisaki–Okamoto transformation, in *Theory of Cryptography*, Proc. 15th Int. Conf. (Baltimore, MD, USA, Nov. 12–15, 2017), Pt. I (Springer, Cham, 2017), pp. 341–371 (Lect. Notes Comput. Sci., Vol. 10677), DOI: `10.1007/978-3-319-70500-2_12`.
10. **N. Bindel, M. Hamburg, K. Hövelmanns,** [et al.], Tighter proofs of CCA security in the quantum random oracle model, in *Theory of Cryptography*, Proc. 17th Int. Conf. (Nuremberg, Germany, Dec. 1–5, 2019), Pt. II (Springer, Cham, 2019), pp. 61–90 (Lect. Notes Comput. Sci., Vol. 11892), DOI: `10.1007/978-3-030-36033-7_3`.
11. **A. Fiat** and **A. Shamir,** How to prove yourself: Practical solutions to identification and signature problems, in *Advances in Cryptology — CRYPTO'86*, Proc. Conf. Theory and Applications of Cryptographic Techniques (Santa Barbara, USA, Aug. 11–15, 1986) (Springer, Heidelberg, 1987), pp. 186–194 (Lect. Notes Comput. Sci., Vol. 263), DOI: `10.1007/3-540-47721-7_12`.
12. **S. Gao** and **T. Mateer,** Additive fast Fourier transforms over finite fields, *IEEE Trans. Inf. Theory* **56** (12), 6265–6272 (2010).
13. **N. Sendrier,** Finding the permutation between equivalent linear codes: The support splitting algorithm, *IEEE Trans. Inf. Theory* **46** (4), 1193–1203 (2000).
14. **J. Stern,** A method for finding codewords of small weight, in *Coding Theory and Applications*, Proc. 3rd Int. Colloq. (Toulon, France, Nov. 2–4, 1988) (Springer, Heidelberg, 1988), pp. 106–113 (Lect. Notes Comput. Sci., Vol. 388), DOI: `10.1007/BFb0019850`.

15. **V. M. Sidelnikov** and **S. O. Shestakov,** On insecurity of cryptosystems based on generalized Reed–Solomon codes, *Diskretn. Mat.* **4** (3), 57–63 (1992) [Russian] [*Discrete Math. Appl.* **2** (4), 439–444 (1992), DOI: 10.1515/dma.1992.2.4.439].
16. **V. V. Davydov, V. V. Beliaev, E. F. Kustov,** [et al.], Modern variations of McEliece and Niederreiter cryptosystems, *J. Sci. Tech. Inf. Technol. Mech. Opt.* **22** (2), 324–331 (2022), DOI: 10.17586/2226-1494-2022-22-2-324-331.
17. **V. M. Sidelnikov,** A public-key cryptosystem based on binary Reed–Muller codes, *Diskretn. Mat.* **6** (2), 3–20 (1994) [Russian] [*Discrete Math. Appl.* **4** (3), 191–207 (1994), DOI: 10.1515/dma.1994.4.3.191].
18. **L. Minder** and **A. Shokrollahi,** Cryptanalysis of the Sidelnikov cryptosystem, in *Advances in Cryptology — EUROCRYPT 2007*, Proc. 26th Annu. Int. Conf. Theory and Applications of Cryptographic Techniques (Barcelona, Spain, May 20–24, 2007) (Springer, Heidelberg, 2007), pp. 347–360 (Lect. Notes Comput. Sci., Vol. 4515), DOI: 10.1007/978-3-540-72540-4_20.
19. **R. Overbeck** and **N. Sendrier,** Code-based cryptography, in *Post-Quantum Cryptography* (Springer, Heidelberg, 2009), pp. 95–145, DOI: 10.1007/978-3-540-88702-7_4.
20. **M. A. González de la Torre, L. Hernández Encinas,** and **J. I. Sánchez García,** Structural analysis of code-based algorithms of the NIST post-quantum call, *Logic J. IGPL* **33** (5), ID jzae071 (2024), DOI: 10.1093/jigpal/jzae071.
21. **G. Alagic, M. Bros, P. Ciadoux,** [et al.], Status report on the fourth round of the NIST post-quantum cryptography standardization process (NIST, Gaithersburg, MD, 2025), DOI: 10.6028/NIST.IR.8545.
22. **G. Alagic, D. C. Apon, D. Cooper,** [et al.], Status report on the third round of the NIST post-quantum cryptography standardization process (NIST, Gaithersburg, MD, 2022), DOI: 10.6028/NIST.IR.8413-upd1.
23. **G. Alagic, J. Alperin-Sheriff, D. C. Apon,** [et al.], Status report on the second round of the NIST post-quantum cryptography standardization process (NIST, Gaithersburg, MD, 2020), DOI: 10.6028/NIST.IR.8309.
24. **M. R. Albrecht, D. J. Bernstein, T. Chou,** [et al.], Classic McEliece, in *Post-Quantum Cryptography. Round 3 Submissions* (NIST, Gaithersburg, MD, 2020), URL: csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions (accessed: 6.03.2026).
25. **D. J. Bernstein, T. Chou, C. Cid,** [et al.], Classic McEliece, in *Post-Quantum Cryptography. Round 4 Submissions* (NIST, Gaithersburg, MD, 2022), URL: csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-4-submissions (accessed: 6.03.2026).
26. **D. J. Bernstein, T. Chou, C. Cid,** [et al.], Classic McEliece. Implementation (Univ. Ill. Chic., Chicago, 2022), URL: classic.mceliece.org/impl.html (accessed: 6.03.2026).

