МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 621.391.7 DOI 10.17223/20710410/67/1

О ПАРАМЕТРАХ СИСТЕМЫ ШИФРОВАНИЯ ТИПА МАК-ЭЛИСА НА D-КОДАХ, ОСНОВАННЫХ НА ДВОИЧНЫХ КОДАХ РИДА — МАЛЛЕРА

Ю. В. Косолапов, Е. А. Лелюк

Южный федеральный университет, г. Ростов-на-Дону, Россия

E-mail: yvkosolapov@sfedu.ru, lelukevgeniy@mail.ru

Исследуются характеристики кодовой системы шифрования типа Мак-Элиса на специальной сумме тензорных произведений базовых кодов, называемой D-кодом. В качестве базовых выбраны двоичные коды Рида — Маллера. Ранее были найдены условия на эти D-коды, при выполнении которых соответствующая система шифрования устойчива к известным структурным атакам на основе произведения Шура — Адамара. Однако при использовании декодера, работающего в пределах половины кодового расстояния, система типа Мак-Элиса на D-кодах обеспечивает стойкость, сравнимую со стойкостью классической системы шифрования Мак-Элиса на кодах Гоппы, при существенно большем размере ключа. В настоящей работе построены два вероятностных декодера для D-кодов. В случае использования этих декодеров найдены параметры некоторых D-кодов, которые при меньшем размере ключа, чем в классической системе, обеспечивают сравнимую стойкость к атаке декодированием по информационным совокупностям. Однако наличие непренебрежимой вероятности ошибочного декодирования пока ограничивает область применения системы шифрования на D-кодах эфемерными механизмами защищённой передачи сеансового ключа (IND-CPA KEM).

Ключевые слова: *D-коды, схема Мак-Элиса, механизм защищённой передачи сеансового ключа.*

ON THE PARAMETERS OF A McELIECE-TYPE CRYPTOSYSTEM ON D-CODES BASED ON BINARY REED — MULLER CODES

Yu. V. Kosolapov, E. A. Lelyuk

Southern Federal University, Rostov-on-Don, Russia

The characteristics of a McEliece-type code cryptosystem on a special sum of tensor products of base codes, called D-code, are investigated. Binary Reed — Muller codes were chosen as the base codes. Previously, conditions were found for these D-codes, under which the corresponding cryptosystem is resistant to known structural attacks based on the Schur — Hadamard product. However, when using a decoder operating within half the code distance, a McEliece-type system on D-codes provides security comparable to the strength of the classical McEliece cryptosystem on Goppa codes, with a significantly larger key size. In this paper, two probabilistic decoders for D-codes are constructed. In the case of using these decoders, parameters

of some D-codes have been found that provide comparable resistance to information set decoding type attacks, while having a smaller key size than in the classical system. However, the presence of a non-negligible decoding failure rate currently limits the scope of application of the D-code cryptosystem to ephemeral session key encapsulation mechanisms (IND-CPA KEM).

Keywords: *D-codes, McEliece scheme, key encapsulation mechanism.*

Введение

Класс кодов Гоппы в настоящее время рассматривается как основной для системы шифрования типа Мак-Элиса. Это обусловлено тем, что большинство попыток заменить коды Гоппы другими кодами с целью уменьшить размер открытого ключа не увенчалось успехом из-за их слабой стойкости к структурным атакам (атакам восстановления ключа) [1]. Примечательно, что в 1978 г. Р. Мак-Элис предложил новую концепцию асимметричной системы шифрования именно на кодах Гоппы [2]. Далее эту систему будем называть оригинальной, или Original McEliece. В конкурсе NIST [4] система шифрования Мак-Элиса на кодах Гоппы носит название Classic McEliece [3]. Отличие Original McEliece от Classic McEliece заключается в правиле шифрования: в Classic McEliece шифрование выполняется по схеме, предложенной Х. Нидеррайтером в [5], так как это позволяет сократить размер открытого ключа. Заметим, что фактически схема Original McEliece являлась основой проекта NTS-KEM [6], который позже был объединён с проектом Classic McEliece по причине использования одного класса кодов Гоппы.

Отсутствие в настоящий момент эффективных структурных атак для схемы Мак-Элиса на кодах Гоппы, как представляется, только мотивирует исследователей к поиску новых способов её раскалывания [7–9]. В этой связи актуальным является не только построение принципиально новых кодовых систем шифрования, что делается в [10–12], но и поиск новых кодов для схемы Мак-Элиса. Необходимыми условиями для таких кодов являются, во-первых, стойкость к известным структурным атакам и, во-вторых, наличие эффективного декодера.

Результаты исследования стойкости системы типа Мак-Элиса на D-кодах на основе двоичных кодов Рида — Маллера [13, 14] и наличие эффективного декодера [15] позволяют рассматривать D-коды в качестве такой альтернативы. Однако стоит отметить, что высокая стойкость к атакам на шифртекст в случае использования декодеров D-кодов, работающих в пределах половины кодового расстояния, достигается только при больших размерах открытого ключа [14]. Это связано с небольшим кодовым расстоянием D-кодов. Повысить стойкость к атакам на шифртекст можно путём увеличения веса добавляемого при шифровании вектора ошибок. Это, в свою очередь, позволяет для заданного уровня стойкости уменьшить размер ключа, используя код с меньшими значениями длины и размерности. В частных случаях рассматриваемые в [14] D-коды являются подкодами кодов Рида — Маллера. Так как для двоичных кодов Рида — Маллера известны декодеры, которые позволяют исправлять ошибки веса, существенно большего половины кодового расстояния (см. обзор [16]), то при использовании таких декодеров существует потенциальная возможность сокращения размера ключа системы на D-кодах при сохранении её высокой стойкости к атакам восстановления ключа и атакам на шифртекст.

В [14] указаны параметры D-кодов, для которых при использовании декодеров, работающих за пределами половины кодового расстояния, может быть обеспечена

стойкость соответствующей системы шифрования типа Мак-Элиса не ниже стойкости систем на кодах Гоппы при сопоставимых размерах ключа. Стоит отметить, что сопоставимые размеры ключа достигаются при большей длине D-кодов относительно кодов Гоппы, то есть при уменьшении информационной скорости. Большая длина D-кодов, в свою очередь, ограничивает применение известных декодеров кодов Рида — Маллера для непосредственного декодирования D-кодов из-за их высокой вычислительной сложности и сложности по требуемой памяти. Применение декодера из [15] для D-кодов также ограничено высокой сложностью по памяти. Поэтому вопрос эффективного декодирования D-кодов большой длины остаётся актуальным.

В настоящей работе для эффективного декодирования D-кодов используется блочная структура кодовых слов: каждый блок декодируется декодером кода Рида — Маллера меньшей длины, а затем в случае неправильного декодирования выполняется блочное декодирование по информационным совокупностям. Такой подход позволяет существенно ускорить расшифрование, несмотря на «дорогую» операцию блочного декодирования по информационным совокупностям, а использование для кодов Рида — Маллера декодера из [17], который исправляет ошибки веса, существенно большего половины кодового расстояния, позволяет увеличить вес добавляемой ошибки и сократить размер открытого ключа при обеспечении высокой стойкости.

Работа, кроме введения и заключения, состоит из пяти пунктов. В п. 1 формально описаны схема асимметричного шифрования и механизм защищённой передачи сеансового ключа KEM на основе этой схемы. В п. 2 в компактном виде приводится описание системы шифрования типа Мак-Элиса в случае произвольного линейного кода, а также механизм защищённой передачи сеансового ключа KEM на основе системы типа Мак-Элиса. В п. 3 рассмотрено построение декодеров для D-кодов, применимых в схеме типа Мак-Элиса. В п. 4 содержатся результаты исследования, цель которого — поиск параметров D-кодов для системы типа Мак-Элиса, обеспечивающих сопоставимые с системами шифрования на кодах Гоппы стойкость и размер открытого ключа при использовании построенных декодеров. В п. 5 приводятся характеристики механизмов KEM на основе системы шифрования на D-кодах.

1. Схема асимметричного шифрования и механизм KEM

При описании криптографических схем будем придерживаться следующих обозначений: случайный и равновероятный выбор элемента a из множества A будем обозначать $a \leftarrow A$; генерацию элемента a с помощью алгоритма Alg — соответственно $a \leftarrow \mathsf{Alg}$; обращение к алгоритму Alg, не принимающему параметры, — $\mathsf{Alg}(\cdot)$.

Схема асимметричной системы шифрования (PKE, Public Key Encryption) для заданного множества сообщений $\mathcal{M}$ может быть описана тройкой $\mathsf{PKE} = (\mathsf{Gen}^{\mathsf{PKE}}, \mathsf{Enc}, \mathsf{Dec})$. Здесь

1) $\mathsf{Gen}^{\mathsf{PKE}}$ — полиномиальный вероятностный алгоритм генерации ключей pk и sk $(pk, sk) \leftarrow \mathsf{Gen}^{\mathsf{PKE}}(\cdot)$: закрытого sk и соответствующего ему открытого pk;
2) Enc — полиномиальный вероятностный алгоритм шифрования, преобразующий произвольное сообщение $\mathbf{m} \in \mathcal{M}$ в шифртекст $\mathbf{c}$: $\mathbf{c} \leftarrow \mathsf{Enc}(pk, \mathbf{m}; \mathbf{r})$, где $\mathbf{r}$ — случайное значение, которое может вырабатываться при шифровании;
3) Dec — полиномиальный алгоритм расшифрования, такой, что $\mathsf{Dec}(sk, \mathbf{c})$ возвращает $\mathbf{m}$, если $\mathbf{c}$ представляет шифртекст, для которого известен полиномиальный алгоритм нахождения $\mathbf{m}$ с помощью sk; в противном случае $\mathsf{Dec}(sk, \mathbf{c})$ возвращает сообщение об ошибке $\perp \notin \mathcal{M}$.

Под OW-стойкостью (One-Way-стойкостью) схемы PKE обычно понимают стойкость к атакам дешифрования шифртекста, полученного при шифровании случайно и равновероятно выбранного открытого текста. При этом уровень стойкости $\lambda_{\mathsf{OW}} \in \mathbb{N}$ может быть определён как отрицательный двоичный логарифм вероятности успеха наилучшей из таких атак.

Асимметричные шифрсистемы обычно не используются для защиты непосредственно пользовательских данных, но при этом являются основой многих криптографических протоколов, в частности механизмов защищённой передачи сеансового ключа (KEM, Key Encapsulation Mechanism). Одной из возможных целей протокола KEM является передача от отправителя к получателю сеансового ключа $\mathbf{K}$ ($\mathbf{K} \leftarrow \{0,1\}^m$) симметричного шифра (где обычно $m \in \{128, 192, 256\}$), на котором выполняется шифрование пользовательских данных.

Механизм защищённой передачи для фиксированного множества сеансовых ключей $\mathcal{K}$ и схемы $\mathsf{PKE} = (\mathsf{Gen}^{\mathsf{PKE}}, \mathsf{Enc}, \mathsf{Dec})$ может быть описан тройкой алгоритмов $\mathsf{KEM} = (\mathsf{Gen}^{\mathsf{KEM}}, \mathsf{Encaps}, \mathsf{Decaps})$. Здесь

1) $\mathsf{Gen}^{\mathsf{KEM}}$ — полиномиальный вероятностный алгоритм генерации открытого pk' и закрытого sk' ключей (строится на основе $\mathsf{Gen}^{\mathsf{PKE}}$);
2) Encaps — полиномиальный вероятностный алгоритм получения пары $(\mathbf{K}, \mathbf{c})$: $(\mathbf{K}, \mathbf{c}) \leftarrow \mathsf{Encaps}(pk')$, где $\mathbf{c}$ называется инкапсуляцией сеансового ключа $\mathbf{K} \in \mathcal{K}$ (строится на основе Enc);
3) Decaps — полиномиальный алгоритм извлечения (декапсуляции) сеансового ключа $\mathbf{K}$ из $\mathbf{c}$ (строится на основе Dec). Алгоритм $\mathsf{Decaps}(sk', \mathbf{c})$ возвращает $\mathbf{K}$, если $\mathbf{c}$ представляет корректную инкапсуляцию ключа $\mathbf{K}$, в противном случае $\mathsf{Decaps}(sk', \mathbf{c})$ может возвращать сообщение об ошибке $\perp \notin \mathcal{K}$ (явный отказ, explicit rejection) либо вектор, отличный от $\mathbf{K}$, но обычно зависящий от $\mathbf{c}$ и sk' (неявный отказ, implicit rejection).

Уровень стойкости $\lambda_{\mathsf{KEM}} \in \mathbb{N}$ механизма KEM может быть определён как отрицательный двоичный логарифм вероятности успеха наилучшей известной атаки на KEM. Обычно для механизмов KEM, основанных на схемах PKE, выполняется неравенство

$$\lambda_{\mathsf{OW}} \geqslant \lambda_{\mathsf{KEM}} \geqslant m. \tag{1}$$

Механизмы KEM условно можно разделить на механизмы с многоразовым использованием пары $(pk', sk') \leftarrow \mathsf{Gen}^{\mathsf{KEM}}$ и с одноразовым, или эфемерным использованием. Вероятность

$$1 - \delta = 1 - \Pr\left[\perp \leftarrow \mathsf{Decaps}(sk', \mathbf{c}) : (pk', sk') \leftarrow \mathsf{Gen}^{\mathsf{KEM}},\ (\mathbf{K}, \mathbf{c}) \leftarrow \mathsf{Encaps}(pk')\right] \tag{2}$$

события, заключающегося в успешной декапсуляции сеансового ключа, назовём *отказоустойчивостью* KEM. Эта вероятность может влиять как на удобство использования механизма, так и на его стойкость. Неудобства, связанные с ненулевым δ, заключаются в следующем. При многоразовом использовании в случае ошибки декапсуляции отправителю сеансового ключа необходимо повторно использовать алгоритм Encaps, а получателю — алгоритм Decaps; в случае одноразового использования KEM получателю, помимо этого, необходимо генерировать новую пару $(pk', sk') \leftarrow \mathsf{Gen}^{\mathsf{KEM}}$ и передавать pk' отправителю.

