ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ ИМ. А.Н.ТИХОНОВА

Анджушева Манца Мергеновна, группа БИБ191

ОБЗОР АЛГОРИТМОВ КОДОВОЙ КРИПТОГРАФИИ

Междисциплинарная курсовая работа по направлению подготовки
«10.03.01 Информационная безопасность»
студента образовательной программы
«Информационная безопасность»

Студент _____ М.М. Анджушева
подпись И.О. Фамилия

Научный руководитель
старший научный сотрудник
научно-учебной лаборатории
телекоммуникационных систем МИЭМ,
доцент кафедры информационной
безопасности киберфизических систем
Ф.И. Иванов
И.О. Фамилия

Москва, 2021

АННОТАЦИЯ

Цель: обзор известных асимметричных криптосистем на алгебраических кодах, исследование современного состояния, существующих противоречий и перспектив практического применения на постквантовый период

Работа посвящена исследованию современного состояния кодовых криптосистем, подробному изучению тех из них, в отношении которых по сей день не применим успешный криптоанализ. Рассмотрена оригинальная система Мак-Элиса, основанная на двоичных неприводимых кодах Гоппы, а также наиболее успешные с точки зрения стойкости и реализации её вариации.

Значительное внимание уделено теории алгебраического кодирования, изучены структуры кодов, рассмотренных в работе, и их декодеры. Произведён анализ большого объёма современной научной литературы, преимущественно зарубежной.

В результате составлен обзор актуальных модификаций криптосистемы Мак-Элиса, исследован вопрос их стойкости и применимости в настоящее время.

СОДЕРЖАНИЕ

Введение

Концепция криптографии с открытым ключом впервые была изложена Диффи и Хеллманом в 1976 году. В своей основополагающей работе [1] они предложили использовать односторонние функции для построения публичного ключа так, чтобы атакующему пришлось бы решать трудную задачу (например, задачу об упаковке рюкзака), в то время как законный пользователь, обладая секретным ключом, сможет свести эту задачу к выполнимой и легко декодировать сообщение.

Оказалось, что большинство таких криптосистем в определённых обстоятельствах ненадёжны. Так, Шамир в [2] указал на слабые места в преобразованиях и взломал систему Меркли-Хеллмана. Кроме того, в настоящее время ведутся активные работы по созданию квантового компьютера, и есть серьёзные опасения, что в скором будущем современные стандарты асимметричного шифрования перестанут быть надёжными.

Одним из перспективных направлений постквантовой криптографии является кодовая криптография. Первой и наиболее изученной схемой несимметричного шифрования, основанной на использовании алгебраических блоковых кодов, является предложенная в 1978 году криптосистема Мак-Элиса [3]. Главная её идея состоит в том, чтобы спрятать структуру кода с известным полиномиальным алгоритмом декодирования и замаскировать его под случайный линейный код. Стойкость такой криптосистемы обусловлена тем, что задача декодирования произвольного линейного кода является NP-трудной.

Оригинальная криптосистема Мак-Элиса, предложенная около 40 лет назад, остаётся стойкой ко всем известным методам криптоанализа, что в исторической ретроспективе подтверждает надёжность и перспективность крипто-кодовых преобразований, особенно в контексте построения эффективных постквантовых алгоритмов криптографической защиты. Несмотря на это, криптосистема Мак-Элиса не получила широкого практического применения – в основном из-за большого размера ключей.

Чтобы обойти эту проблему, вместо кодов Гоппы, использованных в оригинальной криптосистеме, было предложено использовать другие коды, например коды Рида-Маллера [4], обобщённые коды Рида–Соломона [5], LDPC-коды [6] и другие. Однако для многих из предложенных вариантов криптосистемы были найдены атаки на структуру матрицы публичного ключа, позволяющие вычислить закрытый ключ из открытого. Все эти атаки использовали различные особенности данных кодов. Так, атаки на криптосистему Мак-Элиса, построенную на кодах Рида–Маллера [7] или на полярных кодах [8], использовали поиск слов кода минимального веса.

В данном обзоре речь пойдёт об оригинальной криптосистеме Мак-Элиса и о тех её модификациях, которые до сих пор не поддаются эффективному криптоанализу.

Цель: обзор известных асимметричных криптосистем на алгебраических кодах, исследование современного состояния, существующих противоречий и перспектив практического применения на постквантовый период.

Задачи:

- углубить знания о криптосистемах с открытым ключом;

- изучить криптосистему Мак-Элиса – первую кодовую криптосистему;

- исследовать линейные коды, имеющие полиномиальную сложность декодирования, на возможность построить на их основе асимметричные криптосистемы;

- описать те алгоритмы кодовой криптографии, которые остаются стойкими на сегодняшний день.

1 Алгебраические коды для несимметричных криптосистем

Кодовая криптография опирается на теорию алгебраического кодирования, поэтому для дальнейшего изложения я кратко введу те определения и обозначения, которые будут использованы везде в последующих разделах.

Определение 1. Линейным q-ичным (n, k)-кодом C называется любое k-мерное подпространство линейного пространства F_q^n всевозможных векторов длины n.

Важной характеристикой кода является его минимальное расстояние d в смысле расстояния Хэмминга:

$$d = \min_{x,y \in C, x \neq y} d(x,y).$$

Для линейного кода можно записать:

$$d = \min_{x \in C, x \neq 0} d(x,0) = \min_{x \in C, x \neq 0} w(x),$$

где *w(x)* – число ненулевых элементов в последовательности x или вес вектора x в метрике Хэмминга.

Будем использовать тот факт, что код с минимальным расстоянием d исправляет любые комбинации ошибок кратности $t \leq \lfloor (d-1)/2 \rfloor$.

Определение 2. Порождающей матрицей G линейного (n, k)-кода называется матрица размера $k \times n$, строки которой – его базисные векторы.

Если передаётся сообщение $m = (m_1, \dots, m_k)$, то соответствующее ему кодовое слово вычисляется по формуле

$$c = mG.$$

Определение 3. Проверочной матрицей H линейного (n, k)-кода называется матрица размера $r \times n$, где r = n – k – избыточность кода. Проверочная и порождающая матрицы связаны соотношением

$$GH^T = 0.$$

Теперь перейдём к постановке задачи декодирования. Рассмотрим код, исправляющий ошибки кратности t. Пользователь принимает вектор r – сумму кодового слова и вектора ошибки:

$$r = mG + e$$

где e – вектор размера $1 \times n$, $w(e) \leq t$.

Задача декодирования: по данному вектору с ошибкой найти ближайшее кодовое слово из кода. Формально, это задача поиска такого кодового слова c, что

$$c = \underset{\substack{c' \in C \\ w(e) \leq t}}{\operatorname{argmin}} d(c', r),$$

Следует отметить, что при больших n и k задача декодирования произвольного линейного кода является чрезвычайно сложной математической задачей. В общем случае эта задача относится к классу NP-сложных. Однако для алгебраических кодов, со специфической структурой матриц G и H, декодирование является полиномиально разрешимой задачей.

Теперь перейдём к рассмотрению специальных классов кодов, которые нашли широкое применение в криптографии.