27. CIRCL: Cloudflare interoperable reusable cryptographic library (Cloudflare, San Francisco, 2023), URL: `github.com/cloudflare/circl` (accessed: 6.03.2026).

28. Implement Classic McEliece (Cloudflare, San Francisco, 2022), URL: `github.com/cloudflare/circl/pull/378` (accessed: 6.03.2026).

29. Open Quantum Safe liboqs: C library for prototyping and experimenting with quantum-resistant cryptography, 2025, URL: `github.com/open-quantum-safe/liboqs/tree/main/src/kem/classic_mceliece` (accessed: 6.03.2026).

30. **T. Wiggers** and **D. Stebila,** Clean, portable, tested implementations of post-quantum cryptography, 2023, URL: `github.com/PQClean/PQClean` (accessed: 6.03.2026).

31. **A. Hülsing, K.-C. Ning, P. Schwabe, F. Weber,** and **P. R. Zimmermann,** Post-quantum WireGuard, in *Proc. 42nd IEEE Symp. Security and Privacy* (San Francisco, USA, May24–27, 2021) (IEEE Comput. Soc., Los Alamitos, CA, 2021), pp. 304–321, DOI: `10.1109/SP40001.2021.00030`.

32. Software co-design acceleration of Classic McEliece key encapsulation mechanism, 2021, URL: `github.com/beatsnbytes/classic_mceliece` (accessed: 6.03.2026).

33. Discrete math final project for 2018 — Implementation of the McEliece cryptosystem, 2018, URL: `github.com/arpanrau/McEliece-Implementation` (accessed: 6.03.2026).

34. **T. Nießen,** Purely educational PoC design and implementation of a PQC key exchange using Classic McEliece, 2019, URL: `github.com/tniessen/node-mceliece-key-exchange-poc` (accessed: 6.03.2026).

35. **D. J. Bernstein,** The McEliece cryptosystem, in *Talks 1st Post-Quantum Cryptography Summer School in Universities* (Chengdu, China, July 17, 2024), URL: `cr.yp.to/talks/2024.07.17/slides-djb-20240717-mceliece-4x3.pdf` (accessed: 6.03.2026).

36. Information technology. Cryptographic data security. Hash function, *GOST R 34.11—2018* (Standartinform, Moscow, 2018) [Russian].

37. **V. V. Vysotskaya** and **I. V. Chizhov,** Design criteria of a new code-based KEM, *J. Comput. Virol. Hacking Tech.* **20** (3), 497–511 (2024), DOI: `10.1007/s11416-024-00527-z`.

38. **J. Ge, H. Liao,** and **R. Xue,** Measure-rewind-extract: Tighter proofs of one-way to hiding and CCA security in the quantum random oracle model, in *Advances in Cryptology — ASIACRYPT 2024*, Proc. 30th Int. Conf. Theory and Application of Cryptology and Information Security (Kolkata, India, Dec. 9–13, 2024), Pt. IV (Springer, Singapore, 2024), pp. 3–34 (Lect. Notes Comput. Sci., Vol. 15487), DOI: `10.1007/978-981-96-0894-2_1`.