Наиболее значительные различия механизмов защищённой передачи с многоразовым и одноразовым использованием проявляются во влиянии значения δ на стойкость этих механизмов. Обычно выделяют два типа стойкости механизма KEM: стойкие к

атаке на основе подобранного открытого текста (IND-CPA, INDistinguishability under Chosen Plaintext Attack) и стойкие к атаке на основе подобранного шифртекста (IND-CCA1/2, INDistinguishability under Chosen Ciphertext Attack).

IND-CCA-модель соответствует многоразовому использованию ключа, так как в рамках этой модели атакующий имеет возможность многократно подбирать шифртекст и наблюдать результаты декапсуляции. Следовательно, при многоразовом использовании пары (pk', sk') не исключается возможность получения атакующим информации об sk' на основании информации об ошибках декапсуляции [18]. Например, для механизма KEM BIKE [19] и его некоторых вариаций большое значение δ позволяет провести атаку с опорой на реакцию, нацеленную на восстановление закрытого ключа [20, 21]. Поэтому для многоразового использования ключей от KEM требуется обеспечение IND-CCA-стойкости уровня λ_{KEM}, для достижения которой, в свою очередь, требуется, чтобы $\delta \leqslant 2^{-\lambda_{\mathsf{KEM}}}$. Стоит отметить, что для механизмов KEM с IND-CCA-стойкостью у исследователей пока нет единого мнения в пользу явного либо неявного отказа при декапсуляции в алгоритме Decaps [22], хотя в механизмах из конкурса NIST используется неявный отказ.

В случае эфемерной защищённой передачи информация об ошибке декапсуляции представляется малопригодной для атакующего. Поэтому для таких механизмов достаточно IND-CPA-стойкости [23] и, согласно [24], вероятность ошибочного декодирования влияет только на отказоустойчивость информационной системы. Верхняя граница значения δ в (2) определяется тем, насколько часто в информационной системе допускается повторное выполнение KEM парой участников информационного взаимодействия. Верхнюю границу δ можно определить в виде $10^{-\gamma}$, где $\gamma \in \mathbb{N}$. Далее параметр γ будем называть *уровнем отказоустойчивости* информационной системы, в которой используется механизм KEM, или кратко — уровнем отказоустойчивости KEM. В [24] в качестве допустимых значений γ рассматриваются значения из множества $\{5, 6, 7\}$, при этом отмечается, что $\gamma = 5$ считается «золотым стандартом» отказоустойчивости информационных систем. В [25] в качестве допустимого рассматривается значение $\gamma = 9$. Отметим также, что в эфемерных механизмах KEM может использоваться явный отказ вместо неявного, что позволяет сократить размер закрытого ключа sk' и время декапсуляции [24].

Таким образом, условие на вероятность δ в (2) для одноразового и многоразового использования ключей можно сформулировать так:

$$\delta \leqslant \begin{cases} 2^{-\lambda_{\mathsf{KEM}}}, & \text{многоразовое использование,} \\ 10^{-\gamma}, & \text{одноразовое использование,} \end{cases} \tag{3}$$

где λ_{KEM} — требуемый уровень IND-CCA-стойкости механизма KEM; γ — требуемый уровень отказоустойчивости IND-CPA-стойкого механизма KEM.

2. Система шифрования типа Мак-Элиса и механизм KEM на её основе

Введём необходимые обозначения. Пусть $\mathbb{F}_q$ — поле Галуа мощности q. Линейным $[n, k, d]_q$-кодом C будем называть подпространство пространства $\mathbb{F}_q^n$ размерности k с минимальным кодовым расстоянием $d = \min\{\mathrm{wt}(\mathbf{c}) : \mathbf{c} \in C \setminus \{\mathbf{0}\}\}$, где $\mathrm{wt}(\mathbf{c})$ — вес Хэмминга (далее — просто вес) вектора $\mathbf{c}$; $\mathbf{0}$ — нулевой вектор из $\mathbb{F}_q^n$. В ряде случаев будем опускать d и писать $[n, k]_q$-код, если контекст не требует явного указания этого параметра. Декодером кода C будем называть отображение $\mathsf{Decoder} : \mathbb{F}_q^n \to C \cup \{\perp\}$, где $\perp$ — сообщение об ошибке, возвращаемое декодером при невозможности декодирования входного вектора. Сообщение об ошибке $\perp$ может возвращаться декодером

Decoder в случае, когда, например, входной вектор не может быть однозначно декодирован. Алгоритм, реализующий декодирование, также будем обозначать Decoder. Отметим, что входными параметрами такого алгоритма, кроме декодируемого вектора $\mathbf{z}$, могут быть специфичные для кода и/или декодера параметры, например условие выхода из алгоритма и т. п. Такие специфичные параметры $p_1, p_2, \dots$ будем отделять точкой с запятой и писать $\mathsf{Decoder}(\mathbf{z}; p_1, p_2, \dots)$.

Пусть C — линейный $[n, k, d]_q$-код с фиксированной порождающей матрицей G, для которого известен эффективный декодер Decoder; G^{-1} — псевдообратная к G матрица, то есть такая матрица, что GG^{-1} — единичная $(k \times k)$-матрица. Пусть также $\mathrm{GL}_k(\mathbb{F}_q)$ — кольцо обратимых $(k \times k)$-матриц; $\mathcal{P}_n$ — кольцо $(n \times n)$-матриц перестановок; $\mathcal{E}_{n,t} = \{\mathbf{y} \in \mathbb{F}_q^n : \mathrm{wt}(\mathbf{y}) = t\}$. Схему $\mathsf{PKE} = (\mathsf{Gen}^{\mathsf{PKE}}, \mathsf{Enc}, \mathsf{Dec})$ типа Мак-Элиса на основе кода C обозначим $\mathsf{McE}(C) = (\mathsf{Gen}^{\mathsf{McE}(C)}, \mathsf{Enc}, \mathsf{Dec})$; соответствующие алгоритмы приведены на рис. 1. Отметим, что здесь и далее рассматривается схема Мак-Элиса, когда код C не является секретным, поэтому матрица G является частью алгоритмов $\mathsf{Gen}^{\mathsf{McE}(C)}$ и Dec. Так как в ряде случаев декодер кода C может исправлять векторы ошибок веса больше, чем $\lfloor (d-1)/2 \rfloor$, то для алгоритма $\mathsf{Gen}^{\mathsf{McE}(C)}$ введён один входной параметр: вес t добавляемого вектора ошибок. В алгоритме шифрования Enc вектор $\mathbf{e}$ веса t может выбираться, например, случайно из $\mathcal{E}_{n,t}$ или генерироваться детерминированно, например, в зависимости от $\mathbf{m}$ с помощью инъективного отображения из $\mathbb{F}_q^k$ в $\mathcal{E}_{n,t}$. В первом случае схему $\mathsf{McE}(C)$ будем называть схемой с вероятностным шифрованием, во втором — с детерминированным шифрованием.

	$\mathsf{Gen}^{\mathsf{McE}(C)}(t)$:	$\mathsf{Enc}(pk, \mathbf{m}; \mathbf{e})$:	$\mathsf{Dec}(sk, \mathbf{c})$:
1	$S \leftarrow \mathrm{GL}_k(\mathbb{F}_q)$	$\mathbf{c} = m\tilde{G} + e$	$\mathbf{x} \leftarrow \mathsf{Decoder}(cP^{-1})$
2	$P \leftarrow \mathcal{P}_n$	***вернуть*** $\mathbf{c}$	*если* $\mathbf{x} \neq \perp$
3	$\tilde{G} = SGP$		$\mathbf{m}' = \mathbf{x}\, G^{-1} S^{-1}$
4	$pk = (\tilde{G}, t)$		***вернуть*** $\mathbf{m}'$
5	$sk = (S, P)$		*иначе*
6	***вернуть*** (pk, sk)		***вернуть*** $\perp$

Рис. 1. Схема асимметричного шифрования $\mathsf{McE}(C) = (\mathsf{Gen}^{\mathsf{McE}(C)}, \mathsf{Enc}, \mathsf{Dec})$

Важными характеристиками систем типа Мак-Элиса являются уровень OW-стойкости, размер открытого ключа (число байт для хранения матрицы $\tilde{G}$) и время расшифрования. В случае, когда для системы шифрования типа Мак-Элиса неизвестны эффективные атаки восстановления ключа, уровень стойкости системы принято оценивать с помощью вероятности успешного декодирования по информационным совокупностям (далее — ДИС или ISD от англ. Information Set Decoding). Этот подход к декодированию линейных кодов предложен в работе [26] и в настоящее время считается наиболее эффективной атакой на шифртекст для систем шифрования типа Мак-Элиса. Несмотря на то, что известен ряд усовершенствований этого алгоритма (см., например, [27, 28]), в настоящей работе под уровнем OW-стойкости λ_{OW} системы шифрования типа Мак-Элиса на $[n, k, d]_q$-коде понимается отрицательный двоичный логарифм вероятности успеха классического алгоритма ДИС, то есть вероятности выбора (по схеме с возвращением) информационной совокупности, свободной от ошибок:

$$\lambda_{\mathsf{OW}} = -\log_2 \left(\frac{\mathrm{C}_{n-t}^k}{\mathrm{C}_n^k} \right) = \log_2 \left(\frac{\mathrm{C}_n^k}{\mathrm{C}_{n-t}^k} \right). \tag{4}$$

В табл. 1 приведены характеристики асимметричных схем шифрования типа Мак-Элиса на кодах Гоппы: $[n, k]$ — длина и размерность используемого $[n, k, d]_2$-кода Гоппы C; $t \leqslant \lfloor d-1 \rfloor/2$ — вес вектора ошибок, добавляемого в алгоритме Enc на рис. 1; λ_{OW} — уровень стойкости, вычисленный в соответствии с (4); размер ключа в мегабайтах (Мбайт) для систем Original McEliece и Classic McEliece; T — время расшифрования (в миллисекундах) информационного сообщения при использовании декодера из [3] (измерение времени выполнения операций во всех экспериментах производилось на компьютере с процессором Intel Core i7-4771 3,50 ГГц и оперативной памятью 32 Гбайт). Стоит отметить, что названия вариантов системы шифрования Original McEliece, а также параметры $[n, k]$ и t выбраны такими же, как и для системы шифрования Classic McEliece из работы [3].

Значения λ_{OW} из табл. 1 в настоящей работе рассматриваются как целевые при разработке схемы шифрования.

Таблица 1

Характеристики системы Мак-Элиса на кодах Гоппы

Система шифрования	$[n, k]$	t	λ_{OW}	Размер ключа, Original McEliece Мбайт	Размер ключа, Classic McEliece Мбайт	T
mceliece348864	$[3488, 2720]$	64	142,8	1,13	0,25	20
mceliece460896	$[4608, 3360]$	96	185,92	1,85	0,5	51
mceliece6688128	$[6688, 5024]$	128	262,28	4,01	1	98
mceliece6960119	$[6960, 5413]$	119	265,6	4,49	1	95
mceliece8192128	$[8192, 6528]$	128	302,21	6,38	1,29	120

На основе асимметричной системы шифрования $\mathsf{PKE} = (\mathsf{Gen}^{\mathsf{PKE}}, \mathsf{Enc}, \mathsf{Dec})$ типа Мак-Элиса на $[n, k, d]_q$-коде C наиболее стойким механизмом KEM с IND-CCA-стойкостью считается механизм, получаемый с помощью преобразования Фудзисаки — Окамото (FO) [18]. В частности, для всех λ_{OW} из табл. 1 система шифрования Classic McEliece обеспечивает IND-CCA-стойкость протокола KEM на основании преобразования Фудзисаки — Окамото [3]. Тройка алгоритмов преобразования $\mathsf{FO}^{\not\perp}$ [18] механизма KEM с неявным отказом (сообщение $\perp$ не возвращается) показана на рис. 2, где $\mathsf{G} : \mathbb{F}_q^k \to \mathcal{E}_{n,t}$ — инъективное отображение; $\mathsf{H} : \mathbb{F}_q^* \to \{0,1\}^m$ — криптографическая хеш-функция.

	$\mathsf{Gen}^{\mathsf{KEM}}(t)$:	$\mathsf{Encaps}(pk')$:	$\mathsf{Decaps}(sk', \mathbf{c})$:
1	$s \leftarrow \mathbb{F}_q^k$	$\mathbf{r} \leftarrow \mathbb{F}_q^k$	$\mathbf{r}' \leftarrow \mathsf{Dec}(sk, \mathbf{c})$
2	$(pk, sk) \leftarrow \mathsf{Gen}^{\mathsf{McE}(C)}(t)$	$\mathbf{c} \leftarrow \mathsf{Enc}(pk, \mathbf{r}; \mathsf{G}(\mathbf{r}))$	*если* $\mathbf{r}' \neq \perp$
3	$pk' = pk$	$\mathbf{K} \leftarrow \mathsf{H}(\mathbf{r} \parallel \mathbf{c})$	$\mathbf{c}' \leftarrow \mathsf{Enc}(pk, \mathbf{r}'; \mathsf{G}(\mathbf{r}'))$
4	$sk' = (sk, s)$	***вернуть*** $(\mathbf{K}, \mathbf{c})$	*если* $\mathbf{c} = \mathbf{c}'$
5	***вернуть*** (pk', sk')		***вернуть*** $\mathsf{H}(\mathbf{r}' \parallel \mathbf{c})$
6			***вернуть*** $\mathsf{H}(s \parallel \mathbf{c})$

Рис. 2. Механизм $\mathsf{KEM} = (\mathsf{Gen}^{\mathsf{KEM}}, \mathsf{Encaps}, \mathsf{Decaps})$, полученный применением преобразования $\mathsf{FO}^{\not\perp}$ на основе схемы асимметричного шифрования $\mathsf{McE}(C)$ с детерминированным правилом шифрования

3. D-коды и их эффективное декодирование

В работе [14] предлагается система шифрования типа Мак-Элиса, где кодом C является бинарный D-код. Однако в силу небольшого отношения кодового расстояния D-кода к длине кода для обеспечения высокого уровня стойкости λ_{OW} при использовании декодеров, работающих в пределах половины кодового расстояния, необходимо использовать открытые ключи большого размера. В частности, в [14] показано, что для обеспечения стойкости, сопоставимой со стойкостью Classic McEliece с набором параметров mceliece348864, необходимо использовать открытый ключ размера 154,85 Мбайт. Большие размеры ключей приводят не только к сопутствующим проблемам их хранения и/или передачи, но и к росту вычислительной сложности декодера и/или росту его сложности по используемой памяти. Например, конструкция мажоритарного декодера, предложенная в [15], представляет собой граф, число узлов которого быстро растёт с ростом длины кода.