Определение 4. $G(x)$ – неприводимый многочлен с коэффициентами из поля $GF(q^m)$. $L = \{\propto_0, \propto_2, \dots, \propto_{n-1}\}$ – множество различных элементов поля $GF(q^m)$ такое, что $G(\propto_i) \neq 0 \ \forall \propto_i \in L$.

Неприводимым кодом Гоппы Γ(L, G) длины n с символами из поля $GF(q)$ называется код, состоящий из всех таких векторов $u = (u_0, \dots, u_{n-1})$, что

$$R_u(x) = \sum_{i=0}^{n-1} \frac{u_i}{x - u_i} \equiv 0 \; mod \; G(x).$$

Проверочная матрица кода имеет вид:

$$H = \begin{pmatrix} 1 & \dots & 1 \\ \alpha_1 & \dots & \alpha_n \\ \dots & \dots & \dots \\ \alpha_1^{r-1} & \dots & \alpha_n^{r-1} \end{pmatrix} \begin{pmatrix} G^{-1}(\alpha_1) & \dots & \dots \\ \dots & G^{-1}(\alpha_2) & \dots \\ \dots & \dots & \dots \\ \dots & \dots & G^{-1}(\alpha_n) \end{pmatrix},$$

где $r = \deg G(x)$.

Определение 5. Линейный код C называется квазициклическим, если существует такое $n_0 \in \mathbb{N}$, что любой циклический сдвиг слова $c \in C$ на n_0 позиций снова является кодовым словом кода C:

$$x^{n_0 m} c \; mod \; (x^n - 1) \in C \; \forall m \in \mathbb{N}.$$

Если $n = n_0 p$, то матрицы G и H могут быть составлены из $p \times p$ циркулянтов.

Определение 6. Квадратная матрица D называется циркулянтом, если все её строки/столбцы d_i. $i > 1$ являются различными циклическими сдвигами первой строки/столбца d_1. Таким образом, циркулянт полностью определяется своей первой строкой/столбцом.

Определение 7. Линейный код C длины n называется *регулярным квазициклическим кодом с малой плотностью проверок (QC-LDPC)*, если:

1) он квазициклический;
2) все строки его проверочной матрицы H имеют малый вес w_r по сравнению с длиной строки;
3) все столбцы его проверочной матрицы H имеют малый вес w_c по сравнению с высотой столбца.

2 Классическая криптосистема Мак-Элиса

Данная криптосистема впервые была опубликована в 1978 году Робертом Мак-Элисом. В оригинале она использует двоичные неприводимые коды Гоппы [9], так как для

их декодирования существует, например, алгоритм Паттерсона [10] и они обладают хорошими криптосвойствами. В общем случае система определяется следующим образом:

Пусть $G - k \times n$ порождающая матрица линейного q-ичного (n, k, d)-кода C, для которого известен эффективный алгоритм декодирования $\mathrm{Dec_C}$ ошибок, весом не более t. S – случайная обратимая $k \times k$ матрица, P - случайная $n \times n$ перестановочная матрица.

Секретный ключ: $(Dec_C,\ S,\ P)$

Открытый ключ: $G' = S \cdot G \cdot P$

Шифрование: пусть m – вектор длины k и пусть e - случайный вектор длины n с весом t. Тогда шифротекст: $c = m \cdot G' + e$.

Дешифрование:

1. $c' \leftarrow c \cdot P^{-1}$
2. $m' \leftarrow Dec_C(c')$. Получим $m' = m \cdot S$
3. $m \leftarrow m' \cdot S^{-1}$

Пример. Возьмём порождающую матрицу G кода Гоппы $\Gamma\left(x^2 + x + 1, GF(2^4)\right)$.

$$G = \begin{pmatrix} 1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&1 \\ 1&0&1&0&1&1&0&1&0&0&0&0&0&0&1&0 \\ 1&0&0&1&1&1&1&1&0&0&0&0&0&1&0&0 \\ 0&0&1&1&1&1&0&0&0&0&0&0&1&0&0&0 \\ 0&1&1&1&1&0&1&0&0&0&0&1&0&0&0&0 \\ 0&1&1&0&0&0&1&1&0&0&1&0&0&0&0&0 \\ 0&1&1&0&0&1&0&1&0&1&0&0&0&0&0&0 \\ 1&0&1&0&0&1&1&0&1&0&0&0&0&0&0&0 \end{pmatrix}$$

Сгенерируем матрицы S и P:

$$S = \begin{pmatrix} 1&1&1&1&1&1&1&1 \\ 0&1&1&1&1&1&1&1 \\ 0&0&1&1&1&1&1&1 \\ 0&0&0&1&1&1&1&1 \\ 0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&1&1&1 \\ 0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&1 \end{pmatrix}$$

$$P = \begin{pmatrix} 1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0 \end{pmatrix}$$

Вычислим открытый ключ:

$$G' = S \cdot G \cdot P = \begin{pmatrix} 0\,1\,0\,1\,1\,0\,1\,0\,0\,1\,1\,1\,1\,1\,1 \\ 1\,0\,1\,1\,0\,0\,1\,0\,0\,1\,0\,1\,1\,1\,1 \\ 0\,1\,1\,0\,0\,1\,1\,0\,1\,1\,0\,1\,1\,1\,0 \\ 1\,1\,1\,1\,1\,0\,1\,1\,0\,1\,0\,1\,1\,1\,0\,0 \\ 1\,0\,1\,0\,0\,1\,1\,1\,0\,1\,0\,1\,1\,0\,0\,0 \\ 1\,1\,0\,0\,1\,0\,1\,0\,0\,1\,0\,1\,0\,0\,0\,0 \\ 1\,0\,1\,0\,1\,0\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0 \\ 1\,1\,0\,1\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0 \end{pmatrix}$$

Будем передавать сообщение $m = (11001001)$. Вектор ошибок: $e = (0000100010000000)$. Ошибки были сделаны в 5 и 9 разрядах. Получим:

$$c = m \cdot G' + e = (1001011010\ 111000).$$

Получатель вычисляет:

$$c' = c \cdot P^{-1} = (1000110101\ 110001).$$

Затем этот вектор нужно декодировать с помощью алгоритма декодирования кода Гоппы. В результате декодирования получаем:

$$m' = (10001110).$$

Правильность данного результата можно проверить, умножив $m'G$ и сравнив с cP^{-1}. Разность этих векторов e' имеет вес 2 и совпадает с $e \cdot P^{-1}$.

Чтобы получить m, умножаем m' на S^{-1}:

$$m = m' \cdot S^{-1} = (11001001).$$

Получили передаваемое сообщение.

2.1 Направления криптоанализа кодовых систем

2.1.1 Атака по информационным совокупностям

Одной из лучших атак для криптоанализа кодовых систем является атака по информационным совокупностям (ISD), которая имеет много различных вариаций. Базовый ISD алгоритм был впервые изложен Пранжем в 1962 году [11]:

Пусть $\mathcal{J}$ – некоторое подмножество $\{1, \dots, n\}$, $G_{\mathcal{J}}$- матрица, содержащая только столбцы с индексами из $\mathcal{J}$ в G.

Аналогично, через $e_{\mathcal{J}}$ обозначим вектор длины $|\mathcal{J}|$ с элементами e, которые помещаются в индексы из $\mathcal{J}$.