39. **J. Stern,** A new identification scheme based on syndrome decoding, in *Advances in Cryptology — CRYPTO'93*, Proc. 13th Annu. Int. Cryptology Conf. (Santa Barbara, USA, Aug. 22–26, 1993) (Springer, Heidelberg, 1994), pp. 13–21 (Lect. Notes Comput. Sci., Vol. 773), DOI: `10.1007/3-540-48329-2_2`.

40. **V. V. Vysotskaya, Chizhov I. V.** The security of the code-based signature scheme based on the Stern identification protocol, *Prikl. Diskretn. Mat.*, No. 57, 67–90 (2022) [Russian], DOI: 10.17223/20710410/57/5.
41. **K. D. Tsaregorodtsev,** Ternary forking lemma and its application to the analysis of one code-based signature, *Prikl. Diskretn. Mat.*, No. 59, 58–71 (2023) [Russian], DOI: 10.17223/20710410/59/3.
42. **V. V. Vysotskaya** and **D. K. Das,** Analyzing the resistance of the post-quantum signature "Shipovnik" to attacks against hash functions, in *Dokl. XXVI Int. Sci. Pract. Conf. RusCrypto* (Moscow, Russia, Mar. 19–22, 2024) (RusCrypto, Moscow, 2024), URL: ruscrypto.ru/resource/archive/rc2024/files/05_vysotskaya_das.pdf (accessed: 6.03.2026).
43. A public implementation of the signature algorithm "Shipovnik" for TK26 (QApp, Moscow, 2023) [Russian], URL: github.com/QAPP-tech/shipovnik_tc26 (accessed: 6.03.2026).
44. **E. Prange,** The use of information sets in decoding cyclic codes, *IRE Trans. Inf. Theory* **8** (5), 5–9 (1962), DOI: 10.1109/TIT.1962.1057777.
45. **N. Sendrier,** Decoding one out of many, in *Post-Quantum Cryptography*, Proc. 4th Int. Workshop (Taipei, China, Nov. 29 – Dec. 2, 2011) (Springer, Heidelberg, 2011), pp. 51–67 (Lect. Notes Comput. Sci., Vol. 7071), DOI: 10.1007/978-3-642-25405-5_4.
46. **C. Löndahl, T. Johansson, M. K. Shooshtari, M. Ahmadian-Attari,** and **M. R. Aref,** Squaring attacks on McEliece public-key cryptosystems using quasi-cyclic codes of even dimension, *Des. Codes Cryptogr.* **80** (2), 359–377 (2016), DOI: 10.1007/s10623-015-0099-x.
47. **Q. Guo, T. Johansson,** and **C. Löndahl,** A new algorithm for solving ring-lpn with a reducible polynomial, *IEEE Trans. Inf. Theory* **61** (11), 6204–6212 (2015), DOI: 10.1109/TIT.2015.2475738.
48. **Q. Guo, T. Johansson,** and **P. Stankovski,** A key recovery attack on MDPC with CCA security using decoding errors, in *Advances in Cryptology — ASIACRYPT 2016*, Proc. 22nd Int. Conf. Theory and Application of Cryptology and Information Security (Hanoi, Vietnam, Dec. 4–8, 2016), Pt. I (Springer, Heidelberg, 2016), pp. 789–815 (Lect. Notes Comput. Sci., Vol. 10031), DOI: 10.1007/978-3-662-53887-6_29.
49. **A. Nilsson, T. Johansson,** and **P. S. Wagner,** Error amplification in code-based cryptography, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2019** (1), 238–258 (2019), DOI: 10.46586/tches.v2019.i1.238-258.
50. **C. Aguilar-Melchor, O. Blazy, J.-C. Deneuville,** [et al.], Efficient encryption from random quasi-cyclic codes, *IEEE Trans. Inf. Theory* **64** (5), 3927–3943 (2018), DOI: 10.1109/TIT.2018.2804444.
51. **P. Gaborit, C. Aguilar-Melchor, N. Aragon,** [et al.], HQC cryptosystem specification (NIST, Gaithersburg, MD, 2025), URL: pqc-hqc.org/doc/hqc_specifications_2025_08_22.pdf (accessed: 6.03.2026).
52. **P. Gaborit, C. Aguilar-Melchor, N. Aragon,** [et al.], HQC. NIST submission packages, 2025, URL: pqc-hqc.org/doc/archive_submissions.zip (accessed: 6.03.2026).