В [14] отмечено, что уменьшения размера ключа при сохранении стойкости можно потенциально добиться путём применения декодеров кода Рида — Маллера, позволяющих исправлять многие шаблоны ошибок веса, превышающего половину кодового расстояния. Далее даётся определение D-кодов, а также строятся декодеры упомянутого типа для D-кодов, построенных на основе двоичных кодов Рида — Маллера.

3.1. Коды Рида — Маллера и их декодирование

Среди линейных кодов важным является класс двоичных $[n, k, d]_2$-кодов Рида — Маллера $\mathrm{RM}(r, m)$, где $n = 2^m$; $k = \sum\limits_{i=0}^{r} \mathrm{C}_r^i$; $d = 2^{m-r}$ [16]. В частности, известно, что эти коды достигают ёмкости канала со стираниями и симметричных каналов без памяти [29, 30]. Для кодов Рида — Маллера существуют разные подходы к декодированию [16], которые условно можно разделить на гарантированное и вероятностное декодирование. В первом случае декодеры нацелены на гарантированное исправление любых векторов ошибок, имеющих вес в пределах половины кодового расстояния. Во втором случае декодеры направлены на исправление векторов ошибок большого веса, при этом они обладают ненулевой вероятностью ошибочного декодирования (DFR, Decoding Failure Rate). Примерами гарантированных декодеров являются классический декодер Рида [31] и мажоритарный декодер Мэсси [32, 33]. Одним из первых вероятностных декодеров является декодер Сидельникова — Першакова [34]; одним из последних — декодер Йе — Аббе, который, согласно [17], обладает лучшей корректирующей способностью по сравнению с декодером Сидельникова — Першакова. Отметим, что вероятностные декодеры, как правило, вычислительно сложнее, чем эффективные гарантированные.

3.2. D-коды

Подробное описание D-кодов, методы и алгоритмы их декодирования, а также их криптографические свойства представлены в работах [13–15, 35]. D-код — это специальная сумма тензорных произведений кодов, которые назовём *базовыми*. Напомним, что тензорным произведением кодов C_1 и C_2 является код $C_1 \otimes C_2$, порождающая матрица G которого может быть представлена как тензорное произведение порождающих матриц кодов C_1 и C_2. Под тензорным произведением $A \otimes B$ матриц $A = (a_{i,j})$ и B размера $k_1 \times n_1$ и $k_2 \times n_2$ соответственно будем понимать $(k_1 k_2 \times n_1 n_2)$-матрицу вида

$$\begin{pmatrix} a_{1,1}B & \cdots & a_{1,n_1}B \\ \vdots & \ddots & \vdots \\ a_{k_1,1}B & \cdots & a_{k_1,n_1}B \end{pmatrix}.$$

D-коды, где базовыми являются коды Рида — Маллера, определяются следующим образом. Пусть $m_1, m_2 \in \mathbb{N}$; $(k_i)_{i=1}^s$ — возрастающая последовательность, $k_1 \geqslant 0$, $k_s \leqslant m_1$; $(\ell_i)_{i=1}^s$ — убывающая последовательность, $\ell_1 \leqslant m_1$, $\ell_s \geqslant 0$. Тогда D-код C длины $n = 2^{m_1+m_2}$ и размерности k представим в виде

$$C = \sum_{i=1}^{s} \left(\mathrm{RM}(k_i, m_1) \otimes \mathrm{RM}(\ell_i, m_2)\right), \tag{5}$$

где под суммой кодов понимается сумма подпространств. Далее рассмотрим только D-коды на основе кодов Рида — Маллера (коды вида (5)).

3.3. Декодирование D-кодов

Рассмотрим задачу декодирования D-кода C с характеристиками $[n, k, d]_2$. Пусть G — порождающая матрица кода C, $\mathbf{m} \in \mathbb{F}_2^k$, $\mathbf{m}G + \mathbf{e} = \mathbf{z}$, $\mathrm{wt}(\mathbf{e}) = t \in \mathbb{N}$. Вопросам построения и эффективной реализации одного декодера для D-кодов посвящены работы [15, 35]. В основе декодирования лежит мажоритарный подход, предложенный Дж. Мэсси [32], который, применительно к D-кодам, позволяет исправлять все шаблоны ошибок веса не более половины кодового расстояния: $t \leqslant \lfloor (d-1)/2 \rfloor$. Поэтому этот декодер относится к классу гарантированных. Для удобства гарантированный декодер, описанный в [15], обозначим $\mathsf{GraphDecoder}$.

Особая структура D-кодов и наличие вероятностных декодеров для кодов Рида — Маллера позволяют построить вероятностный декодер и для D-кодов. Далее строятся два вероятностных декодера: с использованием декодирования по информационным совокупностям и на основе декодирования кодов-произведений, в которых кодовое слово может быть представлено в виде матрицы. Декодер D-кода, когда не важен класс декодера (гарантированный или вероятностный), будем обозначать $\mathsf{DDecoder}$.

Для описания вероятностных декодеров потребуется полиномиальный алгоритм $\mathsf{ExitCond}(G, \mathbf{m}', \mathbf{z}, \mathsf{op}_1, \mathsf{op}_2)$ проверки правильности информационного вектора $\mathbf{m}' \in \mathbb{F}_2^k$, получаемого в ходе декодирования алгоритмом $\mathsf{DDecoder}$ зашумлённого вектора $\mathbf{z} \in \mathbb{F}_2^n$ D-кода, заданного порождающей матрицей G. Этот алгоритм возвращает 1, если $\mathbf{m}'$ соответствует $\mathbf{z}$, и 0 в противном случае. Алгоритм $\mathsf{ExitCond}$ используется как параметр алгоритма $\mathsf{DDecoder}$ для D-кода. Заметим, что вектор $\mathbf{m}'$ вычисляется внутри алгоритма декодирования, а так как перед проверкой к вектору $\mathbf{m}'$ в ряде случаев должно быть применено некоторое преобразование (такими преобразованиями могут быть, например, умножение вектора на матрицу, отбрасывание фиксированных координат и т. п.), четвёртым параметром алгоритма $\mathsf{ExitCond}$ является преобразование op_1 вектора $\mathbf{m}'$ перед проверкой. Аналогичное преобразование, соответствующее параметру op_2, может быть применено перед проверкой к вектору $\mathbf{z}$. Примеры op_1 и op_2 содержатся в п. 4.2.

Отметим, что проверка только на основе веса может быть ложно положительной при $t > \lfloor (d-1)/2 \rfloor$. Снизить вероятность такого события можно, например, в случае, когда известно, что добавляемая ошибка генерируется с помощью некоторого детерминированного инъективного алгоритма $\mathsf{G} : \mathbb{F}_2^k \to \mathcal{E}_{n,t}$. На рис. 3 приведены примеры двух алгоритмов проверки правильности информационного вектора: на основе только веса ошибки и на основе значения инъективного отображения G.

	$\mathsf{ExitCond}_{\mathsf{wt}}(G, \mathbf{z}, \mathbf{m}', \mathsf{op}_1, \mathsf{op}_2)$:	$\mathsf{ExitCond}_{\mathsf{G}}(G, \mathbf{z}, \mathbf{m}', \mathsf{op}_1, \mathsf{op}_2)$:
1	$\mathbf{e} = \mathbf{z} - \mathsf{op}_1(\mathbf{m}')G$	$\mathbf{e} = \mathsf{op}_2(\mathbf{z}) - \mathsf{op}_1(\mathbf{m}')G$
2	*если* $\mathrm{wt}(\mathbf{e}) = t$	*если* $\mathrm{wt}(\mathbf{e}) = t$ *и* $\mathsf{G}(\mathsf{op}_1(\mathbf{m}')) = \mathbf{e}$
3	***вернуть*** 1	***вернуть*** 1
4	*иначе*	*иначе*
5	***вернуть*** 0	***вернуть*** 0

Рис. 3. Алгоритмы проверки правильности информационного вектора: $\mathsf{ExitCond}_{\mathrm{wt}}$ — на основе только веса вектора ошибок; $\mathsf{ExitCond}_{\mathsf{G}}$ — на основе веса и значения инъективного отображения G

Введём обозначение для вероятностей ξ_{G}, ξ_{wt} ложно положительной проверки правильности информационного вектора для алгоритмов $\mathsf{ExitCond}_{\mathsf{G}}$, $\mathsf{ExitCond}_{\mathsf{wt}}$:

$$\xi_{\mathrm{A}} = \Pr\left[\mathsf{ExitCond}_{\mathsf{A}}(G, \mathbf{m}', \mathbf{z}, \mathsf{op}_1, \mathsf{op}_2) = 1 \,|\, \mathbf{m}' \neq \mathbf{m}\right], \quad \mathsf{A} \in \{\mathsf{wt}, \mathsf{G}\}.$$

Стоит отметить, что во всех проведённых экспериментах проверка только на основе веса не дала ложно положительных ответов, хотя теоретические оценки вероятности ложно положительного ответа в этом случае отсутствуют. В то же время проверка на основе инъективного отображения G даёт возможность оценить эту вероятность. Действительно, $\mathsf{ExitCond}_{\mathsf{G}}(G, \mathbf{m}', \mathbf{z}, \mathsf{op}_1, \mathsf{op}_2) = 1$ при $\mathbf{m}' \neq \mathbf{m}$ означает, что

$$\mathsf{op}_2(\mathbf{z}) = \mathsf{op}_1(\mathbf{m})G + \mathsf{G}(\mathsf{op}_1(\mathbf{m})) = \mathsf{op}_1(\mathbf{m}')G + \mathsf{G}(\mathsf{op}_1(\mathbf{m}')).$$

Отсюда получаем, что $\mathsf{G}(\mathsf{op}_1(\mathbf{m}')) - \mathsf{G}(\mathsf{op}_1(\mathbf{m})) \in C$. Если при $n - k > k$ отображение G можно выбрать так, что для разных $\mathbf{m}$ и $\mathbf{m}'$ векторы $\mathsf{G}(\mathsf{op}_1(\mathbf{m}))$ и $\mathsf{G}(\mathsf{op}_1(\mathbf{m}'))$ принадлежат разным смежным классам фактор-множества $\mathbb{F}_2^n/C$, то $\xi_{\mathsf{G}} = 0$. Построение такого отображения составляет отдельную задачу, тем не менее представляется, что можно построить такое отображение G, для которого вероятностью ξ_{G} можно пренебречь.

Проверки правильности декодирования нередко используются для раннего выхода в итерационных алгоритмах декодирования, используемых при защите от помех. Однако в таких алгоритмах обычно выполняется проверка того, что декодированный вектор является кодовым, но проверка правильности информационного вектора не выполняется. При синтезе схем KEM, как отмечено в п. 1, важную роль играет значение вероятности ошибочной декапсуляции δ (см. (2)). Поэтому при выборе конкретного алгоритма проверки следует исходить из допустимой вероятности этой ошибки. Таким условием для механизмов KEM является условие (3): для IND-CCA-стойкости уровня λ_{KEM} механизма KEM необходимо, чтобы $\xi_{\mathsf{A}} \leqslant 2^{-\lambda_{\mathsf{KEM}}}$, а для уровня отказоустойчивости γ механизма KEM с IND-CPA-стойкостью необходимо выполнение неравенства $\xi_{\mathsf{A}} \leqslant 10^{-\gamma}$. Далее будем полагать, что алгоритм проверки выбирается, исходя из этих условий. Отметим, что проверка правильности информационного вектора может проводиться также на основе контрольной суммы, являющейся частью информационного сообщения. Однако в случае, когда такая проверка применяется в алгоритме декодирования, используемом при синтезе KEM, необходимо обеспечить соблюдение условия (1). Это обусловлено тем, что из OW-стойкости схемы PKE не вытекает такая же OW-стойкость схемы PKE, в которой часть сообщения отводится под контрольную сумму. В настоящей работе проверки на основе контрольной суммы не рассматриваются.

Декодер на основе декодирования базового кода и ДИС

Пусть $n_i = 2^{m_i}$, $i = 1, 2$; C — код вида (5). Из представления (5) следует, что

$$C \subseteq \mathbb{F}_2^{n_1} \otimes \mathrm{RM}(\ell_1, m_2) = \underbrace{\mathrm{RM}(\ell_1, m_2) \times \cdots \times \mathrm{RM}(\ell_1, m_2)}_{n_1}, \tag{6}$$

где $A \times B$ — внешняя прямая сумма подпространств A и B. Следовательно, кодовое слово кода C представляет собой конкатенацию кодовых слов кода $\mathrm{RM}(\ell_1, m_2)$. Поэтому декодирование кода C может заключаться в применении некоторого известного декодера $\mathsf{DecRow} : \{0,1\}^{2^{m_2}} \to \mathrm{RM}(\ell_1, m_2) \cup \{\perp\}$ кода $\mathrm{RM}(\ell_1, m_2)$ к каждому из n_1 блоков длины n_2 декодируемого вектора $\mathbf{z} = \mathbf{z}_1 \parallel \cdots \parallel \mathbf{z}_{n_1}$. Такой способ декодирования назовём $\mathsf{BlockDecoder}_{n_1,n_2}$:

$$\mathsf{BlockDecoder}_{n_1,n_2}(\mathbf{z}, \mathsf{DecRow}) = \mathsf{DecRow}(\mathbf{z}_1) \parallel \cdots \parallel \mathsf{DecRow}(\mathbf{z}_{n_1}).$$

Пара нижних индексов n_1, n_2 в названии декодера означает, что перед применением декодера DecRow выполняется разбиение входного вектора $\mathbf{z}$ на n_1 блоков длиной n_2 каждый.

В качестве DecRow в $\mathsf{BlockDecoder}_{n_1,n_2}$ могут использоваться как гарантированные, так и вероятностные алгоритмы декодирования, например декодеры Рида, Мэсси, Сидельникова — Першакова и Йе — Аббе. При этом в обоих случаях алгоритм $\mathsf{BlockDecoder}_{n_1,n_2}$ может вернуть неправильный результат, то есть некоторое количество блоков $\mathbf{z}_i$ зашумлённого кодового вектора $\mathbf{z}$ будет декодировано неправильно. Если блок был декодирован правильно, то назовём его *хорошим*, в противном случае — *плохим*. Отметим, что в худшем случае после работы алгоритма $\mathsf{BlockDecoder}_{n_1,n_2}$ неизвестно, какие блоки хорошие, а какие — плохие.