Алгоритм ISD получения m:

1. Выбрать случайную информационную совокупность $\mathcal{J} \subset \{1, \dots, n\}$.
2. Если на $x_{\mathcal{J}}$ нет ошибок, то: $x_{\mathcal{J}} = m_{\mathcal{J}} G_{\mathcal{J}} + e_{\mathcal{J}}$.
3. Если $w\left(x + x_{\mathcal{J}} G_{\mathcal{J}}^{-1} G\right) = t$, то нет ошибок на $x_{\mathcal{J}}$, а поэтому $w\left(e_{\mathcal{J}}\right) = 0$. В этом случае $m = x_{\mathcal{J}} G_{\mathcal{J}}^{-1}$. Иначе вернуться к шагу 1.

Оценим сложность описанной атаки:

Вероятность того, что выбранные k индексов свободны от ошибок:

$$P = \frac{C_{n-t}^{k}}{C_{n}^{k}} = \frac{C_{n-k}^{t}}{C_{n}^{t}}.$$

Тогда среднее число попыток декодирования τ:

$$\tau = \frac{1}{P} = \frac{C_{n}^{t}}{C_{n-k}^{t}}.$$

Значение τ рассматривают как сложность ISD алгоритма. Для классической криптосистемы Мак-Элиса на основе (1024, 524, 101)-кода Гоппы $\tau \approx 2^{53}$.

Более точная оценка приведена в [12]. Сложность самых эффективных на сегодняшний день ISD-атак для кода с параметрами $k = \big(0.8 + o(1)\big)n,\ n \to \infty$ составляет $\approx (5 + o(1))^t$, где $t = \frac{(0.2+o(1))n}{\lg n}$ – корректирующая способность кода. Улучшить эти результаты могут только квантовые вычисления. Так, алгоритм Гровера даёт квадратичный прирост в скорости, но сложность все ещё остаётся экспоненциальной.

2.1.2 Атака Штерна

Шифротекст $c = m \cdot G' + e$ находится на расстоянии t от кодовых слов кода C'. При этом минимальное расстояние кода $C' = 2t + 1$, поскольку C' эквивалентен коду C с точностью до перестановки. Построим порождающую матрицу A кода $\mathcal{A}$:

$$A = \begin{pmatrix} G' \\ c \end{pmatrix}.$$

Код $\mathcal{A}$ имеет расстояние t, генерируемое единственным вектором веса t. Этот вектор и есть неизвестный вектор ошибки e. Атака состоит в том, чтобы по известной проверочной матрице кода $\mathcal{A}$ найти t таких её столбцов, сумма которых равна нулевому вектору. Существует много способов решения этой задачи, но все они имеют экспоненциальную сложность.

2.2 Параметры классических криптосистем

Оригинальные параметры Мак-Элиса n = 1024, k = 524, t = 50, предложенные в [3], сейчас считаются ненадёжными, поскольку обеспечивают уровень защиты примерно в 50 бит.

Ниже приведены рекомендованные параметры [13] для криптосистем на основе двоичных кодов Гоппы.

Таблица 3.1 – Рекомендованные параметры криптосистемы Мак-Элиса на двоичных кодах Гоппы

Уровень защиты	(n, k, t)	Размер публичного ключа (КБайт)	
		Полный ключ	Систематический
80	(2048, 1751, 27)	438	64
80	(1702, 1219, 45)	254	72
80	(2048, 1696, 32)	424	73
128	(3178, 2384, 68)	925	232
128	(4096, 3604, 41)	1802	217
256	(6944, 5208, 136)	4415	1104

3 Криптосистема на QC-LDPC кодах

Одним из наиболее распространённых на практике и простых по структуре кодов с малой плотностью проверок на чётность является квазициклический МПП-код (QC-LDPC). Как представитель МПП-кодов, он поддаётся алгоритму декодирования с использованием графа Таннера [14]. Однако для QC-LDPC кодов известны и другие, более эффективные алгоритмы декодирования [15], что в совокупности с другими факторами делает рассматриваемые коды надёжной заменой кодов Гоппы в криптосистеме Мак-Элиса.

С использованием QC-LDPC кодов можно значительно сократить размер ключей: знание одной строки каждого циркулянта достаточно для его описания. Идея криптосистемы на QC-LDPC кодах представлена на рисунке 3.1, взятом из [15].

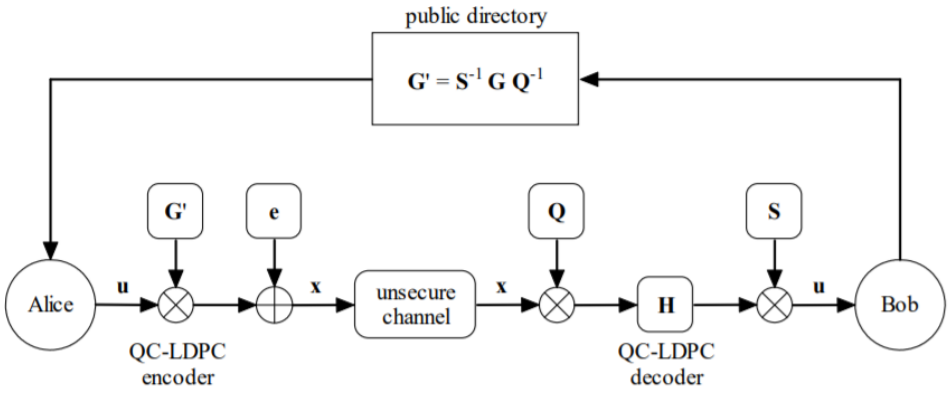

Рисунок 3.1 – Криптосистема Мак-Элиса, построенная на QC-LDPC-кодах

3.1 Конструкция кодов

Пусть $n_0, l \in \mathbb{N}, n_0 > l$. Рассмотрим $I_{p_{ji}} - m \times m$ матрица p_{ij}-кратного циклического сдвига столбцов $m \times m$ единичной матрицы $I, 1 \leq j \leq l, 1 \leq i \leq n_0, 1 \leq p_{ij} \leq m$. Построим $l \times n_0$ матрицу H следующего вида:

$$H = \begin{pmatrix} I_{p_{11}} & \cdots & I_{p_{1n_0}} \\ \cdots & \cdots & \cdots \\ I_{p_{l1}} & \cdots & I_{p_{ln_0}} \end{pmatrix}.$$

Ключевой момент заключается в том, что проверочная матрица H квазициклического МПП-кода полностью определяется размерами сдвигов p_{ij}, поэтому достаточно хранить матрицу

$$\widetilde{H} = \begin{pmatrix} p_{11} & \cdots & p_{1n_0} \\ \cdots & \cdots & \cdots \\ p_{l1} & \cdots & p_{ln_0} \end{pmatrix}.$$

Хранение данной матрицы позволяет достичь m-кратную экономию памяти.

3.2 Генерация ключей

Выбирают длину кода $n = n_0 \cdot p$, размерность $k = k_0 \cdot p$ и избыточность $r = p$, где n_0 – малое натуральное число (например, 3 или 4), $k_0 = n_0 - 1$, p – большое натуральное число порядка нескольких тысяч.