53. P. Gaborit, C. Aguilar-Melchor, N. Aragon, [et al.], HQC. Optimized implementation, 2024, URL: `web.archive.org/web/20250712014511/pqc-hqc.org/doc/hqc-optimized-implementation_2024-10-30.zip` (accessed: 6.03.2026).

54. N. Aragon, P. Barreto, S. Bettaieb, [et al.], BIKE cryptosystem specification (NIST, Gaithersburg, MD, 2024), URL: `bikesuite.org/files/v5.2/BIKE_Spec.2024.10.10.1.pdf` (accessed: 6.03.2026).

55. R. Misoczki, J.-P. Tillich, N. Sendrier, [et al.], MDPC-McEliece: New McEliece variants from moderate density parity-check codes, in *Proc. 2013 IEEE Int. Symp. Information Theory* (Istanbul, Turkey, July 7–12, 2013) (IEEE, Piscataway, 2013), pp. 2069–2073, DOI: `10.1109/ISIT.2013.6620590`.

56. N. Aragon, P. Barreto, S. Bettaieb, [et al.], BIKE. Reference implementation, 2024, URL: `bikesuite.org/reference.html` (accessed: 6.03.2026).

57. Additional implementation of BIKE (AWS Labs, Seattle, 2024), URL: `github.com/awslabs/bike-kem` (accessed: 6.03.2026).

58. S. Mangard, E. Oswald, and **T. Popp,** *Power Analysis Attacks: Revealing the Secrets of Smart Cards* (Springer, New York, 2007), DOI: `10.1007/978-0-387-38162-6`.

59. F. X. Standaert, Introduction to side-channel attacks, in *Secure Integrated Circuits and Systems* (Springer, New York, 2010), pp. 27–42, DOI: `10.1007/978-0-387-71829-3_2`.

60. E. Peeters, F. X. Standaert, and **J. J. Quisquater,** Power and electromagnetic analysis: Improved model, consequences and comparisons, *Integration* **40** (1), 52–60 (2007), DOI: `10.1016/j.vlsi.2005.12.013`.

61. F. X. Standaert, F. Mace, E. Peeters, [et al.], Updates on the security of FPGAs against power analysis attacks, in *Reconfigurable Computing: Architectures and Applications*, Rev. Sel. Pap. 2nd Int. Workshop (Delft, The Netherlands, Mar. 1–3, 2006) (Springer, Heidelberg, 2006), pp. 335–346 (Lect. Notes Comput. Sci., Vol. 3985), DOI: `10.1007/11802839_42`.

62. A. E. Zhukov, Side channel attacks, *Inf. Secur., Inside*, No. 5, 28–33 (2010) [Russian].

63. S. Chari, J. R. Rao, and **P. Rohatgi,** Template attacks, in *Cryptographic Hardware and Embedded Systems — CHES 2002*, Rev. Pap. 4th Int. Workshop (Redwood Shores, CA, USA, Aug. 13–15, 2002) (Springer, Heidelberg, 2003), pp. 13–28 (Lect. Notes Comput. Sci., Vol. 2523), DOI: `10.1007/3-540-36400-5_3`.

64. R. Anderson and **M. Kuhn,** Tamper resistance — A cautionary note, in *Proc. 2nd USENIX Workshop Electronic Commerce* (Oakland, CA, USA, Nov. 18–21, 1996) (Carnegie Mellon Univ., Pittsburgh, PA, 1996), pp. 1–11.

65. P. Tuyls, G.-J. Schrijen, B. Škorić, [et al.], Read-proof hardware from protective coatings, in *Cryptographic Hardware and Embedded Systems — CHES 2006*, Proc. 8th Int. Workshop (Yokohama, Japan, Oct. 10–13, 2006) (Springer, Heidelberg, 2006), pp. 369–383 (Lect. Notes Comput. Sci., Vol. 4249), DOI: `10.1007/11894063_29`.

66. A. Shamir, Protecting smart cards from passive power analysis with detached power supplies, in *Cryptographic Hardware and Embedded Systems — CHES 2000*, Proc. 2nd Int. Workshop (Worcester, MA, USA, Aug. 17–18, 2000) (Springer, Heidelberg, 2000), pp. 71–77 (Lect. Notes Comput. Sci., Vol. 1965), DOI: 10.1007/3-540-44499-8_5.