Чтобы повысить вероятность правильного декодирования, целесообразно после блочного декодирования воспользоваться декодированием по информационным совокупностям. Такой подход — блочное декодирование и применение ДИС — использован в работе [14] при разработке комбинированной атаки на шифртекст. Здесь этот подход использован для декодирования и представлен в алгоритме 1. Напомним, что информационной совокупностью линейного $[n, k]_q$-кода называется такое множество $\tau \subset \{1, \ldots, n\}$ мощности k, что столбцы любой порождающей матрицы кода с номерами из τ линейно независимы. Обычно при декодировании по информационным совокупностям (см., например, оригинальную работу [26]) случайным образом выполняется поиск такой информационной совокупности τ, чтобы номера ненулевых координат вектора ошибки не принадлежали τ. В [14] и алгоритме 1 выполняется поиск номеров хороших блоков, на основе которых может быть построена информационная совокупность.

Для описания алгоритма декодирования понадобится блочное представление порождающей матрицы D-кода. Из (5) вытекает, что любая порождающая $(k \times n)$-матрица G этого кода представима в виде

$$G = [G_1 | G_2 | \ldots | G_{n_1}], \tag{7}$$

где строки $(k \times n_2)$-матрицы (блока) G_i — кодовые слова кода $\mathrm{RM}(\ell_1, m_2)$, $i \in [\![1, n_1]\!]$. Здесь и далее запись $[\![a, b]\!]$ означает диапазон целых чисел от a до b включительно. Если M — матрица, состоящая из r столбцов, $\omega \subseteq [\![1, r]\!]$, то матрицу, состоящую из столбцов с номерами из ω, будем обозначать $\Pi_\omega(M)$. Через τ_0 обозначим заранее

известную фиксированную информационную совокупность кода C, $M_0 = \Pi_{\tau_0}(G)$. Параметрами алгоритма 1, помимо декодируемого слова $\mathbf{z}$, декодера DecRow, являются K — количество выбираемых блоков для применения ДИС, $I_{\max}$ — максимальное количество итераций ДИС (под одной итерацией понимаются шаги с 10 по 18) и $\mathsf{ExitCond}$ — условие раннего выхода из ДИС (примеры алгоритма $\mathsf{ExitCond}$ приведены на рис. 3). Параметр K выбирается исходя из особенностей кода, в частности необходимо, чтобы вероятность того, что выбранные K блоков составляют матрицу ранга k, была непренебрежимой. В данной работе в качестве нижней границы такой вероятности для выбора параметра K использовано значение 0,01. Параметр $I_{\max}$ выбирается исходя из вычислительных возможностей и допустимой вероятности ошибочного декодирования.

Алгоритм 1. $\mathsf{ISDDecoder}$ — декодер D-кода с применением ДИС

Вход: $\mathbf{z}; \tau_0, G, \mathsf{ExitCond}, \mathsf{op}_1, \mathsf{op}_2, \mathsf{DecRow}, K, I_{\max}$.

1: $result = \perp$.
2: $\mathbf{z}' = (\mathbf{z}'_1 \parallel \ldots \parallel \mathbf{z}'_{n_1}) = \mathsf{BlockDecoder}_{n_1,n_2}(\mathbf{z}, \mathsf{DecRow})$.
3: $B = \{j \in [\![1, n_1]\!] : \mathbf{z}'_j = \perp\}$.
4: **Если** $\left(\bigcup\limits_{j \in B} [\![(j-1)n_2 + 1, jn_2]\!]\right) \bigcap \tau_0 = \varnothing$, **то**
5: $\quad \mathbf{m}' = \Pi_{\tau_0}(\mathbf{z}')M_0^{-1}$. *// попытка декодирования без ДИС*
6: **Если** $\mathsf{ExitCond}(G, \mathbf{m}', \mathbf{z}, \mathsf{op}_1, \mathsf{op}_2) = 1$, **то**
7: $\quad result = \mathbf{m}'$
8: **иначе**
9: $\quad i = I_{\max}$.
10: $\quad$ **Пока** $i > 0$ и $result = \perp$:
// попытка декодирования с ДИС
11: $\quad\quad$ выбрать случайно $W \subset [\![1, n_1]\!] \setminus B$, $|W| = K$;
12: $\quad\quad G' = \Pi_\omega(G)$, где $\omega = \bigcup\limits_{j \in W} [\![(j-1)n_2 + 1, jn_2]\!]$.
13: $\quad\quad$ **Если** $\operatorname{rank}(G') = k$, **то**
14: $\quad\quad\quad$ найти такое $\tau \subset \omega$, что $|\tau| = k$, $\operatorname{rank}(\Pi_\tau(G)) = k$;
15: $\quad\quad\quad \mathbf{m}' = \Pi_\tau(\mathbf{z}')(\Pi_\tau(G))^{-1}$.
16: $\quad\quad\quad$ **Если** $\mathsf{ExitCond}(G, \mathbf{m}', \mathbf{z}, \mathsf{op}_1, \mathsf{op}_2) = 1$, **то**
17: $\quad\quad\quad\quad result = \mathbf{m}'$.
18: $\quad\quad i = i - 1$.
19: **Вернуть** $result$.

Оценим вероятность ошибочного декодирования для алгоритма 1. В случае $i \in \mathbb{N}$ итераций ДИС эту вероятность можно вычислить по следующей формуле:

$$\mathrm{DFR}_1(i) = 1 - \left(P_{\mathrm{dec}}^{(0)} + \sum_{j=1}^{n_1} \left(1 - (1 - P_1^{(j)} P_2)^i\right) P_{\mathrm{dec}}^{(j)}\right). \tag{8}$$

Здесь $P_1^{(j)} = \mathrm{C}_{n_1-j}^K / \mathrm{C}_{n_1}^K$ — вероятность выбора на шаге 11 K хороших блоков из n_1 блоков при наличии j плохих блоков; P_2 — вероятность того, что выбранные K блоков образуют матрицу ранга k (шаг 13); $P_{\mathrm{dec}}^{(j)}$ — вероятность появления j плохих блоков в зашумлённом кодовом векторе. Вероятность $P_{\mathrm{dec}}^{(j)}$ рассчитывается по формуле

$$P_{\mathrm{dec}}^{(j)} = \mathrm{C}_{n_1}^j P_{\mathrm{bad}}^j (1 - P_{\mathrm{bad}})^{n_1 - j}, \tag{9}$$

где P_{bad} — вероятность того, что некоторый фиксированный блок станет плохим после применения на шаге 2 алгоритма $\mathsf{BlockDecoder}_{n_1,n_2}$. Заметим, что P_{bad} зависит от кода, декодера DecRow и числа ошибок t. В настоящей работе P_{bad} оценивается экспериментально. Поясним формулу (8). Значение $\text{DFR}_1(i)$ получается вычитанием из единицы вероятности правильного декодирования за i итераций ДИС, которая, в свою очередь, рассчитывается как сумма двух вероятностей. Первая — вероятность $P_{\text{dec}}^{(0)}$ того, что на шаге 2 после работы алгоритма $\mathsf{BlockDecoder}_{n_1,n_2}$ будет получено правильное кодовое слово. Вторая — вероятность правильного декодирования после i итераций ДИС, в которой учитываются случаи появления j плохих блоков, где $j = 1, \ldots, n_1$. При этом $(1 - P_1^{(j)} P_2)^i$ — вероятность того, что за i итераций не будет найдено набора из K блоков матрицы G в представлении (7), соответствующих хорошим блокам вектора $\mathbf{z}$ и составляющих матрицу ранга k.

Оценим вычислительную эффективность алгоритма 1. Пусть T_{block} — среднее время, необходимое для выполнения алгоритма $\mathsf{BlockDecoder}_{n_1,n_2}$, а T_{ISD} — среднее время, необходимое для выполнения одной итерации ДИС, тогда среднее время T_1, необходимое для правильного декодирования алгоритмом 1, можно вычислить по формуле

$$T_1 = T_{\text{block}} + \sum_{i=1}^{I_{\max}} i T_{\text{ISD}} \sum_{j=0}^{n_1} \left(1 - P_1^{(j)} P_2\right)^{i-1} P_1^{(j)} P_2 P_{\text{dec}}^{(j)}. \tag{10}$$

Поясним формулу (10). Для нахождения T_1 необходимо к времени T_{block} добавить среднее время работы ДИС, которое зависит от среднего количества итераций. Поэтому во втором слагаемом в (10) вычисляется математическое ожидание времени работы ДИС, где для каждого слагаемого в сумме по количеству необходимых итераций i вычисляется вероятность того, что ДИС закончит свою работу после i-й итерации. Благодаря сумме по j учитываются все возможные варианты количества плохих блоков и вероятности их появления.

Декодер на основе декодирования кодов-произведений

Из вида (5) следует, что

$$C \subseteq (\mathbb{F}_2^{n_1} \otimes \text{RM}(\ell_1, m_2)) \cap (\text{RM}(k_s, m_1) \otimes \mathbb{F}_2^{n_2}),\ n_i = 2^{m_i},\ i = 1, 2.$$

Следовательно, алгоритм декодирования зашумлённого кодового слова $\mathbf{z}$ кода C может быть основан на способе декодирования кодов-произведений: 1) представить $\mathbf{z}$ в виде $(n_1 \times n_2)$-матрицы, строки которой — зашумлённые кодовые слова кода $\text{RM}(\ell_1, m_2)$, столбцы — зашумлённые кодовые слова кода $\text{RM}(k_s, m_1)$; 2) применить декодер по строкам, а затем к результату декодирования применить декодер по столбцам. Декодеры для строк и столбцов могут быть разными. Например, к строкам может быть применён декодер из работы [17] для исправления большого числа ошибок, а к столбцам — вычислительно более простой декодер, например мажоритарный, так как ожидается, что строчный декодер исправит «почти все» ошибки и для декодера по столбцам достаточно простого алгоритма. Декодер по строкам может вернуть символ $\perp$, в этом случае соответствующий блок может быть заменён на вектор $(\star, \ldots, \star)$ длины n_2, где $\star$ — символ стирания. Такая замена позволит, например, использовать возможность исправления стираний при декодировании по столбцам.

Заметим, что всегда можно переставить координаты вектора $\mathbf{z}$ так, что вектор с переставленными координатами будет представлять конкатенацию n_2 зашумлённых кодовых слов кода $\text{RM}(k_s, m_1)$ длины n_1 каждый. Такая перестановка зависит только от параметров n_1 и n_2; обозначим её π. Следовательно, при декодировании вектора $\mathbf{z}$ необязательно представлять его в виде матрицы: при декодировании по строкам

к вектору $\mathbf{z}$ применяется алгоритм $\mathsf{BlockDecoder}_{n_1,n_2}$ с декодером по строкам DecRow в качестве второго аргумента, а при декодировании по столбцам к вектору $\pi(\mathbf{z}')$ применяется алгоритм $\mathsf{BlockDecoder}_{n_2,n_1}$, в котором вторым аргументом является декодер по столбцам DecCol. Здесь $\mathbf{z}'$ обозначает результат после декодирования по строкам. Описанный способ декодирования приведён в алгоритме 2. В качестве $\mathsf{ExitCond}$ могут быть использованы алгоритмы, представленные на рис. 3.

Если предполагать, что декодер по столбцам является гарантированным, то есть исправляет ошибки в пределах половины кодового расстояния, то вероятность ошибочного декодирования для декодера, представленного в алгоритме 2, можно вычислить как вероятность появления плохих блоков после работы строчного декодера в количестве, превышающем половину кодового расстояния кода $\mathrm{RM}(k_s, m_1)$. То есть

$$\mathrm{DFR}_2 = \sum_{j=\lfloor (d_2-1)/2 \rfloor + 1}^{n_1} P_{\mathrm{dec}}^{(j)}, \tag{11}$$

где d_2 — минимальное кодовое расстояние кода $\mathrm{RM}(k_s, m_1)$; $P_{\mathrm{dec}}^{(j)}$ имеет вид (9).

Алгоритм 2. $\mathsf{MatrixDecoder}$ — декодер D-кода без применения ДИС

Вход: $\mathbf{z}; \tau_0, G, \mathsf{ExitCond}, \mathsf{op}_1, \mathsf{op}_2, \mathsf{DecRow}, \mathsf{DecCol}$.

1: $result = \perp$.
2: $\mathbf{z}' = (\mathbf{z}'_1 \parallel \ldots \parallel \mathbf{z}'_{n_1}) = \mathsf{BlockDecoder}_{n_1,n_2}(\mathbf{z}, \mathsf{DecRow})$.
3: $B_r = \{j \in [\![1, n_1]\!] : \mathbf{z}'_j = \perp\}$.
4: **Если** $\big(\bigcup\limits_{j \in B_r} [\![(j-1)n_2 + 1, jn_2]\!] \big) \cap \tau_0 = \varnothing$, **то**
5: $\quad \mathbf{m}' = \Pi_{\tau_0}(\mathbf{z}')M_0^{-1}$. *// первая попытка декодирования*
6: $\quad$ **Если** $\mathsf{ExitCond}(G, \mathbf{m}', \mathbf{z}, \mathsf{op}_1, \mathsf{op}_2) = 1$, **то**
7: $\quad\quad result = \mathbf{m}'$,
8: **иначе**
9: $\quad$ заменить в $\mathbf{z}'$ блоки с номерами из B_r на вектор $(\star, \ldots, \star)$ длины n_2;
10: $\quad \mathbf{z}'' = (\mathbf{z}''_1 \parallel \ldots \parallel \mathbf{z}''_{n_2}) = \mathsf{BlockDecoder}_{n_2,n_1}(\pi(\mathbf{z}'), \mathsf{DecCol})$;
11: $\quad B_c = \{j \in [\![1, n_2]\!] : \mathbf{z}''_j = \perp\}$.
12: $\quad$ **Если** $B_c = \varnothing$, **то**
13: $\quad\quad \mathbf{m}' = \Pi_{\tau_0}(\pi^{-1}(\mathbf{z}''))M_0^{-1}$. *// вторая попытка декодирования*
14: $\quad\quad$ **Если** $\mathsf{ExitCond}(G, \mathbf{m}', \mathbf{z}, \mathsf{op}_1, \mathsf{op}_2) = 1$, **то**
15: $\quad\quad\quad result = \mathbf{m}'$.
16: **Вернуть** $result$

Алгоритм 2, в отличие от алгоритма 1, имеет постоянное время выполнения. Оно складывается из времени работы $\mathsf{BlockDecoder}_{n_1,n_2}$ с декодером DecRow на шаге 2 (T_{DecRow}) и времени работы $\mathsf{BlockDecoder}_{n_2,n_1}$ с декодером DecCol на шаге 10 (T_{DecCol}), то есть

$$T_2 = T_{\mathsf{DecRow}} + T_{\mathsf{DecCol}}. \tag{12}$$

4. Система шифрования типа Мак-Элиса на D-кодах и её характеристики

4.1. Система шифрования типа Мак-Элиса на D-кодах

Схему типа Мак-Элиса на основе D-кода C вида (5) с характеристиками $[n, k, d]_2$ и декодером $\mathsf{DDecoder} \in \{\mathsf{GraphDecoder}, \mathsf{ISDDecoder}, \mathsf{MatrixDecoder}\}$ будем обозначать $\mathsf{McE}(D, \mathsf{DDecoder})$; тройка соответствующих алгоритмов приведена на рис. 4.