Алгоритм генерации ключей.

Вход: d_v– вес, n_0, $k_0 = n_0 - 1$, p .

Выход: публичный ключ G', секретный ключ (S, G, Q).

1. Построить разреженную проверочную матрицу

$$H = \left(H_1, H_2, H_3, \dots, H_{n_0}\right),$$

где H_i – циркулянт с весом каждой строки d_v.

2. Не умаляя общности, полагаем, что H_{n_0} – не вырождена. Строим порождающую матрицу

$$G = I\ R,$$

где I – единичная матрица размера $(n_0 - 1)p \times (n_0 - 1)p$,

$$R = \left(\left(H_{n_0}^{-1}H_1\right)^T \left(H_{n_0}^{-1}H_2\right)^T \cdots \left(H_{n_0}^{-1}H_{n_0-1}\right)^T\right)^T.$$

3. Построить невырожденную $k \times k$ матрицу S.
4. Построить невырожденную $n \times n$ матрицу Q с весом строки/столбца m.
5. Вычислить $G' = S^{-1} \cdot G \cdot Q^{-1}$.
6. Вернуть публичный ключ G' и секретный ключ – тройку (S, G, Q).

Структура матрицы Q может вызвать эффект «распространения ошибок» (error propagation effect) [16], но это компенсируется высокой корректирующей способностью QC-LDPC-кодов.

3.3 Шифрование и дешифрование

Зашифрование сообщения m длины k проводят следующим образом:

$$x = mG' + e, \text{где } \mathrm{w}(e) \le t' = \frac{t}{d_v},$$

$$t - \text{корректирующая способность кода}$$

Расшифрование шифротекста x:

1. $x' = xQ = mS^{-1}G + eQ$;

2. Применить алгоритм декодирования принятого слова;

3. Получили $m' = mS^{-1}$;

4. Восстановить передаваемое слово домножением на S: $m = m'S$.

3.4 Размер ключей

Публичный ключ рассматриваемой криптосистемы – это двоичная $k_0 \times n_0$ матрица циркулянтов размера $p \times p$. Поскольку каждый циркулянт полностью описывается одной строкой (столбцом), то для его хранения требуется p бит. Таким образом, размер публичного ключа - $n_0\, k_0 p$ бит.

Размеры ключей для $n_0 = 3$ и $n_0 = 4$ приведены на рисунке 3.2, взятом из [15].

TABLE I
PUBLIC KEY SIZE EXPRESSED IN BYTES.

p [bits]	4096	5120	6144	7168	8192	9216	10240	11264	12288	13312	14336	15360	16384
$n_0 = 3$	1024	1280	1536	1792	2048	2304	2560	2816	3072	3328	3584	3840	4096
$n_0 = 4$	1536	1920	2304	2688	3072	3456	3840	4224	4608	4992	5376	5760	6144

Рисунок 3.2 – Длина открытых ключей криптосистемы Мак-Элиса, построенной на QC-LDPC-кодах

3.5 Возможные атаки

Одной из возможных уязвимостей криптосистемы является атака по дуальному коду в том случае, когда этот код содержит слова малого веса. Идея этой атаки заключается в следующем:

Проверочная матрица открытого ключа может быть представлена в виде:

$$H' = HQ^T$$

Каждая строка этой матрицы – это слово дуального кода, и поэтому он имеет по крайней мере $A_w \geq n - k$ кодовых слов веса $w \leq n_0 d_v m$.

Алгоритм Штерна позволяет найти одно из A_w слов веса w в коде длины n_s и размерности k_s с вероятностью

$$P_{w,A_w} \leq A_w \cdot \frac{\binom{w}{g}\binom{n_s - w}{\frac{k_s}{2} - g}}{\binom{n_s}{\frac{k_s}{2}}} \cdot \frac{\binom{w-g}{g}\binom{n_s - \frac{k_s}{2} - w + g}{\frac{k_s}{2} - g}}{\binom{n_s - \frac{k_s}{2}}{\frac{k_s}{2}}} \cdot \frac{\binom{n_s - \frac{k_s}{2} - w + g}{l}}{\binom{n_s - k_s}{l}},$$

где l и g– два параметра, значения которых должны быть оптимизированы как функции общего количества бинарных операций. Таким образом, среднее количество итераций, необходимых для поиска кодового слова с низким весом, составляет $c \geq P_{w,A_w}^{-1}$. Каждая итерация требует:

$$N = \frac{(n_s - k_s)^3}{2} + k_s(n_s - k_s)^2 + 2gl\binom{k_s/2}{g} + \frac{2g(n_s - k_s)\binom{k_s/2}{g}^2}{2^l}$$

Таким образом,

– использование QC-LDPC кодов позволяет значительно сократить размер открытого ключа;

– описанная криптосистема при некоторых условиях остаётся нераскрытой;

– основной проблемой является выбор параметров QC-LDPC кодов, а особенно выбор векторов ошибок ввиду специфического алгоритма декодирования.

4 Криптосистема на QC-MDPC кодах

Единственное отличие QC-LDPC кодов от QC-MDPC состоит в величине веса строки (столбца) w_r (w_c). В LDPC кодах веса строк обычно меньше 10, в то время как веса (n, r, w)-MDPC кодов оцениваются как $O(\sqrt{n\,logn})$.

4.1 Генерация ключей

Опишем построение *(n, r, w)*-QC-MPDC кодов.

Выбирают длину кода $n = n_0 \cdot p$, размерность $r = p$. Причём r – простое.

Проверочная матрица кода имеет вид:

$$H = \left[H_0 | H_1 | \dots | H_{n_0-1}\right],$$

где H_i – $p \times p$ циркулянт с весом каждой строки.

Первая строка матрицы H – случайный вектор длины n с весом w. Остальные $r-1$ строк получаются квази-циклическими сдвигами первой строки. Каждый блок H_i имеет вес строки w_i, так что $w = \sum_{i=0}^{n_0-1} w_i$.

Полагая, что H_{n_0-1} – не вырождена, строим порождающую матрицу кода:

$$G = \left[\begin{array}{c|c} I & \begin{matrix} \left(H_{n_0-1}^{-1} \cdot H_0\right)^T \\ \left(H_{n_0-1}^{-1} \cdot H_1\right)^T \\ \vdots \\ \left(H_{n_0-1}^{-1} \cdot H_{n_0-2}\right)^T \end{matrix} \end{array}\right],$$

где I – единичная матрица размера $(n_0-1)p \times (n_0-1)p$.

Алгоритм генерации ключей.

1. Сгенерировать проверочную матрицу $H \in F_2^{r \times n}$ *(n, r, w)*-QC-MDPC кода, корректирующего ошибки кратности *t*.
2. Построить порождающую матрицу $G \in F_2^{(n-r)\times n}$ кода.

Публичный ключ – G, секретный ключ – H.

4.2 Шифрование и дешифрование

Зашифрование сообщения $m \in F_2^{n-r}$ в шифротекст $x \in F_2^n$:

1. Случайно сгенерировать вектор ошибок $e \in F_2^n$ веса $w(e) \leq t$.
2. Вычислить $x = m\mathrm{G} + e$.