67. L. Goubin and **J. Patarin,** DES and differential power analysis. The "Duplication" method, in *Cryptographic Hardware and Embedded Systems*, Proc. 1st Int. Workshop (Worcester, MA, USA, Aug. 12–13, 1999) (Springer, Heidelberg, 1999), pp. 158–172 (Lect. Notes Comput. Sci., Vol. 1717), DOI: 10.1007/3-540-48059-5_15.

68. D. May, H. L. Muller, and **N. P. Smart,** Random register renaming to foil DPA, in *Cryptographic Hardware and Embedded Systems — CHES 2001*, Proc. 3rd Int. Workshop (Paris, France, May 14–16, 2001) (Springer, Heidelberg, 2001), pp. 28–38 (Lect. Notes Comput. Sci., Vol. 2162), DOI: 10.1007/3-540-44709-1_4.

69. S. Chari, S. C. Jutla, R. J. Rao, [et al.], Towards sound approaches to counteract power-analysis attacks, in *Advances in Cryptology — CRYPTO'99*, Proc. 19th Annu. Int. Cryptology Conf. (Santa Barbara, USA, Aug. 15–19, 1999) (Springer, Heidelberg, 1999), pp. 398–412 (Lect. Notes Comput. Sci., Vol. 1666), DOI: 10.1007/3-540-48405-1_26.

70. R. Ueno, N. Homma, and **T. Aoki,** Toward more efficient DPA-resistant AES hardware architecture based on threshold implementation, in *Constructive Side-Channel Analysis and Secure Design*, Rev. Sel. Pap. 8th Int. Workshop (Paris, France, Apr. 13–14, 2017) (Springer, Cham, 2017), pp. 50–64 (Lect. Notes Comput. Sci., Vol. 10348), DOI: 10.1007/978-3-319-64647-3_4.

71. P. Schwabe and **K. Stoffelen,** All the AES you need on Cortex-M3 and M4, in *Selected Areas in Cryptography — SAC 2016*, Rev. Sel. Pap. 23rd Int. Conf. (St. John's, NL, Canada, Aug. 10–12, 2016) (Springer, Cham, 2016), pp. 180–194 (Lect. Notes Comput. Sci., Vol. 10532), DOI: 10.1007/978-3-319-69453-5_10.

72. K. Tiri, M. Akmal, and **I. Verbauwhede,** A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards, in *Proc. 28th European Solid-State Circuits Conf.* (Florence, Italy, Sept. 24–26, 2002) (IEEE, Piscataway, 2002), pp. 403–406.

73. N. Patterson, The algebraic decoding of Goppa codes, *IEEE Trans. Inf. Theory* **21** (2), 203–207 (1975), DOI: 10.1109/TIT.1975.1055350.

74. F. Strenzke, E. Tews, G. Molter, [et al.], Side channels in the McEliece PKC, in *Post-Quantum Cryptography*, Proc. 2nd Int. Workshop (Cincinnati, OH, USA, Oct. 17–19, 2008) (Springer, Heidelberg, 2008), pp. 216–229 (Lect. Notes Comput. Sci., Vol. 5299), DOI: 10.1007/978-3-540-88403-3_15.

75. R. Avanzi, S. Hoerder, D. Page, [et al.], Side-channel attacks on the McEliece and Niederreiter public-key cryptosystems, *J. Cryptogr. Eng.* **1** (4), 271–281 (2011), DOI: 10.1007/s13389-011-0024-9.

76. A. Shoufan, F. Strenzke, H. G. Molter, [et al.], A timing attack against Patterson algorithm in the McEliece PKC, in *Information Security and Cryptology — ICISC 2009*, Rev. Sel. Pap. 12th Int. Conf. (Seoul, Korea, Dec. 2–4, 2009) (Springer, Heidelberg, 2010), pp. 161–175 (Lect. Notes Comput. Sci., Vol. 5984), DOI: 10.1007/978-3-642-14423-3_12.

77. F. Strenzke, A timing attack against the secret permutation in the McEliece PKC, in *Post-Quantum Cryptography*, Proc. 3rd Int. Workshop (Darmstadt, Germany, May 25–28, 2010) (Springer, Heidelberg, 2010), pp. 95–107 (Lect. Notes Comput. Sci., Vol. 6061), DOI: 10.1007/978-3-642-12929-2_8.