	$\mathsf{Gen}^{\mathsf{McE}(D)}(t)$:	$\mathsf{Enc}(pk, \mathbf{m}; \mathbf{e})$:	$\mathsf{Dec}^{\mathsf{DDecoder}}(sk, \mathbf{c})$:
1	$S \leftarrow \mathrm{GL}_k(\mathbb{F}_2)$	$\mathbf{c} = \mathbf{m}\tilde{G} + \mathbf{e}$	$\mathbf{x} \leftarrow \mathsf{DDecoder}(\mathbf{c}P^{-1})$
2	$P \leftarrow \mathcal{P}_n$	***вернуть*** $\mathbf{c}$	*если* $\mathbf{x} \neq \perp$
3	$pk = (SGP, t)$		$\mathbf{m}' = \mathbf{x}G^{-1}S^{-1}$
4	$sk = (S, P)$		***вернуть*** $\mathbf{m}$
5	***вернуть*** (pk, sk)		*иначе*
6			***вернуть*** $\perp$

Рис. 4. Схема $\mathsf{McE}(D, \mathsf{DDecoder})$, $\mathsf{DDecoder} \in \{\mathsf{GraphDecoder}, \mathsf{ISDDecoder}, \mathsf{MatrixDecoder}\}$

Формально схема на рис. 4 отличается от схемы на рис. 1 только конкретизацией поля $\mathbb{F}_2$, над которым рассматриваются D-коды, а также указанием на использование декодера $\mathsf{DDecoder}$ для D-кодов в правиле расшифрования. Вес t векторов ошибок, добавляемых в правиле шифрования Enc, определяется из условия

$$t = \begin{cases} \lfloor (d-1)/2 \rfloor, & \mathsf{DDecoder} = \mathsf{GraphDecoder}, \\ t' > \lfloor (d-1)/2 \rfloor, & \mathsf{DDecoder} \in \{\mathsf{ISDDecoder}, \mathsf{MatrixDecoder}\}, \end{cases} \tag{13}$$

где t' выбирается, исходя из кода, декодера, вероятности ошибочного расшифрования и требуемого уровня стойкости λ_{OW} соответствующей системы шифрования. Более подробно выбор t' для некоторых систем $\mathsf{McE}(D, \mathsf{DDecoder})$ обсуждается в следующем пункте. В правиле расшифрования Dec на рис. 4 алгоритм $\mathsf{DDecoder}$ вызывается с одним параметром — значением декодируемого вектора, при этом конкретные реализации декодера $\mathsf{DDecoder}$ могут принимать и дополнительные параметры. В частности, алгоритмы декодирования $\mathsf{ISDDecoder}$ и $\mathsf{MatrixDecoder}$ в качестве дополнительных принимают такие параметры, как информационная совокупность τ_0, порождающая матрица G, условие $\mathsf{ExitCond}$, алгоритмы op_1 и op_2 преобразования информационного и декодируемого векторов перед проверкой условия декодирования и т. п. Можно считать, что при реализации схемы шифрования значения таких параметров фиксированы, поэтому они могут рассматриваться как часть реализации декодера, а не как его параметры.

4.2. Характеристики схемы $\mathsf{McE}(D, \mathsf{DDecoder})$

Обсудим выбор характеристик схемы $\mathsf{McE}(D, \mathsf{DDecoder})$, обеспечивающих заданный уровень OW-стойкости схемы, в частности целевые значения λ_{OW} из табл. 1.

В [13] найдены условия на D-коды, при которых ключи системы шифрования типа Мак-Элиса на этих кодах являются слабыми: построена структурная атака, позволяющая найти часть закрытого ключа. Эта атака не зависит от декодера, поэтому применима к слабым ключам всех вариантов схемы $\mathsf{McE}(D, \mathsf{DDecoder})$. В работе [14] на основе структурной атаки из [13] разработана комбинированная атака на шифртекст схемы $\mathsf{McE}(D, \mathsf{DDecoder})$. Вероятность успеха комбинированной атаки оказывается достаточно большой, что численно продемонстрировано в случае слабых ключей схемы $\mathsf{McE}(D, \mathsf{GraphDecoder})$. Несмотря на отсутствие соответствующих численных подтверждений для схем $\mathsf{McE}(D, \mathsf{ISDDecoder})$ и $\mathsf{McE}(D, \mathsf{MatrixDecoder})$, естественно предположить нецелесообразность применения слабых ключей и в этих реализациях. Поэтому параметры $\mathsf{McE}(D, \mathsf{DDecoder})$ следует определять, с одной стороны, исходя из обеспечения стойкости к структурной атаке, построенной в [13, 14], а с другой — из требования обеспечения заданного уровня OW-стойкости λ_{OW} (4).

В [14] исследованы характеристики схемы шифрования $\mathsf{McE}(D, \mathsf{GraphDecoder})$, где показано, что эта схема достигает высокого уровня стойкости ($\lambda_{\mathsf{OW}} \geqslant 128$) при большом размере открытого ключа (более 154 Мбайт). Целью настоящей работы является уменьшение размера ключа при сохранении уровня OW-стойкости за счёт использования вероятностных декодеров, в частности декодеров ISDDecoder и MatrixDecoder. В качестве DecRow в обоих алгоритмах используется декодер Йе — Аббе [17], корректирующая способность которого выше, чем у декодера Сидельникова — Першакова. Декодер Йе — Аббе используется также в качестве алгоритма DecCol в MatrixDecoder. Для применения условия проверки ExitCond (см. рис. 3), которое используется в алгоритмах декодирования ISDDecoder и MatrixDecoder, при расшифровании в качестве алгоритма op_1 будем использовать алгоритм $\mathsf{mul}_{S^{-1}}$, реализующий умножение вектора на матрицу S^{-1}, так как при проверке необходимо «снять» преобразование информационного вектора, полученное с помощью закрытого ключа S, а в качестве алгоритма op_2 — алгоритм mul_P, реализующий умножение вектора на матрицу P.

Из условия (13) на вес вектора ошибки в правиле шифрования Enc вытекает, что в (9) $P_{\text{bad}} \neq 0$, поэтому декодеры ISDDecoder и MatrixDecoder характеризуются ненулевой вероятностью ошибочного декодирования (см. (8) и (11) соответственно). Из (8) вытекает, что $\lim\limits_{i\to\infty} \mathrm{DFR}_1(i) = 0$, то есть при увеличении числа итераций ДИС в ISDDecoder при фиксированном числе ошибок t вероятность ошибочного декодирования можно сделать сколь угодно малой. Однако в этом случае возрастает время декодирования и, как следствие, время расшифрования. Поэтому количество итераций ДИС ограничено сверху числом $I_{\max}$, которое зависит от максимально допустимого времени расшифрования и среднего времени выполнения одной итерации ДИС, которые, в свою очередь, зависят от прикладной задачи и реализации декодера соответственно. В настоящей работе при исследовании характеристик декодера ISDDecoder значение $I_{\max}$ не превышает 10^6.

Для декодера MatrDecoder, как следует из (11), вероятность ошибочного декодирования зависит только от вероятности P_{bad}, то есть от числа добавляемых ошибок при шифровании, поэтому алгоритмически снизить вероятность ошибочного декодирования не удаётся. Однако, как отмечено выше, в отличие от ISDDecoder декодер MatrDecoder является декодером с постоянным временем декодирования. Это может быть преимуществом MatrDecoder, так как в случае, когда для криптографического протокола актуальны атаки на кодовую систему шифрования по побочным каналам, в частности атаки по времени декодирования, предпочтение отдается декодерам с постоянным временем декодирования.

Для удобства при фиксированном D-коде C, числе ошибок t и числе K хороших блоков минимальное число итераций ДИС, обеспечивающее вероятность ошибочного декодирования (8) в алгоритме ISDDecoder не более $10^{-\gamma}$, обозначим I_γ:

$$I_\gamma = \min\{i \in \mathbb{N} : \mathrm{DFR}_1(i) \leqslant 10^{-\gamma}\}.$$

Найденное значение I_γ предполагается использовать в качестве $I_{\max}$ в алгоритме ISDDecoder для ограничения сверху вероятности ошибочного декодирования и, как следствие, обеспечения требуемого уровня отказоустойчивости γ.

В табл. 2 представлены параметры D-кодов вида (5), выбранные для использования в системе шифрования типа Мак-Элиса. Эти коды получены следующим образом. В качестве $\mathrm{RM}(\ell_1, m_2)$ из вида (6) выбраны коды $\mathrm{RM}(2,7)$, $\mathrm{RM}(3,7)$, $\mathrm{RM}(2,8)$, $\mathrm{RM}(3,8)$, $\mathrm{RM}(2,9)$, обладающие высокой корректирующей способностью при использовании де-

кодера Йе — Аббе [17]. Для каждого из этих кодов путем перебора значений параметра m_1 найдены D-коды вида (5), такие, что соответствующие им системы шифрования, во-первых, являются стойкими к комбинированной атаке из [14], а, во-вторых, уровни стойкости λ_{OW} этих систем к классической атаке декодированием по информационным совокупностям сопоставимы с уровнями стойкости соответствующих систем из табл. 1. Для начальной оценки уровня стойкости λ_{OW} по формуле (4) использовано значение числа добавляемых ошибок t, полученное умножением количества ошибок, исправляемых декодером Йе — Аббе для кода $\mathrm{RM}(\ell_1, m_2)$ (с вероятностью ошибочного декодирования не более 10^{-4}), на количество блоков 2^{m_1}. В дальнейшем оценка значения λ_{OW} уточнялась путём подбора для построенных декодеров количества добавляемых ошибок t при экспериментальных вычислениях вероятности ошибочного декодирования, результаты которых содержатся в табл. 3 и 5. В итоге в табл. 2 попали D-коды, у которых размер ключа не больше, чем у систем соответствующего уровня OW-стойкости из табл. 1. В табл. 2 приводятся номер кода и обозначение $[(r_1^1, r_1^2), (r_2^1, r_2^2), \ldots]$ соответствующего D-кода:

$$[(r_1^1, r_1^2), (r_2^1, r_2^2), \ldots] = \mathrm{RM}(r_1^1, m_1) \otimes \mathrm{RM}(r_1^2, m_2) + \mathrm{RM}(r_2^1, m_1) \otimes \mathrm{RM}(r_2^2, m_2) + \ldots,$$

а также приведены параметры m_1, m_2 и $[n, k, d]$.

Таблица 2

D-коды для системы типа Мак-Элиса

Номер кода	D-код	$[n, k, d]$	m_1	m_2
1	$[(1,2),(2,1)]$	$[16384, 376, 2048]$	5	9
2	$[(1,2),(2,1),(3,0)]$	$[16384, 414, 2048]$	6	8
3	$[(0,2),(3,1)]$	$[16384, 406, 1024]$	6	8
4	$[(1,2),(3,1)]$	$[8192, 402, 512]$	5	8
5	$[(1,2),(4,1)]$	$[8192, 603, 256]$	6	7
6	$[(1,2),(3,1)]$	$[16384, 680, 1024]$	7	7
7	$[(1,2),(3,1)]$	$[16384, 476, 1024]$	5	9
8	$[(2,3),(3,0)]$	$[8192, 1498, 256]$	5	8
9	$[(2,3),(3,1)]$	$[8192, 1578, 256]$	5	8
10	$[(1,2),(3,1)]$	$[16384, 574, 1024]$	6	8
11	$[(2,2),(3,0)]$	$[16384, 876, 1024]$	7	7
12	$[(0,2),(4,1)]$	$[16384, 813, 512]$	7	7
13	$[(1,2),(3,1)]$	$[32768, 672, 2048]$	6	9
14	$[(2,3),(3,2)]$	$[8192, 1858, 256]$	5	8
15	$[(2,2),(3,1)]$	$[16384, 1121, 1024]$	7	7
16	$[(2,3),(3,0)]$	$[16384, 2066, 512]$	6	8
17	$[(2,2),(5,1)]$	$[16384, 1569, 256]$	7	7
18	$[(1,2),(4,1)]$	$[32768, 822, 1024]$	6	9
19	$[(2,3),(3,2)]$	$[16384, 2786, 512]$	6	8

В табл. 3 приводится экспериментальная оценка корректирующей способности декодера $\mathsf{ISDDecoder}$ для кодов из табл. 2. Оценка выполнена следующим образом. Для каждого кода из табл. 2 и заданного значения количества ошибок t производилась генерация случайного информационного вектора длины k и случайного вектора ошибок веса t. Затем информационный вектор кодировался и добавлялась ошибка, после чего полученный зашумлённый кодовый вектор подавался на вход алгоритму $\mathsf{BlockDecoder}_{n_1,n_2}$. После выполнения алгоритма происходил подсчёт количества блоков зашумлённого кодового слова, которые были декодированы неправильно (подсчёт

плохих блоков). Перечисленные действия повторялись указанное в табл. 3 количество раз и на основе итогового количества плохих блоков было рассчитано значение P_{bad} из (9) и $\text{DFR}_1(0) = 1-(1-P_{\text{bad}})^{2^{m_1}}$ в соответствии с (8). Значение $\text{DFR}_1(0)$ фактически означает вероятность ошибочного декодирования алгоритмом ISDDecoder до использования ДИС и может быть учтено для оценки адекватности выбранного количества ошибок для обеспечения требуемой отказоустойчивости. Для каждого кода вычислены значения количества блоков K, выбираемых на шаге 11 алгоритма ISDDecoder, и соответствующие им значения вероятностей P_2 из (8). Значения K вычислены экспериментально: для каждого кода в качестве K выбрано минимальное количество блоков, при котором за 100 попыток составления подматрицы из K случайных блоков соответствующей порождающей матрицы хотя бы одна попытка даёт подматрицу ранга, равного размерности кода. При этом, помимо найденного значения K, использовались также значения $K+1$ и $K+2$ для демонстрации влияния на характеристики декодера. С помощью полученных значений в соответствии с (8) вычислено количество I_9 итераций ДИС, необходимых для обеспечения уровня отказоустойчивости $\gamma = 9$ IND-CPA-стойкого механизма KEM. Дополнительно в табл. 3 указано максимальное за наблюдаемое количество экспериментов количество плохих блоков после декодирования зашумлённого кодового вектора.