Расшифрование шифротекста $x \in F_2^n$ в открытый текст $m \in F_2^{n-r}$:

Пусть ψ_H – алгоритм декодирования QC-MDPC кода, корректирующего t ошибок.

1. $mG \leftarrow \psi_H(mG + e)$.

2. Извлечь открытый текст m из первых $(n - r)$ позиций mG.

4.3 Размер ключей

Размер публичного ключа составляет всего $(n - r)$ бит. На рисунке 4.1, взятом из [17] предлагаются наиболее подходящие для практического применения параметры криптосистемы.

Level security	n_0	n	r	w	t	QC-MDPC key-size
80	2	9602	4801	90	84	4801
80	3	10779	3593	153	53	7186
80	4	12316	3079	220	42	9237
128	2	19714	9857	142	134	9857
128	3	22299	7433	243	85	14866
128	4	27212	6803	340	68	20409
256	2	65542	32771	274	264	32771
256	3	67593	22531	465	167	45062
256	4	81932	20483	644	137	61449

Рисунок 4.1 – Длина открытых ключей (в битах) криптосистемы Мак-Элиса, построенной на QC-MDPC-кодах

4.4 Практическая стойкость к атакам

Выделяют следующие сценарии криптоанализа описанной схемы:

1) выделение публичного ключа из случайной матрицы (найти одно слово кода $C^{\perp}$ весом w);
2) восстановление секретного ключа (найти r слов кода $C^{\perp}$ весом w);
3) атака декодирования (декодировать t ошибок в $(n, n - r)$-линейном коде).

На сегодняшний день лучшим инструментом для решения этих задач является ISD-атака и различные её модификации. Приведём оценки сложности описанных атак, согласно [17].

Пусть $WF_{isd}(n, r, t)$ – сложность алгоритма декодирования t ошибок (n, r)-бинарного линейного кода.

	MDPC	QC-MDPC
Key distinguishing	$\frac{1}{r}\mathrm{WF}_{\mathrm{isd}}(n, n-r, w)$	$\frac{1}{r}\mathrm{WF}_{\mathrm{isd}}(n, n-r, w)$
Key recovery	$\mathrm{WF}_{\mathrm{isd}}(n, n-r, w)$	$\frac{1}{r}\mathrm{WF}_{\mathrm{isd}}(n, n-r, w)$
Decoding	$\mathrm{WF}_{\mathrm{isd}}(n, r, t)$	$\frac{1}{\sqrt{r}}\mathrm{WF}_{\mathrm{isd}}(n, r, t)$

Рисунок 4.2 – Оценка сложности лучших атак криптосистемы Мак-Элиса, построенной на QC-MDPC-кодах

К примеру, для $n_0 = 2, n = 9602, r = 4801, w = 90, t = 84$ уровень защиты от раскрытия секретного ключа составит $2^{92.70}$ и $2^{87.16}$ для атаки декодированием.

Таким образом, QC-MDPC – многообещающий вариант криптосистемы Мак-Элиса, поскольку он обеспечивает:

– внушительное уменьшение размеров ключей;

– хороший уровень защиты (алгебраическая структура почти отсутствует);

– быстрое шифрование/дешифрование.

5 Декодирование

5.1 Декодирование кодов Гоппы

В данном разделе приведём алгоритм декодирования Паттерсона, который предназначен только для двоичных неприводимых кодов Гоппы над $GF(2^m)$. Кроме того, рассмотрим декодер обобщённых кодов Рида – Соломона в применении к кодам Гоппы над произвольным конечным полем. Для этого отметим, что коды Гоппы можно задавать с помощью обобщённых кодов Рида – Соломона.

Пусть $\propto = (\alpha_0, \alpha_1, \dots, \alpha_{n-1})$, где α_i – различные элементы из $GF(q^m), y = (y_0, y_1, \dots, y_{n-1})$ – ненулевые элементы из $GF(q^m)$. Тогда обобщённый код Рида – Соломона $GRS_k(\propto, y)$ состоит из всех кодовых векторов вида $u = \big(y_0 b(\alpha_0), y_1 b(\alpha_1), \dots, y_{n-1} b(\alpha_{n-1})\big)$, где $b(x)$ – информационные многочлены над полем $GF(q^m)$ степени не выше $k - 1$.

Теперь приведём утверждение из [18].

Теорема. Код $\Gamma(L, G)$ представляет собой ограничение кода $GRS_{n-r}(L, y)$ на подполе $F = GF(q)$, т.е. $\Gamma(L, G) = GRS_{n-r}(L, y) \cap F^n$, где $r = \deg G(x), y = (y_0, y_1, \dots, y_{n-1})$,

$$y_i = G(\alpha_i) \prod_{j \neq i} \frac{1}{\alpha_i - \alpha_j}, i = 0, 1, \dots, n-1.$$

Проверочная матрица кода $GRS_{n-r}(L, y)$, который задаёт код $\Gamma(L, G)$, имеет вид

$$\begin{pmatrix} 1 & \dots & 1 \\ \alpha_1 & \dots & \alpha_n \\ \dots & \dots & \dots \\ \alpha_1^{r-1} & \dots & \alpha_n^{r-1} \end{pmatrix} \begin{pmatrix} G^{-1}(\alpha_1) & \dots & \dots \\ \dots & G^{-1}(\alpha_2) & \dots \\ \dots & \dots & \dots \\ \dots & \dots & G^{-1}(\alpha_n) \end{pmatrix}.$$

Таким образом, код $\Gamma(L, G)$ можно задать с помощью обобщённых кодов Рида – Соломона. Воспользуемся этим при формулировке алгоритмов декодирования кодов Гоппы над любым полем.

5.1.1 Алгоритм декодирования Паттерсона

Вход: вектор v.

Выход: исходный кодовый вектор u, если произошло $t \le r$ ошибок, $r = \deg G(x)$.

Шаг 1. Вычислить синдромный многочлен

$$S(x) \equiv \sum_{i=0}^{n-1} \frac{v_i}{x - \alpha_i} \big(mod\ G(x)\big).$$

Если $S(x) = 0$, то алгоритм завершается и возвращается v.

Шаг 2. Вычислить $T(x) \equiv S^{-1}(x)\big(mod\ G(x)\big)$ (то есть находится решение сравнения $S(x)T(x) \equiv 1\ \big(mod\ G(x)\big)$), используя обобщённый алгоритм Евклида. Если $T(x) = x$, то полагается $\sigma(x) = x$ и происходит переход в шаг 5.

Шаг 3. Вычислить многочлен

$$p(x) \equiv \sqrt{x + T(x)}\big(mod\ G(x)\big).$$

Шаг 4. Положить $r_{-1}(x) = G(x),\ r_0(x) = p(x),\ v_{-1}(x) = 0,\ v_0(x) = 1$. Произвести последовательность вычислений обобщённого алгоритма Евклида $(i \ge 1)$

$$r_{i-2}(x) = r_{i-1}(x)q_{i-1}(x) + r_i(x), \qquad v_i(x) = v_{i-2}(x) - v_{i-1}(x)q_{i-1}(x)$$

до тех пор, пока для некоторого j не будут выполнены неравенства

$$\deg r_{j-1}(x) > \left[\frac{r}{2}\right], \qquad \deg r_j(x) \le \left[\frac{r}{2}\right].$$

В этом случае $a(x) = r_j(x),\ b(x) = v_j(x),\ \sigma(x) = r_j^2(x) + xv_j^2(x)$ (с точностью до константы).