78. F. Strenzke, Timing attacks against the syndrome inversion in code-based cryptosystems, in *Post-Quantum Cryptography*, Proc. 5th Int. Workshop (Limoges, France, June 4–7, 2013) (Springer, Heidelberg, 2013), pp. 217–230 (Lect. Notes Comput. Sci., Vol. 7932).

79. S. Heyse, A. Moradi, and **C. Paar,** Practical power analysis attacks on software implementations of McEliece, in *Post-Quantum Cryptography*, Proc. 3rd Int. Workshop (Darmstadt, Germany, May 25–28, 2010) (Springer, Heidelberg, 2010), pp. 108–125 (Lect. Notes Comput. Sci., Vol. 6061), DOI: 10.1007/978-3-642-12929-2_9.

80. H. G. Molter, M. Stöttinger, A. Shoufan, [et al.], A simple power analysis attack on a McEliece cryptoprocessor, *J. Cryptogr. Eng.* **1** (1), 29–36 (2011), DOI: 10.1007/s13389-011-0001-3.

81. A. Shoufan, T. Wink, H. G. Molter, [et al.], A novel processor architecture for McEliece cryptosystem and FPGA platforms, in *Proc. 20th IEEE Int. Conf. Application-Specific Systems, Architectures and Processors* (Boston, MA, USA, July 7–9, 2009) (IEEE Comput. Soc., Los Alamitos, CA, 2009), pp. 98–105, DOI: 10.1109/ASAP.2009.29.

82. S. Heyse, I. Von Maurich, and **T. Güneysu,** Smaller keys for code-based cryptography: QC-MDPC McEliece implementations on embedded devices, in *Cryptographic Hardware and Embedded Systems — CHES 2013*, Proc. 15th Int. Workshop (Santa Barbara, USA, Aug. 20–23, 2013) (Springer, Heidelberg, 2013), pp. 273–292 (Lect. Notes Comput. Sci., Vol. 8086), DOI: 10.1007/978-3-642-40349-1_16.

83. I. Von Maurich, and **T. Güneysu,** Towards side-channel resistant implementations of QC-MDPC McEliece encryption on constrained devices, in *Post-Quantum Cryptography*, Proc. 6th Int. Workshop (Waterloo, ON, Canada, Oct. 1–3, 2014) (Springer, Cham, 2014), pp. 266–282 (Lect. Notes Comput. Sci., Vol. 8772), DOI: 10.1007/978-3-319-11659-4_16.

84. I. Von Maurich, and **T. Güneysu,** Lightweight code-based cryptography: QC-MDPC McEliece encryption on reconfigurable devices, in *2014 Design, Automation and Test in Europe Conf.* (Dresden, Germany, Mar. 24–28, 2014) (IEEE, Piscataway, 2014), pp. 1–6, DOI: 10.7873/DATE.2014.051.

85. C. Chen, T. Eisenbarth, and **I. Von Maurich,** [et al.], Differential power analysis of a McEliece cryptosystem, in *Applied Cryptography and Network Security*, Rev. Sel. Pap. 13th Int. Conf. (New York, USA, June 2–5, 2015) (Springer, Cham, 2015), pp. 538–556 (Lect. Notes Comput. Sci., Vol. 9092).

86. T. Schamberger, J. Renner, G. Sigl, [et al.], A power side-channel attack on the CCA2-secure HQC KEM, in *Smart Card Research and Advanced Applications*, Rev. Sel. Pap. 19th Int. Conf. (Lübeck, Germany, Nov. 18–19, 2020) (Springer, Cham, 2020), pp. 119–134 (Lect. Notes Comput. Sci., Vol. 12609), DOI: 10.1007/978-3-030-68487-7_8.

87. C. Hlauschek, N. Lahr, and **R. L. Schröder,** On the timing leakage of the deterministic re-encryption in HQC KEM (Univ. California, San Diego, 2021) (Cryptol. ePrint Archive, Pap. 2021/1485/20211115:124514), URL: eprint.iacr.org/archive/2021/1485/20211115:124514 (accessed: 6.03.2026).

88. G. Wafo-Tapa, S. Bettaieb, L. Bidoux, [et al.], A practicable timing attack against HQC and its countermeasure, *Adv. Math. Commun.* **16** (3), 621–642 (2022), DOI: 10.3934/amc.2020126.