Как видно из табл. 3, для большинства кодов используемые значения количества добавляемых ошибок t позволяют получить приемлемые значения вероятности $\text{DFR}_1(0)$ и, как следствие, количества итераций I_9. Однако для кодов 2 и 3 вероятность $\text{DFR}_1(0)$ получилась достаточно большой, что привело к значительному увеличению I_9. Также стоит отметить, что для кода 17, несмотря на относительно малую вероятность $\text{DFR}_1(0)$, количество итераций I_9 всё равно получается значительным. Это происходит из-за того, что для кода 17 найденное значение K получается слишком близким к общему количеству блоков (120 из 128), что приводит к тому, что вероятность нахождения требуемой информационной совокупности во время работы каждой итерации ДИС оказывается маленькой и соответственно требуется большее количество итераций для обеспечения уровня отказоустойчивости $\gamma = 9$.

Таблица 3

Оценка корректирующей способности декодера ISDDecoder

Номер кода	K	P_2	t	Кол-во экспериментов	Макс. кол-во плохих блоков	$\text{DFR}_1(0)$	I_9
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
1	16	0,3233	4000	500000	2	0,00259	100
1	17	0,6313	4000	500000	2	0,00259	57
1	18	0,8301	4000	500000	2	0,00259	49
1	16	0,3233	3749	500000	1	0,000136	67
1	17	0,6313	3749	500000	1	0,000136	33
1	18	0,8301	3749	500000	1	0,000136	26
2	42	0,2064	3392	500000	5	0,266	67814
2	43	0,4634	3392	500000	5	0,266	44720
2	44	0,6617	3392	500000	5	0,266	47471
3	42	0,1995	3392	500000	5	0,266	70160
3	43	0,4572	3392	500000	5	0,266	45327
3	44	0,6558	3392	500000	5	0,266	47899
6	64	0,1528	1706	500000	2	0,0152	351
6	65	0,3763	1706	500000	2	0,0152	147

Окончание табл. 3

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
6	66	0,5552	1706	500000	2	0,0152	103
7	26	0,4935	3830	500000	1	0,000332	264
7	27	0,8251	3830	500000	1	0,000332	237
7	28	0,9642	3830	500000	1	0,000332	339
8	26	0,4878	612	50000	1	0,00337	901
8	27	0,8265	612	50000	1	0,00337	1059
8	28	0,9645	612	50000	1	0,00337	2270
9	26	0,4880	612	50000	1	0,00337	901
9	27	0,8238	612	50000	1	0,00337	1062
9	28	0,9645	612	50000	1	0,00337	2270
10	42	0,2072	3237	500000	4	0,101	6920
10	43	0,4565	3237	500000	4	0,101	4099
10	44	0,6628	3237	500000	4	0,101	3742
11	64	0,1598	1706	500000	2	0,0152	336
11	65	0,3660	1706	500000	2	0,0152	151
11	66	0,5648	1706	500000	2	0,0152	101
12	99	0,1772	1706	500000	2	0,0152	3717
12	100	0,3972	1706	500000	2	0,0152	1861
12	101	0,5871	1706	500000	2	0,0152	1420
13	42	0,2126	7687	500000	1	0,000772	231
13	43	0,4690	7687	500000	1	0,000772	113
13	44	0,6626	7687	500000	1	0,000772	88
13	42	0,2126	7772	500000	2	0,00131	275
13	43	0,4690	7772	500000	2	0,00131	135
13	44	0,6626	7772	500000	2	0,00131	105
14	26	0,4883	673	50000	2	0,0147	3827
14	27	0,8288	673	50000	2	0,0147	5088
14	28	0,9667	673	50000	2	0,0147	18011
15	64	0,1550	1706	500000	2	0,0152	346
15	65	0,3624	1706	500000	2	0,0152	152
15	66	0,5592	1706	500000	2	0,0152	102
16	42	0,2090	1292	50000	2	0,0177	941
16	43	0,4511	1292	50000	2	0,0177	508
16	44	0,6626	1292	50000	2	0,0177	407
17	120	0,3592	1706	500000	2	0,0152	329479
17	121	0,6908	1706	500000	2	0,0152	342659
17	122	0,8802	1706	500000	2	0,0152	627527
18	57	0,4184	7279	500000	1	0,0000356	229
18	58	0,7468	7279	500000	1	0,0000356	149
18	59	0,9161	7279	500000	1	0,0000356	146
19	42	0,2060	1083	50000	1	0,00094	255
19	43	0,4562	1083	50000	1	0,00094	125
19	44	0,6560	1083	50000	1	0,00094	95

В табл. 4 приводится оценка времени декодирования с помощью алгоритма ISDDecoder кодов из табл. 2. Для каждого набора параметров экспериментов из табл. 3 (D-кода, количества блоков K и числа ошибок t) приводится время декодирования в миллисекундах (мс), а именно: T_{block} — среднее время выполнения алгоритма $\mathsf{BlockDecoder}_{n_1,n_2}$ на шаге 2; T_{ISD} — среднее время выполнения одной итерации ДИС (шаги 10–18); T_1 — среднее время декодирования с помощью алгоритма ISDDecoder, вычисленное в соответствии с (10); T_1' — время выполнения алгоритма для обеспече-

ния уровня отказоустойчивости $\gamma = 9$ в худшем случае (без раннего выхода из ДИС), то есть $T_1' = T_{\text{block}} + T_{\text{ISD}} I_9$.

Таблица 4

Оценка времени декодирования алгоритмом ISDDecoder

Номер кода	K	t	T_{block}	T_{ISD}	T_1'	T_1
1	*2*	*3*	*4*	*5*	*6*	*7*
1	16	4000	1718	33	5018	1821
1	17	4000	1742	31	3509	1792
1	18	4000	1726	34	3392	1768
1	16	3749	1718	33	3929	1821
1	17	3749	1742	31	2765	1792
1	18	3749	1726	34	2610	1767
2	42	3392	897	42	2849085	1268
2	43	3392	884	40	1789684	1049
2	44	3392	875	42	1994657	1003
3	42	3392	896	44	3087936	1298
3	43	3392	895	43	1949956	1075
3	44	3392	890	39	1868951	1010
6	64	1706	428	86	30614	1000
6	65	1706	430	86	13072	663
6	66	1706	422	89	9589	585
7	26	3830	1725	49	14661	1825
7	27	3830	1706	50	13556	1767
7	28	3830	1707	51	18996	1761
8	26	612	43635	208	231043	44068
8	27	612	43615	206	261769	43869
8	28	612	43114	180	451714	43306
9	26	612	44601	193	218494	45003
9	27	612	44765	193	249731	45004
9	28	612	45051	192	480891	45255
10	42	3237	868	62	429908	1236
10	43	3237	867	68	279599	1053
10	44	3237	864	67	251578	993
11	64	1706	423	131	44439	1256
11	65	1706	422	131	20203	786
11	66	1706	415	132	13747	653
12	99	1706	421	121	450178	1141
12	100	1706	429	120	223749	749
12	101	1706	428	123	175088	650
13	42	7687	3528	161	40719	4287
13	43	7687	3581	158	21435	3919
13	44	7687	3535	158	17439	3774
13	42	7772	3528	161	47803	4288
13	43	7772	3581	158	24911	3919
13	44	7772	3535	158	20125	3775
14	26	673	45288	314	1246966	45974
14	27	673	45792	260	1368672	46133
14	28	673	45173	263	4782066	45476
15	64	1706	425	207	72047	1782
15	65	1706	412	206	31724	990
15	66	1706	418	204	21226	789
16	42	1292	85186	637	684603	88288

Окончание табл. 4

1	*2*	*3*	*4*	*5*	*6*	*7*
16	43	1292	85381	637	408977	86819
16	44	1292	85451	635	343896	86428
17	120	1706	431	385	126849846	1785
17	121	1706	429	386	132266803	1162
17	122	1706	429	381	239088216	1026
18	57	7279	3416	226	55170	3957
18	58	7279	3430	224	36806	3731
18	59	7279	3440	228	36728	3689
19	42	1083	81554	1151	375059	87152
19	43	1083	82954	1153	227079	85487
19	44	1083	82074	1150	191324	83831

В табл. 5 приводится оценка корректирующей способности декодера MatrixDecoder и его времени декодирования для кодов из табл. 2. В качестве алгоритмов DecRow и DecCol использован декодер Йе — Аббе [17]. В табл. 5 также содержатся оценка DFR_2 в соответствии с (11) и оценка T_2 времени выполнения алгоритма (в мс) в соответствии с (12).

Таблица 5

Оценка корректирующей способности декодера MatrixDecoder **и его времени декодирования**

Номер кода	t	DFR_2	T_2	Номер кода	t	DFR_2	T_2
1	4000	1,54E–12	2375	11	1706	5,963E–20	7985
1	3749	1,173E–17	2375	12	1706	2,167E–9	127822
2	3392	0,000273	3172	13	7687	1,344E–14	8673
3	3392	0,000273	3235	13	7772	1,128E–13	8673
4	1744	0,0296	906	14	673	0,000104	45800
5	1194	0,119	9892	15	1706	5,963E–20	8001
6	1706	5,963E–20	7828	16	1292	3,802E–9	81084
7	3830	5,338E–8	2907	17	1706	0,000115	1233434
8	612	5,522E–6	43597	18	7279	6,379E–10	41893
9	612	5,522E–6	46722	19	1083	2,955E–14	80396
10	3237	4,557E–6	3156				

Отметим, что алгоритмы декодирования ISDDecoder и MatrixDecoder имеют свои особенности и могут применяться на практике в зависимости от задачи. Так, декодер MatrixDecoder для некоторых кодов обладает меньшим значением DFR по сравнению с ISDDecoder при сопоставимом времени декодирования. Однако этот DFR для заданного кода и количества добавляемых ошибок является фиксированным, в то время как при использовании ISDDecoder можно добиться сколь угодно малых значений DFR за счёт увеличения максимального времени декодирования. Например, для кода 1 из табл. 2 и $t = 3749$ декодеру ISDDecoder в худшем случае потребуется 26 итераций ДИС и около 2600 мс для обеспечения вероятности ошибочного декодирования 10^{-9}; для тех же параметров декодер MatrixDecoder при меньшем времени декодирования (2300 мс) обеспечивает DFR около 10^{-17}, что значительно меньше. Однако, например, для кодов 4 и 5 декодер MatrixDecoder не может обеспечить даже $\mathrm{DFR} = 10^{-9}$, в отличие от ISDDecoder.

В табл. 6 приводятся параметры системы шифрования типа Мак-Элиса на основе D-кодов из табл. 2. В частности, приводится значение размера открытого ключа в ме-

габайтах (Мбайт), а также значение уровня OW-стойкости λ_{OW} в соответствии с (4) при выбранном значении количества ошибок t.

Таблица 6

Характеристики системы $\mathsf{McE}(D, \mathsf{DDecoder})$, $\mathsf{DDecoder} \in \{\mathsf{ISDDecoder}, \mathsf{MatrixDecoder}\}$

Номер кода	Размер ключа, Мбайт	t	λ_{OW}	Номер кода	Размер ключа, Мбайт	t	λ_{OW}
1	0,734	4000	153,87	11	1,711	1706	143,04
1	0,734	3749	142,82	12	1,588	1706	132,47
2	0,809	3392	140,55	13	2,625	7687	262,28
3	0,793	3392	137,8	13	2,625	7772	265,61
4	0,393	1744	142,82	14	1,814	673	262,22
5	0,589	1194	142,81	15	2,189	1706	184,58
6	1,328	1706	110,3	16	4,035	1292	262,47
7	0,93	3830	185,96	17	3,064	1706	262,40
8	1,463	612	186,11	18	3,211	7279	302,23
9	1,541	612	197,24	19	5,441	1083	302,32
10	1,121	3237	185,94				

В табл. 7 приводится сравнение характеристик системы шифрования на кодах Гоппы ($\mathsf{ME}(G)$) и системы шифрования типа Мак-Элиса на D-кодах. В качестве кодов Гоппы рассматриваются коды с характеристиками из табл. 1, причём размер ключа вычисляется как для случая, когда шифрование выполняется по правилу, предложенному Р. Мак-Элисом (Original McEliece), так и для шифрования по правилу, предложенному Х. Нидеррайтером Classic McEliece. Для сравнения выбраны D-коды из табл. 2, которые при меньшем размере ключа, чем в Original McEliece, обладают таким же уровнем OW-стойкости λ_{OW}, как и соответствующая система Мак-Элиса на кодах Гоппы. Среди множества подходящих D-кодов выбирался код с наименьшим временем декодирования при $\mathrm{DFR} \leqslant 10^{-9}$, то есть в таблице фактически рассматривается случай $\gamma = 9$.