Шаг 5. Вычислить корни многочлена $\sigma(x)$, равные локаторам ошибок: $X_1 = \alpha_{i_1}, \cdots, X_t = \alpha_{i_t}$, где $t = \deg\sigma(x)$. После этого в векторе v исправить ошибки на позициях $i_1, \cdots, i_t$.

Пример. Пусть принят вектор $v = (0\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1)$, в котором не более двух ошибок. Имеем дело с $L = GF(2^4) = \{0, 1, \alpha, \alpha^2, \dots, \alpha^{14}\}, G(x) = x^2 + x + \alpha^3$.

1. Вычислим синдромный многочлен

$$S(x) \equiv \sum_{i=0}^{15} \frac{v_i}{x - \alpha_i} \equiv \frac{1}{x-1} + \frac{1}{x-\alpha} + \frac{1}{x-\alpha^2} + \cdots + \frac{1}{x-\alpha^{14}}$$
$$\equiv \alpha^{14} + \alpha^{12}x \ (mod\ G(x)).$$

2. Вычислим $T(x) \equiv (\alpha^{14} + \alpha^{12}x)^{-1} \equiv \alpha^{14} + \alpha^6 x \ (mod\ G(x))$.
3. Вычислим

$$s(x) \equiv x^{128} \equiv \alpha^9 + x \ (mod\ G(x)).$$
$$p(x) \equiv \sqrt{T(x) + x} \equiv \sqrt{\alpha^{14} + \alpha^{13}x} \equiv (\alpha^{14})^8 + (\alpha^{13})^8(\alpha^9 + x)$$
$$\equiv \alpha^{11} + \alpha^{14}x \ (mod\ G(x)).$$

4. Полагаем $r_{-1}(x) = G(x), r_0(x) = p(x), v_{-1}(x) = 0,\ v_0(x) = 1$. При *j = 0* видно, что $\deg r_{-1}(x) = 2 > \frac{r}{2}, \deg r_0(x) = 1 < \frac{r}{2}$. Поэтому с точностью до константы

$$\sigma(x) = r_0^2(x) + xv_0^2(x) = \alpha^7 + x + \alpha^{13}x^2.$$

5. Корнями многочлена $\sigma(x)$ являются $X_1 = \alpha^3 = \alpha_4,\ X_2 = \alpha^6 = \alpha_7$, поэтому ошибки произошли на 4 и 7 позициях. Следовательно, исходный кодовый вектор равен $u = (0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1)$.

5.1.2 Декодирование кодов Гоппы на основе алгоритма Гао

Данный алгоритм относится к бессиндромному декодированию и впервые был сформулирован в [19].

Вход: вектор v.

Выход: исходный кодовый вектор u, если произошло не более t ошибок, если $r \geq 2t,\ r = \deg G(x),\ u \in \Gamma(L, G) \subseteq GRS_{n-r}(L, y)$.

Шаг 1 (Интерполяция). Построить интерполяционный многочлен $f(x)$, для которого $f(\alpha_i) = y_i^{-1}v_i, i = 0, \dots, n-1$.

Шаг 2 (Незаконченный обобщённый алгоритм Евклида). Пусть $r_{-1}(x) = m(x), r_0(x) = f(x), v_{-1}(x) = 0, v_0(x) = 1$. Произвести последовательность действий обобщённого алгоритма Евклида ($i \geq 1$)

$$r_{i-2}(x) = r_{i-1}(x)q_{i-1}(x) + r_i(x), \qquad v_i(x) = v_{i-2}(x) - v_{i-1}(x)q_{i-1}(x)$$

до тех пор, пока для некоторого j не будут выполнены неравенства

$$\deg r_{j-1}(x) \geq \frac{n+\tilde{k}}{2}, \qquad \deg r_j(x) \leq \frac{n+\tilde{k}}{2},$$

где $\tilde{k} = n - r$ – размерность кода $GRS_{n-r}(L, y)$.

При этом на каждом шаге должно выполняться сравнение $v_i(x)f(x) \equiv r_i(x)\big(mod\, m(x)\big)$.

Шаг 3 (Деление). Информационный многочлен равен $b(x) = r_j(x)/v_j(x)$.

Шаг 4. Вычислить кодовый вектор u с помощью кодирования информационного многочлена $b(x)$ для кода $GRS_{\tilde{k}}(L, y)$:

$$u = \big(y_0 b(\alpha_0), y_1 b(\alpha_1), \dots, y_{n-1} b(\alpha_{n-1})\big)$$

5.2 Декодирование QC-LDPC и QC-MDPC кодов

LDPC и MDPC коды показывают хорошую корректирующую способность в таких каналах как АБГШ (канал с аддитивным гауссовским шумом), однако в криптосистеме Мак-Элиса используется другая модель канала: в кодовое слово вносится фиксированное число ошибок, и каждый бит с ошибкой инвертируется. Поэтому наиболее подходящая модель канала в данном случае – двоичный симметричный канал, за тем лишь исключением, что в кодовое слово вносится ровно t случайных ошибок (в классическом BSC число ошибок колеблется около t).

В связи с этим нас в основном интересуют алгоритмы «жёсткого» декодирования, главным представителем которых является bit-flipping алгоритм, предложенный Галлагером в [14]. Кроме того, рассмотрим «мягкое» декодирование в применении к нашей модели как некое улучшение, демонстрирующее лучшую корректирующую способность кода.

5.2.1 Bit-flipping-декодер

Вход: LDPC-код C (или MDPC-код) с параметрами (n, k, d), заданный проверочной матрицей

$$H = \begin{pmatrix} h_{11} & \dots & h_{1n} \\ \dots & \dots & \dots \\ \pi_{(n-k)1} & \dots & \pi_{(n-k)n} \end{pmatrix}.$$

Вектор $v = u + e, u \in C(\subset F_2^n), e\ (\in F_2^n)$ — вектор ошибок; p — количество итераций алгоритма; T — пороговое значение.

Выход: кодовый вектор $u \in C(\subset F_2^n)$.

Шаг 1. Положим счетчик r равным нулю.

Шаг 2. Вычислим синдром $s = uH^T$. Если $s = (0, \dots, 0)$ или $r = p$, то переходим на шаг 5.

Шаг 3. Выделим из вектора $s = (s_1, \dots, s_{n-k})$ единичные координаты, т. е. $s_i = 1, i = \overline{1, (n-k)}$. Составим множество $L = \{i | s_i = 1\}$. Вычислим $\overline{h'} = (h_1', \dots, h_n')$, где

$$h_l' = \sum_{i \in L} h_{il}.$$

Величины $h_{il}, l = 1, \dots, n$ следует полагать неотрицательными целыми числами.

Шаг 4. В векторе $\overline{h'} = (h_1', \dots, h_n')$ находим все элементы $h_1' > T$. Среди них выбираем случайный h_l'и инвертируем бит v_l вектора v. Добавляем к счётчику r единицу и переходим на шаг 2.