89. T. B. Paiva and **R. Terada,** A timing attack on the HQC encryption scheme, in *Selected Areas in Cryptography — SAC 2019*, Rev. Sel. Pap. 26th Int. Conf. (Waterloo, ON, Canada, Aug. 12–16, 2019) (Springer, Cham, 2019), pp. 551–573 (Lect. Notes Comput. Sci., Vol. 11959), DOI: 10.1007/978-3-030-38471-5_22.

90. N. Lahr, R. Niederhagen, R. Petri, [et al.], Side channel information set decoding using iterative chunking: Plaintext recovery from the "Classic McEliece" hardware reference implementation, in *Advances in Cryptology — ASIACRYPT 2020*, Proc. 26th Int. Conf. Theory and Application of Cryptology and Information Security (Daejeon, South Korea, Dec. 7–11, 2020), Pt. I (Springer, Cham, 2020), pp. 881–910 (Lect. Notes Comput. Sci., Vol. 12491), DOI: 10.1007/978-3-030-64837-4_29.

91. Q. Guo, A. Johansson, and **T. Johansson,** A key-recovery side-channel attack on Classic McEliece implementations, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2022** (4), 800–827 (2022), DOI: 10.46586/tches.v2022.i4.800-827.

92. S. Pircher, J. Geier, J. Danner, [et al.], Key-recovery fault injection attack on the Classic McEliece KEM, in *Code-Based Cryptography*, Rev. Sel. Pap. 10th Int. Workshop (Trondheim, Norway, May 29–30, 2022) (Springer, Cham, 2022), pp. 37–61 (Lect. Notes Comput. Sci., Vol. 13839), DOI: 10.1007/978-3-031-29689-5_3.

93. V. Grosso, P.-L. Cayrel, B. Colombier, [et al.], Punctured syndrome decoding problem: Efficient side-channel attacks against Classic McEliece, in *Constructive Side-Channel Analysis and Secure Design*, Proc. 14th Int. Workshop (Munich, Germany, Apr. 3–4, 2023) (Springer, Cham, 2023), pp. 170–192 (Lect. Notes Comput. Sci., Vol. 13979), DOI: 10.1007/978-3-031-29497-6_9.

94. M. Brinkmann, C. Chuengsatiansup, A. May, [et al.], Leaky McEliece: Secret key recovery from highly erroneous side-channel information, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2025** (2), 94–125 (2025), DOI: 10.46586/tches.v2025.i2.94-125.

95. S. Bitzer, J. Delvaux, E. Kirshanova, [et al.], How to lose some weight: A practical template syndrome decoding attack, *Des. Codes Cryptogr.* **93** (7), 2503–2519 (2025), DOI: 10.1007/s10623-025-01603-1.

96. V.-F. Drăgoi, B. Colombier, N. Vallet, [et al.], Full key-recovery cubic-time template attack on Classic McEliece decapsulation (Univ. California, San Diego, 2024) (Cryptol. ePrint Archive, Pap. 2024/1694), URL: eprint.iacr.org/2024/1694 (accessed: 6.03.2026).

97. R. Ueno, K. Xagawa, Y. Tanaka, [et al.], Curse of re-encryption: A generic power/EM analysis on post-quantum KEMs, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2022** (1), 296–322 (2022), DOI: 10.46586/tches.v2022.i1.296-322.

98. D. J. Bernstein, T. Lange, and **C. Peters,** Attacking and defending the McEliece cryptosystem, in *Post-Quantum Cryptography*, Proc. 2nd Int. Workshop (Cincinnati, OH, USA Oct. 17–19, 2008) (Springer, Heidelberg, 2008), pp. 31–46 (Lect. Notes Comput. Sci., Vol. 5299), DOI: 10.1007/978-3-540-88403-3_3.

99. A. Canteaut and **F. Chabaud,** A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511, *IEEE Trans. Inf. Theory* **44** (1), 367–378 (1998), DOI: 10.1109/18.651067.

100. A. Canteaut and **N. Sendrier,** Cryptanalysis of the original McEliece cryptosystem, in *Advances in Cryptology — ASIACRYPT'98*, Proc. Int. Conf. Theory and Application of Cryptology and Information Security (Beijing, China, Oct. 18–22, 1998) (Springer, Heidelberg, 1998), pp. 187–199 (Lect. Notes Comput. Sci., Vol. 1514), DOI: 10.1007/3-540-49649-1_16.