Таблица 7

Сравнение характеристик систем шифрования типа Мак-Элиса

$\mathsf{McE}(G)$					$\mathsf{McE}(D, \mathsf{DDecoder})$			
Криптосистема mceliece	t	λ_{OW}	Размер ключа, Original McEliece, Мбайт	Размер ключа, Classic McEliece, Мбайт	Номер кода	t	λ_{OW}	Размер ключа, Мбайт
348864	64	142,8	1,13	0,25	1	3749	142,82	0,73
460896	96	185,92	1,85	0,5	7	3830	185,96	0,93
6688128	128	262,28	4,01	1	13	7687	262,28	2,63
6960119	119	265,6	4,49	1	13	7772	265,61	2,63
8192128	128	302,21	6,38	1,29	18	7279	302,23	3,21

Как видно из табл. 7, использование D-кодов в системе шифрования типа Мак-Элиса позволяет сократить размер открытого ключа относительно системы Original McEliece. Отметим, что в табл. 7 имеются D-коды, для которых система шифрования обладает меньшим размером ключа даже при значительно большем уровне стойкости λ_{OW}. Например, система на коде 7 обладает меньшим размером открытого ключа и большим уровнем стойкости относительно системы Original McEliece с параметрами mceliece348864, а система на коде 18 — относительно системы с параметрами

mceliece6960119. При этом для уровня отказоустойчивости $\gamma = 9$ размер открытых ключей системы на D-кодах не столь существенно, но все же превышает размер открытых ключей системы Classic McEliece. Представляется, что размер ключей системы на D-кодах может быть уменьшен, если ослабить требование к отказоустойчивости, то есть при $\gamma < 9$.

Сравнивая системы шифрования Мак-Элиса на кодах Гоппы и D-кодах, можно заметить, что время расшифрования у последней больше при декодировании как алгоритмом ISDDecoder, так и алгоритмом MatrixDecoder. Однако в табл. 1 приводится время расшифрования при использовании оптимизированной реализации декодера из [3], в то время как в табл. 4 и 5 указано время работы алгоритмов, для которых не ставилась цель создания оптимальной реализации. В действительности существует большой потенциал для ускорения реализации алгоритмов ISDDecoder и MatrixDecoder, например, они хорошо поддаются распараллеливанию, в том числе за счёт параллельного декодирования блоков в $\mathsf{BlockDecoder}_{n_1,n_2}$ и $\mathsf{BlockDecoder}_{n_2,n_1}$.

5. Механизм KEM на основе $\mathsf{McE}(D, \mathsf{DDecoder})$ и его характеристики

Тройку алгоритмов механизма защищённой передачи, полученного путём применения преобразования Фудзисаки — Окамото $\mathsf{FO}^{\not\perp}$ на основе схемы шифрования $\mathsf{McE}(D, \mathsf{DDecoder})$, обозначим $\mathsf{KEM}(D, \mathsf{DDecoder}) = (\mathsf{Gen}^{\mathsf{KEM}(D)}, \mathsf{Encaps}, \mathsf{Decaps})$ (рис. 5).

	$\mathsf{Gen}^{\mathsf{KEM}(D)}(t)$:	$\mathsf{Encaps}(pk')$:	$\mathsf{Decaps}(sk', \mathbf{c}, \mathsf{ExitCond})$:
1	$\mathbf{s} \leftarrow \mathbb{F}_2^k$	$\mathbf{r} \leftarrow \mathbb{F}_2^k$	$\mathbf{r}' \leftarrow \mathsf{Dec}^{\mathsf{DDecoder}}(sk, \mathbf{c})$
2	$(pk, sk) \leftarrow \mathsf{Gen}^{\mathsf{McE}(D)}(t)$	$\mathbf{c} \leftarrow \mathsf{Enc}(pk, \mathbf{r}; \mathsf{G}(\mathbf{r}))$	*если* $\mathbf{r}' \neq \perp$
3	$pk' = pk$	$\mathbf{K} \leftarrow \mathsf{H}(\mathbf{r} \parallel \mathbf{c})$	*если* $\mathsf{ExitCond} = \mathsf{ExitCond}_{\mathsf{wt}}$
4	$sk' = (sk, \mathbf{s})$	***вернуть*** $(\mathbf{K}, \mathbf{c})$	$\mathbf{c}' \leftarrow \mathsf{Enc}(pk, \mathbf{r}'; \mathsf{G}(\mathbf{r}'))$
5	***вернуть*** (pk', sk')		*если* $\mathbf{c} = \mathbf{c}'$
6			***вернуть*** $\mathsf{H}(\mathbf{r}' \parallel \mathbf{c})$
7			*иначе*
8			***вернуть*** $\mathsf{H}(\mathbf{r}' \parallel \mathbf{c})$
9			***вернуть*** $\mathsf{H}(\mathbf{s} \parallel \mathbf{c})$

Рис. 5. Механизм $\mathsf{KEM}(D, \mathsf{DDecoder}) = (\mathsf{Gen}^{\mathsf{KEM}}, \mathsf{Encaps}, \mathsf{Decaps})$, полученный путём применения преобразования $\mathsf{FO}^{\not\perp}$ на основе схемы $\mathsf{McE}(D, \mathsf{DDecoder})$ с детерминированным правилом шифрования; $\mathsf{ExitCond} \in \{\mathsf{ExitCond}_{\mathsf{wt}}, \mathsf{ExitCond}_{\mathsf{G}}\}$

Согласно [18], при выполнении условия (3) для многоразового использования ключей механизм KEM, полученный применением преобразования Фудзисаки — Окамото $\mathsf{FO}^{\not\perp}$, обеспечивает IND-CCA-стойкость уровня λ_{KEM}, если для схемы асимметричного шифрования $\mathsf{McE}(D, \mathsf{DDecoder})$ выполняется условие (1). Отсюда вытекает и IND-CPA-стойкость механизма при выполнении тех же условий.

Выше отмечалось, что выбор параметров $\mathsf{McE}(D, \mathsf{DDecoder})$ следует выполнять, исходя из обеспечения стойкости к структурной атаке, построенной в [13, 14], и из требования обеспечения заданного уровня OW-стойкости λ_{OW}. При синтезе механизма $\mathsf{KEM}(D, \mathsf{DDecoder})$ на основе $\mathsf{McE}(D, \mathsf{DDecoder})$ дополнительными условиями на выбор параметров схемы $\mathsf{McE}(D, \mathsf{DDecoder})$ являются условия (1) и (3).

Схема $\mathsf{McE}(D, \mathsf{DDecoder})$ в случае применения гарантированного декодера, например GraphDecoder [15], позволяет построить $\mathsf{KEM}(D, \mathsf{GraphDecoder})$ с многоразовым использованием ключей, так как условие (3) выполняется ($\delta = 0$). Однако результа-

ты исследования [14] показали, что для обеспечения целевых значений уровней OW-стойкости, сопоставимых со значениями из табл. 1, приходится использовать ключи существенно большего размера, чем в Original McEliece.

Как показывают результаты в табл. 7, при использовании вероятностных декодеров ISDDecoder и MatrixDecoder в настоящей работе не удалось найти D-коды, для которых бы размер ключа системы шифрования не превышал размер ключа системы Original McEliece из табл. 1, обеспечивая при этом уровень стойкости не менее соответствующего значения λ_{OW}, и для которых вероятность ошибочного декодирования не превышала бы $2^{-\lambda_{\mathsf{OW}}}$ (при $I_{\max} \leqslant 10^6$). Как следствие, найденные D-коды (см. табл. 2) для схемы $\mathsf{McE}(D, \mathsf{DDecoder})$ с построенными вероятностными декодерами не позволяют при меньшем размере ключа, чем в Original McEliece, построить механизм $\mathsf{KEM}(D, \mathsf{DDecoder})$ с соответствующим уровнем IND-CCA-стойкости λ_{KEM}. Действительно, для обеспечения IND-CCA-стойкости необходимо, с одной стороны, чтобы уровень стойкости λ_{KEM} был не меньше, чем длина m сеансового ключа K (см. (1)), которая обычно не меньше 128, а с другой — чтобы DFR был меньше, чем $2^{-\lambda_{\mathsf{KEM}}}$ (см. (3)). Согласно табл. 3 и 5, минимальный полученный DFR для используемых параметров D-кодов оказался значительно больше 2^{-128}. Однако найденные коды позволяют обеспечить уровень отказоустойчивости $\gamma = 9$ для IND-CPA-стойкого механизма KEM с одноразовым использованием ключей (см. условие (3) для одноразового использования). В частности, такие коды удалось найти за счёт подбора максимального числа итераций $I_{\max}$ в алгоритме ISDDecoder.

Таким образом, система Мак-Элиса на кодах Гоппы (как Original McEliece, так и Classic McEliece), в силу нулевой вероятности ошибочного декодирования, позволяет построить KEM с многоразовым использованием ключей; как следствие, этот же механизм может применяться для одноразового использования. Схема Мак-Элиса на D-кодах при сопоставимых размерах ключа (меньшем размере по отношению к Original McEliece и несколько большем — по отношению к Classic McEliece) в случае использования вероятностных алгоритмов декодирования 1 и 2 позволяет построить только KEM с одноразовым использованием ключей.

Заключение

Известные асимметричные кодовые системы шифрования на основе кодов Рида — Маллера [36–38] оказались нестойкими к структурным атакам [39–44]. В [13–15] и настоящей работе предпринимается попытка реанимировать двоичные коды Рида — Маллера в контексте их применения в асимметричных системах шифрования. Для достижения этой цели используется, с одной стороны, конструкция D-кодов (для противодействия известным структурным атакам), а с другой — наличие относительно эффективных декодеров для кодов Рида — Маллера, способных исправлять ошибки веса более половины кодового расстояния (для уменьшения размера открытого ключа). В настоящей работе удалось построить механизм KEM с меньшим ключом, чем в оригинальной системе шифрования на кодах Гоппы Original McEliece, только для передачи эфемерных (одноразовых) сеансовых криптографических ключей. Однако, учитывая интерес криптографического сообщества к KEM с одноразовым использованием ключей [24], это ограничение представляется не таким существенным. Кроме того, как было недавно показано, двоичные коды Рида — Маллера достигают ёмкости в симметричных каналах без памяти [30], поэтому ожидается дальнейшее развитие методов эффективного декодирования этих кодов, которые могут позволить не только уменьшить время расшифрования в предлагаемой системе шифрования, но и повы-

сить класс стойкости за счёт уменьшения вероятности ошибочного декодирования. Возможность исправлять ошибки большого веса также может быть использована для усиления структурной стойкости системы, например, за счёт добавления случайных столбцов к порождающей матрице кода и/или замены матрицы перестановки P невырожденной матрицей специального вида.

Авторы выражают благодарность рецензенту за внимательное прочтение статьи, а также за полезные замечания и советы, которые позволили ее улучшить.

ЛИТЕРАТУРА

1. *Weger V., Gassner N., and Rosenthal J.* A Survey on Code-Based Cryptography. arXiv: 2201.07119, 2024. 168 p.
2. *McEliece R. J.* A public-key cryptosystem based on algebraic coding theory // Deep Space Network Progress Report. 1978. No. 44. P. 114–116.
3. *Albrecht M. R., Bernstein D. J., Chou T., et al.* Classic McEliece: Conservative Code-Based Cryptography. NIST PQC Call for Proposals, 2022. Round 4 Submission. `https://classic.mceliece.org/nist/mceliece-20201010.pdf`.
4. *Alagic G., Apon D., Cooper D., et al.* Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. NIST, 2022. `https://csrc.nist.gov/pubs/ir/8413/upd1/final`.
5. *Niederreiter H.* Knapsack-type cryptosystems and algebraic coding theory // Prob. Contr. Inform. Theory. 1986. V. 15. No. 2. P. 157–166.
6. *Albrecht M., Cid C., Paterson K. G., et al.* Nts-kem // NIST PQC Round. 2019. V. 2. P. 4–13.
7. *Couvreur A., Mora R., and Tillich J.-P.* A New Approach Based on Quadratic Forms to Attack the McEliece Cryptosystem. Cryptology ePrint Archive. 2023. Paper 2023/950.
8. *Bardet M., Mora R., and Tillich J.-P.* Polynomial Time Key-Recovery Attack on High Rate Random Alternant Codes. arXiv: 2304.14757, 2023.
9. *Mora R. and Tillich J.-P.* On the dimension and structure of the square of the dual of a Goppa code // Des. Codes Cryptogr. 2023. No. 91. P. 1351–1372.
10. *Ivanov F., Kabatiansky G., Krouk E., and Rumenko N.* A new code-based cryptosystem // LNCS. 2020. V. 12087. P. 41–49.
11. *Lau T. Sh. C., Ivanov F., Ariffin M. R. K., et al.* New code-based cryptosystems via the IKKR framework // J. Inform. Security Appl. 2023. V. 76. Article 103530.
12. *Krouk E., Kabatiansky G., and Tavernier C.* McEliece-type cryptosystem based on correction of errors and erasures // 2023 XVIII Intern. Symp. REDUNDANCY. Moscow, Russia, 2023. P. 173–177.
13. *Косолапов Ю. В., Лелюк Е. А.* О структурной стойкости криптосистемы типа Мак-Элиса на сумме тензорных произведений бинарных кодов Рида — Маллера // Прикладная дискретная математика. 2022. № 57. С. 22–39.
14. *Косолапов Ю. В., Лелюк Е. А.* Криптосистема типа Мак-Элиса на D-кодах // Математические вопросы криптографии. 2024. Т. 15. №. 2. С. 69–90.
15. *Деундяк В. М., Лелюк Е. А.* Теоретико-графовый метод декодирования некоторых групповых MLD-кодов // Дискретн. анализ и исслед. опер. 2020. Т. 27. № 2. С. 17–42.
16. *Abbe E., Sberlo O., Shpilka A., and Ye M.* Reed — Muller codes // Foundations and Trends in Commun. Inform. Theory. 2023. V. 20. No. 1–2. P. 1–156.
17. *Ye M. and Abbe E.* Recursive projection-aggregation decoding of Reed — Muller codes // IEEE Trans. Inform. Theory. 2020. V. 66. No. 8. P. 4948–4965.
18. *Hofheinz D., Hövelmanns K., and Kiltz E.* A modular analysis of the Fujisaki — Okamoto transformation // LNCS. 2017. V. 10677. P. 341–371