Шаг 5. $u := v$.

Замечание 1. Входной параметр p задаёт максимальное количество итераций алгоритма со 2-го по 4-й шаги, но декодер может восстановить кодовое слово за меньшее число итераций.

Замечание 2. При выборе параметра T нужно руководствоваться следующими соображениями. Если известен параметр d используемого (N, K, d)-кода C, то по нему можно вычислить t — число гарантированно исправляемых ошибок, и тогда количество итераций декодера ограничивается этим значением:

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

Если параметр d неизвестен, то его можно оценить с помощью границы Синглтона:

$$d \leq n - k + 1$$

и, используя (3), получить $p = \left\lfloor \frac{n-k}{2} \right\rfloor$.

Замечание 3. Структура декодера такова, что восстановление корректного кодового слова не гарантируется, даже если в зашумленном слове $v = u + e$ возникло не более t ошибок.

Замечание 4. Корректирующую способность BF-декодера может ухудшить неудачный выбор порога T. При его большом значении на шаге 4 декодера в векторе $\overline{h'}$ может не найтись координаты, превосходящей порог T, следовательно, не будут исправлены ошибочные биты. При выборе малого значения T на шаге 4 BF-декодера в векторе $\overline{h'}$ может появиться несколько координат, значение которых превышает порог. Среди них могут быть и координаты, не содержащие ошибку. Таким образом, выбор параметра T может в значительной мере повлиять на качество декодирования.

В связи с указанными замечаниями целесообразно внести изменения в BF-декодер, которые позволят определять величину порога динамически, в зависимости от степени повреждения кодового вектора.

5.2.2 Модификация bit-flipping

Вход: LDPC-код C (или MDPC-код) с параметрами (n, k, d), заданный проверочной матрицей

$$H = \begin{pmatrix} h_{11} & \dots & h_{1n} \\ \dots & \dots & \dots \\ \pi_{(n-k)1} & \dots & \pi_{(n-k)n} \end{pmatrix}.$$

Вектор $v = u + e, u \in C(\subset F_2^n), e\ (\in F_2^n)$ — вектор ошибок; p — количество итераций алгоритма; T — некоторое пороговое значение, выбранное заранее.

Выход: кодовый вектор $u \in C(\subset F_2^n)$.

Шаг 1. Положим счётчик r равным нулю.

Шаг 2. Вычислим синдром $s = uH^T$. Если $s = (0, \dots, 0)$ или $r = p$, то переходим на шаг 7.

Шаг 3. Выделим из вектора $s = (s_1, \dots, s_{n-k})$ единичные координаты, т. е. $s_i = 1, i = \overline{1, (n-k)}$. Составим множество $L = \{i | s_i = 1\}$. Вычислим $\overline{h'} = (h_1', \dots, h_n')$, где

$$h_l' = \sum_{i \in L} h_{il}.$$

Величины $h_{il}, l = 1, \dots, n$ следует полагать неотрицательными целыми числами.

Шаг 4. Инициализируем значение порога $T = \max\limits_{l=1..N}(h_l') - 1$.

Шаг 5. Если $T \geq 0$:

Выберем произвольный элемент h_q' вектора $\overline{h'}$ — такой, что $h_l' > T$.

Инвертируем бит v_q.

Шаг 6. Добавим к счётчику r единицу и перейдём на шаг 2.

Шаг 7. $u := v$.

Модифицированный алгоритм в целом выполняет меньше итераций, чем BF-декодер, т. к. на шаге 4 порог выбирается динамически. Следовательно, декодер не выполняет бесполезные итерации, на которых не изменяются биты вектора v. Значение порога в модифицированном декодере зависит от числа ошибок, повредивших кодовое слово, и сразу устанавливается таким, что в зашумленное кодовое слово v гарантированно вносятся изменения.

Проведённые в [22] имитационные эксперименты демонстрируют лучшую корректирующую способность модифицированного декодера по отношению к оригинальному.

5.2.3 LLR-SPA

Приведём Log-Likelihood Ratios Sum-Product алгоритм, который является оптимальным алгоритмом «мягкого» декодирования LDPC кодов. Для этого введём следующие обозначения:

$A(k)$ – множество вершин символов v_i, соединённых с узлом проверки c_k (помним, что проверочная матрица H кода эквивалентна графу Таннера).

$B(i)$ – множество вершин проверок c_k, соединённых с узлом символов v_i.

$q_{i\to k}(x), x \in \{0,1\}$ – это информация, которую вершина v_i передаёт проверке c_k.

$r_{k\to i}(x), x \in \{0,1\}$ – это информация, которую вершина c_k передаёт проверке v_i .

Сообщения, отправленные от узлов-переменных к узлам-проверок:

$$\Gamma_{i\to k}(x_i) = ln\left[\frac{q_{i\to k}(0)}{q_{i\to k}(1)}\right]$$

Сообщения, отправленные от узлов-проверок к узлам-переменным:

$$\Lambda_{k\to i}(x_i) = ln\left[\frac{r_{k\to i}\,(0)}{r_{k\to i}(1)}\right]$$

Логарифм отношения правдоподобия (Log-Likelihood Ratio) от i-ого бита кодового слова:

$$LLR(x_i) = ln\left[\frac{P(x_i = 0|y_i = y)}{P(x_i = 1|y_i = y)}\right],$$

где $P(x_i = x|y_i = y), x \in \{0,1\}$ – вероятность, что передавали $x_i = x$, при условии, что приняли $y_i = y$.

Как было сказано выше, криптосистема Мак-Элиса представляет из себя двоичный симметричный канал (с вероятностью ошибки $p = \frac{t}{n}$) в том смысле, что на выходе из канала нет вероятностей в явном виде (в случае с AWGN-каналом, к примеру, мы можем посчитать вероятности как интеграл плотности распределения нормальной случайной величины). Поэтому для работы с LLR необходимо провести следующие приготовления:

$$P(y_i = 0|x_i = 1) = p$$
$$P(y_i = 1|x_i = 1) = 1 - p$$
$$P(y_i = 0|x_i = 0) = 1 - p$$
$$P(y_i = 1|x_i = 0) = p$$

Таким образом,

$$LLR(x_i|y_i = y) = \ln\left(\frac{1-p}{p}\right) = \ln\left(\frac{n-t}{t}\right)$$

$$LLR(x_i|y_i = 1) = \ln\left(\frac{p}{1-p}\right) = \ln\left(\frac{t}{n-t}\right)$$

Алгоритм декодирования.