101. E. Eaton, M. Lequesne, A. Parent, [et al.], QC-MDPC: A timing attack and a CCA2 KEM, in *Post-Quantum Cryptography*, Proc. 9th Int. Conf. (Fort Lauderdale, FL, USA, Apr. 9–11, 2018) (Springer, Cham, 2018), pp. 47–76 (Lect. Notes Comput. Sci., Vol. 10786), DOI: 10.1007/978-3-319-79063-3_3.

102. N. Drucker, S. Gueron, and **D. Kostic,** QC-MDPC decoders with several shades of gray, in *Post-Quantum Cryptography*, Proc. 11th Int. Conf. (Paris, France, Apr. 15–17, 2020) (Springer, Cham, 2020), pp. 35–50 (Lect. Notes Comput. Sci., Vol. 12100), DOI: 10.1007/978-3-030-44223-1_3.

103. T. Schamberger, L. Holzbaur, J. Renner, [et al.], A power side-channel attack on the Reed–Muller Reed–Solomon version of the HQC cryptosystem, in *Post-Quantum Cryptography*, Proc. 13th Int. Conf. (Eindhoven, The Netherlands, Sept. 28-30, 2022) (Springer, Cham, 2022), pp. 327–352 (Lect. Notes Comput. Sci., Vol. 13512), DOI: 10.1007/978-3-031-17234-2_16.

104. G. Goy, A. Loiseau, and **P. Gaborit,** A new key recovery side-channel attack on HQC with chosen ciphertext, in *Post-Quantum Cryptography*, Proc. 13th Int. Conf. (Eindhoven, The Netherlands, Sept. 28–30, 2022) (Springer, Cham, 2022), pp. 353–371 (Lect. Notes Comput. Sci., Vol. 13512), DOI: 10.1007/978-3-031-17234-2_17.

105. T. Chou, QcBits: Constant-time small-key code-based cryptography, in *Cryptographic Hardware and Embedded Systems — CHES 2016*, Proc. 18th Int. Conf. (Santa Barbara, USA, Aug. 17–19, 2016) (Springer, Heidelberg, 2016), pp. 280–300 (Lect. Notes Comput. Sci., Vol. 9813), DOI: 10.1007/978-3-662-53140-2_14.

106. M. Rossi, M. Hamburg, M. Hutter, [et al.], A side-channel assisted cryptanalytic attack against QcBits, in *Cryptographic Hardware and Embedded Systems — CHES 2017*, Proc. 19th Int. Conf. (Taipei, China, Sept. 25–28, 2017) (Springer, Cham, 2017), pp. 3–23 (Lect. Notes Comput. Sci., Vol. 10529), DOI: 10.1007/978-3-319-66787-4_1.

107. B.-Y. Sim, J. Kwon, K. Y. Choi, [et al.], Novel side-channel attacks on quasi-cyclic code-based cryptography, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2019** (1), 180–212 (2019), DOI: 10.46586/tches.v2019.i4.180-212.

108. Q. Guo, C. Hlauschek, T. Johansson, [et al.], Don't reject this: Key-recovery timing attacks due to rejection-sampling in HQC and BIKE, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2022** (3), 223–263 (2022), DOI: 10.46586/tches.v2022.i3.223-263.

109. N. Sendrier, Secure sampling of constant-weight words — Application to BIKE (Univ. California, San Diego, 2021) (Cryptol. ePrint Archive, Pap. 2021/1631), URL: eprint.iacr.org/2021/1631 (accessed: 6.03.2026).

110. S. Huang, Q. R. Sim, C. Chuengsatiansup, [et al.], Cache-timing attack against HQC (Univ. California, San Diego, 2023) (Cryptol. ePrint Archive, Pap. 2023/102), URL: eprint.iacr.org/2023/102 (accessed: 6.03.2026).

111. G. Goy, J. Maillard, P. Gaborit, [et al.], Single trace HQC shared key recovery with SASCA, *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2024** (2), 64–87 (2024), DOI: 10.46586/tches.v2024.i2.64-87.

Aleksandr O. Bakharev
Denis M. Voronov
Nikolay A. Kolomeec
Natalia N. Tokareva
Irina S. Khilchuk
Aleksandr S. Shaporenko

Received June 18, 2025
Revised August 11, 2025
Accepted September 22, 2025