19. *Arago N., Barreto P., Bettaieb S., et al.* BIKE: Bit Flipping Key Encapsulation. `https://hal.science/hal-01671903/document`. 2017.
20. *Guo Q., Johansson T., and Wagner P. S.* A key recovery reaction attack on QC-MDPC // IEEE Trans. Inform. Theory. 2018. V. 65. No. 3. P. 1845–1861.
21. *Vedenev K. V. and Kosolapov Y. V.* A reaction attack against cryptosystems based on quasigroup MDPC codes // 2023 XVIII Intern. Symp. REDUNDANCY. Moscow, Russia, 2023. P. 70–75.
22. *Joux A., Loss J., and Wagner B.* Kleptographic Attacks against Implicit Rejection. `https://eprint.iacr.org/2024/260`. 2024.
23. *Campagna M. and Crockett E.* Hybrid Post-Quantum Key Encapsulation Methods (PQ KEM) for Transport Layer Security 1.2 (TLS). `https://www.ietf.org/archive/id/draft-campagna-tls-bike-sike-hybrid-07.html`. 2021.
24. *Drucker N., Gueron S., and Kostic D.* A Lean BIKE KEM Design for Ephemeral Key Agreement. `https://csrc.nist.gov/csrc/media/Events/2024/fifth-pqc-standardization-conference/documents/papers/a-lean-bike-kem.pdf`. 2024.
25. *Baldi M., Barenghi A., Chiaraluce F., et al.* LEDAcrypt: Low-dEnsity parity-check coDebAsed cryptographic systems. `https://www.ledacrypt.org/documents/LEDAcrypt_v3.pdf`. 2019.
26. *Prange E.* The use of information sets in decoding cyclic codes // IRE Trans. Inform. Theory. 1962. V. 8. No. 5. P. 5–9.
27. *Both L. and May A.* Decoding linear codes with high error rate and its impact for LPN security // LNCS. 2018. V. 10786. P. 25–46.
28. *May A. and Ozerov I.*, On computing nearest neighbors with applications to decoding of binary linear codes // LNCS. 2015. V. 9056. P. 203–228.
29. *Kudekar S., Kumar S., Mondelli M., et al.* Reed — Muller codes achieve capacity on erasure channels // Proc. STOC'16. N.Y.: ACM, 2016. P. 658–669.
30. *Abbe E. and Sandon C.* A proof that Reed — Muller codes achieve Shannon capacity on symmetric channels // IEEE 64th Ann. Symp. FOCS. Santa Cruz, CA, USA, 2023. P. 177–193.
31. *Reed I. S.* A class of multiple-error-correcting codes and the decoding scheme // IEEE Trans. Inform. Theory. 1954. V. 4. No. 4. P. 38–492.
32. *Massey J. L.* Threshold Decoding. Cambridge, MA: MIT Press, 1963.
33. *Циммерман К.-Х.* Методы теории модулярных представлений в алгебраической теории кодирования. М.: МЦНМО, 2011.
34. *Сидельников В. М., Першаков А. С.* Декодирование кодов Рида — Маллера при большом числе ошибок // Пробл. передачи информ. 1992. Т. 28. № 3. С. 80–94.
35. *Kasami T. and Lin S.* On the construction of a class of majority-logic decodable codes // IEEE Trans. Inform. Theory. 1971. V. 17. No. 5. P. 600–610.
36. *Сидельников В. М.* Открытое шифрование на основе двоичных кодов Рида — Маллера // Дискретная математика. 1994. Т. 6. № 2. С. 3–20.
37. *Kabatiansky G. and Tavernier C.* A new code-based cryptosystem via pseudorepetition of codes // 16th Intern. Workshop Algebraic Combinat. Coding Theory. Svetlogorsk, Russia, 2018. P. 189–191.
38. *Egorova E., Kabatiansky G., Krouk E., and Tavernier C.* A new code-based public-key cryptosystem resistant to quantum computer attacks // J. Phys. Conf. Ser. 2019. V. 1163. P. 1–5.
39. *Minder L. and Shokrollahi A.* Cryptanalysis of the Sidelnikov cryptosystem // LNCS. 2007. V. 4515. P. 347–360.

40. *Бородин М. А., Чижов И. В.* Эффективная атака на криптосистему Мак-Элиса, построенную на основе кодов Рида — Маллера // Дискретная математика. 2014. Т. 26. № 1. С. 10–20.
41. *Высоцкая В. В.* Квадрат кода Рида — Маллера и классы эквивалентности секретных ключей криптосистемы Мак-Элиса — Сидельникова // Прикладная дискретная математика. Приложение. 2017. № 10. С. 66–68.
42. *Высоцкая В. В.* О структурных особенностях пространства ключей криптосистемы Мак-Элиса — Сидельникова на обобщенных кодах Рида — Соломона // Дискретная математика. 2024. Т. 36. № 4. С. 28–43.
43. *Давлетшина А. М.* Поиск эквивалентных ключей криптосистемы Мак-Элиса — Сидельникова, построенной на двоичных кодах Рида — Маллера // Прикладная дискретная математика. Приложение. 2019. № 12. С. 98–100.
44. *Чижов И. В., Бородин М. А.* Классификация произведений Адамара подкодов коразмерности 1 кодов Рида — Маллера // Дискретная математика. 2020. Т. 32. № 1. С. 115–134.

REFERENCES

1. *Weger V., Gassner N., and Rosenthal J.* A Survey on Code-Based Cryptography. arXiv:2201.07119, 2024. 168 p.
2. *McEliece R. J.* A public-key cryptosystem based on algebraic coding theory. Deep Space Network Progress Report, 1978, no. 44, pp. 114–116.
3. *Albrecht M. R., Bernstein D. J., Chou T., et al.* Classic McEliece: Conservative Code-Based Cryptography. NIST PQC Call for Proposals, 2022. Round 4 Submission. `https://classic.mceliece.org/nist/mceliece-20201010.pdf`.
4. *Alagic G., Apon D., Cooper D., et al.* Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. NIST, 2022. `https://csrc.nist.gov/pubs/ir/8413/upd1/final`.
5. *Niederreiter H.* Knapsack-type cryptosystems and algebraic coding theory. Prob. Contr. Inform. Theory, 1986, vol. 15, no. 2, pp. 157–166.
6. *Albrecht M., Cid C., Paterson K. G., et al.* Nts-kem. NIST PQC Round, 2019, vol. 2, pp. 4–13.
7. *Couvreur A., Mora R., and Tillich J.-P.* A New Approach Based on Quadratic Forms to Attack the McEliece Cryptosystem. Cryptology ePrint Archive, 2023, Paper 2023/950.
8. *Bardet M., Mora R., and Tillich J.-P.* Polynomial Time Key-Recovery Attack on High Rate Random Alternant Codes. arXiv:2304.14757, 2023.
9. *Mora R. and Tillich J.-P.* On the dimension and structure of the square of the dual of a Goppa code. Des. Codes Cryptogr., 2023, no. 91, pp. 1351–1372.
10. *Ivanov F., Kabatiansky G., Krouk E., and Rumenko N.* A new code-based cryptosystem. LNCS, 2020, vol. 12087, pp. 41–49.
11. *Lau T. Sh. C., Ivanov F., Ariffin M. R. K., et al.* New code-based cryptosystems via the IKKR framework. J. Inform. Security Appl., 2023, vol. 76, article 103530.
12. *Krouk E., Kabatiansky G., Tavernier C* McEliece-type cryptosystem based on correction of errors and erasures. 2023 XVIII Intern. Symp. REDUNDANCY, Moscow, Russia, 2023, pp. 173–177.
13. *Kosolapov Yu. V. and Lelyuk E. A.* O strukturnoy stoykosti kriptosistemy tipa Mak-Elisa na summe tenzornykh proizvedeniy binarnykh kodov Rida — Mallera [On the structural security of a McEliece-type cryptosystem based on the sum of tensor products of binary Reed — Muller codes]. Prikladnaya Diskretnaya Matematika, 2022, no. 57, pp. 22–39. (in Russian)

14. *Kosolapov Yu. V. and Lelyuk E. A.* Kriptosistema tipa Mak-Elisa na D-kodakh [The McEliece-type cryptosystem based on D-codes]. Matematicheskie Voprosy Kriptografii, 2024, vol. 15, no. 2, pp. 69–90. (in Russian)
15. *Deundyak V. M. and Lelyuk E. A.* A graph-theoretical method for decoding some group MLD-codes. J. Appl. Industr. Math., 2020, vol. 14, no. 2, pp. 265–280.
16. *Abbe E., Sberlo O., Shpilka A., and Ye M.* Reed — Muller Codes. Foundations and Trends in Commun. Inform. Theory, 2023, vol. 20, no. 1–2, pp. 1–156.
17. *Ye M. and Abbe E.* Recursive projection-aggregation decoding of Reed — Muller codes. IEEE Trans. Inform. Theory, 2020, vol. 66, no. 8, pp. 4948–4965.
18. *Hofheinz D., Hövelmanns K., and Kiltz E.* A modular analysis of the Fujisaki — Okamoto transformation. LNCS, 2017, vol. 10677, pp. 341–371.
19. *Arago N., Barreto P., Bettaieb S., et al.* BIKE: Bit Flipping Key Encapsulation. `https://hal.science/hal-01671903/document`, 2017.
20. *Guo Q., Johansson T., and Wagner P.S.* A key recovery reaction attack on QC-MDPC. IEEE Trans. Inform. Theory, 2018, vol. 65, no. 3, pp. 1845–1861.
21. *Vedenev K. V. and Kosolapov Y. V.* A reaction attack against cryptosystems based on quasi-group MDPC codes. 2023 XVIII Intern. Symp. REDUNDANCY, Moscow, Russia, 2023, pp. 70–75.
22. *Joux A., Loss J., and Wagner B.* Kleptographic Attacks against Implicit Rejection. `https://eprint.iacr.org/2024/260`, 2024.
23. *Campagna M. and Crockett E.* Hybrid Post-Quantum Key Encapsulation Methods (PQ KEM) for Transport Layer Security 1.2 (TLS). `https://www.ietf.org/archive/id/draft-campagna-tls-bike-sike-hybrid-07.html`, 2021.
24. *Drucker N., Gueron S., and Kostic D.* A Lean BIKE KEM Design for Ephemeral Key Agreement. `https://csrc.nist.gov/csrc/media/Events/2024/fifth-pqc-standardization-conference/documents/papers/a-lean-bike-kem.pdf`, 2024.
25. *Baldi M., Barenghi A., Chiaraluce F., et al.* LEDAcrypt: Low-dEnsity parity-check coDebAsed cryptographic systems. `https://www.ledacrypt.org/documents/LEDAcrypt_v3.pdf`, 2019.
26. *Prange E.* The use of information sets in decoding cyclic codes. IRE Trans. Inform. Theory, 1962, vol. 8, no. 5, pp. 5–9.
27. *Both L. and May A.* Decoding linear codes with high error rate and its impact for LPN security. LNCS, 2018, vol. 10786, 2018, pp. 25–46.
28. *May A. and Ozerov I.*, On computing nearest neighbors with applications to decoding of binary linear codes. LNCS, 2015, vol. 9056, pp. 203–228.
29. *Kudekar S., Kumar S., Mondelli M., et al.* Reed — Muller codes achieve capacity on erasure channels. Proc. STOC'16, N.Y., ACM, 2016, pp. 658–669.
30. *Abbe E. and Sandon C.* A proof that Reed — Muller codes achieve Shannon capacity on symmetric channels. IEEE 64th Ann. Symp. FOCS, Santa Cruz, CA, USA, 2023, pp. 177–193.
31. *Reed I. S.* A class of multiple-error-correcting codes and the decoding scheme. IEEE Trans. Inform. Theory, 1954, vol. 4, no. 4, pp. 38–492.
32. *Massey J. L.* Threshold Decoding. Cambridge, MA, MIT Press, 1963.
33. *Tsimmerman K.-Kh.* Metody teorii modulyarnykh predstavleniy v algebraicheskoy teorii kodirovaniya [Methods of the Modular Representations Theory in Algebraic Coding Theory]. Moscow, MCCME, 2011. (in Russian)
34. *Sidel'nikov V. M. and Pershakov A. S.* Decoding of Reed — Muller codes with a large number of errors. Problems Inform. Transmission, 1992, vol. 28, no. 3, pp. 269–281.

35. *Kasami T. and Lin S.* On the construction of a class of majority-logic decodable codes. IEEE Trans. Inform. Theory, 1971, vol. 17, no. 5, pp. 600–610.
36. *Sidel'nikov V. M.* Open coding based on Reed — Muller binary codes. Discrete Math. Appl., 1994, vol. 4, no. 3, pp. 191–207.
37. *Kabatiansky G. and Tavernier C.* A new code-based cryptosystem via pseudorepetition of codes. 16th Intern. Workshop Algebraic Combinat. Coding Theory, Svetlogorsk, Russia, 2018, pp. 189–191.
38. *Egorova E., Kabatiansky G., Krouk E., and Tavernier C.* A new code-based public-key cryptosystem resistant to quantum computer attacks. J. Phys. Conf. Ser., 2019, vol. 1163, pp. 1–5.
39. *Minder L. and Shokrollahi A.* Cryptanalysis of the Sidelnikov cryptosystem. LNCS, 2007, vol. 4515, pp. 347–360.
40. *Borodin M. A. and Chizhov I. V.* Effective attack on the McEliece cryptosystem based on Reed — Muller codes. Discrete Math. Appl., 2014, vol. 24, no. 5, pp. 273–280.
41. *Vysotskaya V. V* Kvadrat koda Rida-Mallera i klassy jekvivalentnosti sekretnyh kljuchej kriptosistemy Mak-Jelisa-Sidel'nikova [Reed-Muller code square and equivalence classes of secret keys of the McEliece-Sidelnikov cryptosystem]. Prikladnaja diskretnaja matematika. Prilozhenie, 2017, no. 10. pp. 66–68. (in Russian)
42. *Vysotskaya V. V.* O strukturnykh osobennostyakh prostranstva klyuchey kriptosistemy Mak-Elisa — Sidel'nikova na obobshchennykh kodakh Rida — Solomona [On the structural features of the key space of the McEliece — Sidelnikov cryptosystem based on generalized Reed — Solomon codes]. Diskretnaya Matematika, 2024, vol. 36, iss. 4, pp. 28–43. (in Russian)
43. *Davletshina A. M.* Poisk ekvivalentnykh klyuchey kriptosistemy Mak-Elisa — Sidel'nikova, postroennoy na dvoichnykh kodakh Rida — Mallera [Search for equivalent keys of the McEliece — Sidelnikov cryptosystem built on the Reed — Muller binary codes]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2019, no. 12, pp. 98–100. (in Russian)
44. *Chizhov I. V. and Borodin M. A.* Classification of Hadamard products of one-codimensional subcodes of Reed — Muller codes. Discrete Math. Appl., 2022, vol. 32, no. 5, pp. 297–311.