Шаг 1 (Инициализация). $\forall i, k$, таких что v_i и c_k соединены ребром в графе Таннера, установить:

$$\Gamma_{i\to k}(x_i) = LLR(x_i)$$
$$\Lambda_{k\to i}(x_i) = 0$$

Шаг 2 (От проверки к символу). Сообщения, отправленные от узлов проверок к узлам символов, вычисляются по формуле:

$$\Lambda_{k\to i}(x_i) = 2\cdot tanh^{-1}\left\{\prod_{j\in A(k)\setminus i}\tanh\left[\frac{1}{2}\Gamma_{i\to k}(x_i)\right]\right\}$$

Шаг 3 (От символа к проверке). Сообщения, отправленные от узлов символов к узлам проверок, вычисляются по формуле:

$$\Gamma_{i\to k}(x_i) = LLR(x_i) + \sum_{j\in B(i)\setminus k}\Lambda_{j\to i}(x_i)$$

На этом же шаге вычисляется:

$$\Gamma_i(x_i) = LLR(x_i) + \sum_{j\in B(i)}\Lambda_{j\to i}(x_i)$$

Шаг 4 (Принятие решения). По следующему правилу вычисляется оценка $\widehat{x_i}$ кодового слова x_i:

$$\widehat{x_i} = \begin{cases} 0, \text{если } \Gamma_i(x_i) \geq 0 \\ 1, \text{если } \Gamma_i(x_i) < 0 \end{cases}$$

После каждой прогонки алгоритма проверяем равенство синдрома нулю. В случае выполнения этого условия декодирование останавливается, и полученная оценка кодового слова возвращается как результат. Иначе – возврат на шаг 2 с обновлёнными $\Gamma_{i\to k}(x_i)$. Для завершения алгоритма также задают условия на максимальное число итераций.

В [20] проведено компьютерное моделирование, которое позволяет оценить корректирующую способность описанного декодера. Результаты симуляции приведены на рисунке 5.2.

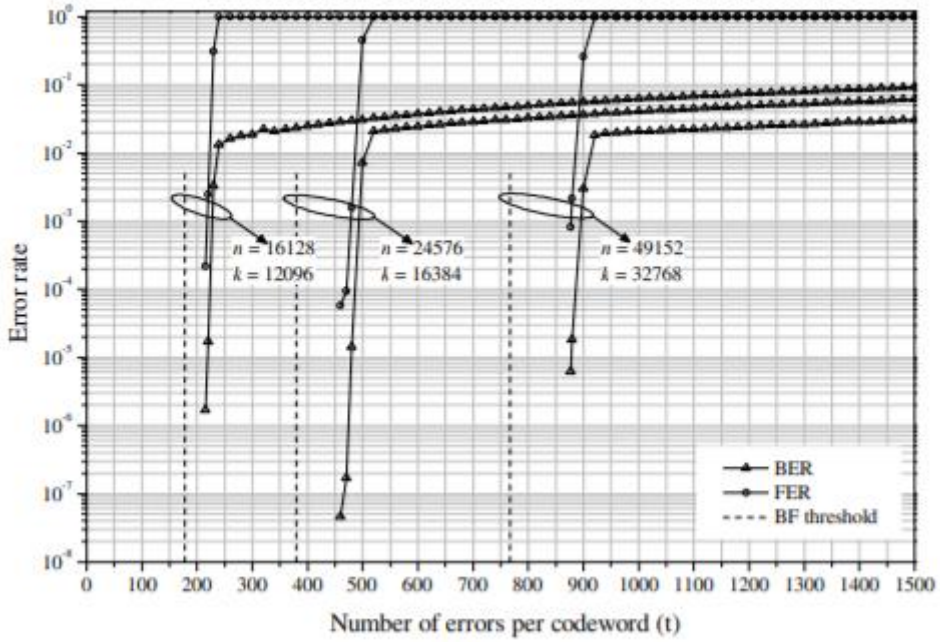

Рисунок 5.2 – Корректирующая способность LLR-SPA декодера как функция от числа ошибок в кодовом слове для трёх QC-LDPC-кодов

Согласно полученным результатам, LLR-SPA декодер имеет экстремально низкую вероятность ошибки и демонстрирует преимущество над BF-декодером, рассмотренным в предыдущих пунктах.

Заключение

Оригинальная криптосистема Мак-Элиса, разработанная в 1978 году, остаётся криптостойкой и в настоящие дни. Она представляет большой интерес исследователей и является кандидатом для пост-квантовой криптографии в связи с устойчивостью к атаке с использованием квантового компьютера. Основным её недостатком является большой размер ключей, поэтому в рамках данной работы существенное внимание уделялось возможным модификациям, позволяющим уменьшить ключ без потери криптостойкости.

В основе этих модификаций лежат коды Галлагера с малой плотностью проверок на чётность. Они позволяют сократить ключевое пространство и в некоторых случаях уменьшить сложность декодирования.

При этом, применяя декодеры LDPC и MDPC кодов, мы столкнулись с тем, что они показывают хорошую корректирующую способность в таких каналах как АБГШ (канал с аддитивным гауссовским шумом), однако в криптосистеме Мак-Элиса используется другая модель канала: в кодовое слово вносится фиксированное число ошибок, и каждый бит с ошибкой инвертируется. Поэтому наиболее подходящая модель в данном случае – двоичный симметричный канал, за тем лишь исключением, что в кодовое слово вносится ровно t случайных ошибок (в классическом BSC число ошибок колеблется около t). Этим обусловлена специфика приведённых в работе алгоритмов декодирования.

Кроме рассмотренных криптосистем существует также много других. В этой области непрерывно появляются новые исследования, ведутся активные обсуждения и публикуются научные статьи. Это подтверждает актуальность предложенной темы и намечает перспективы развития в данной предметной области.

Основным результатом настоящей работы является структурирование и подробный анализ ключевых аспектов кодовой криптографии, обзор криптосистем Мак-Элиса с использованием двоичных кодов Гоппы, QC-LDPC и QC-MDPC кодов.

Список использованных источников

1. Hellman M. Diffie W. New directions in cryptography// IEEE transactions on Information Theory – 1976. – P. 644 – 654.

2. A. Shamir. A polynomial-time algorythm for breaking the basic Merkle-Hellman cryptosystem// IEEE Transactions on Information Theory – 1978. – Vol. 24, No. 5 – P. 699 – 704.

3. R. McEliece. A public-key cryptosystem based on algebraic Coding Theory// DSN Progress Report – 1978. – P. 114 – 116.

4. V. Sidelnikov. A public-key cryptosystem based on ReedMuller codes// Discrete Math. Appl. – 1994. – Vol. 4, No. 3 – P. 191 – 207.

5. Niederreiter H. Knapsack-Type Cryprosystem and Algebraic Coding Theory// Problems of Control and Information Theory – 1986. – Vol. 15, No. 2 – P. 159 – 166.

6. Chiaraluce F. Baldi M. Cryproanalysis of new instance of McEliece cryptosystem based on QC-LDPC codes// IEEE Int. Symposium Inf. Theory – 2007. – P. 2591 – 2595.

7. Shokrollahi A. Minder L. Cryptoanalysis of Sidelnikov cryptosystem// Advances in Cryptology - EUROCRYPT 2007 – 2007. – Vol. 4515 – P. 347 – 360.

8. Bardet M., Chaulet J., Dragoi V., Otmani A., Tillich J.-P. Cryptoanalysis of McEliece Public Key Cryptosystem Based on Polar Codes// Post-Quantum Cryptography: 7th International Workshop, PQCrypto 2016, Fukuoka, Japan – 2016. – P. 118 – 143.

9. Гоппа В.Д. Новый класс линейных корректирующих кодов// Проблемы передачи информации – 1970. – Т. 6, N 3 – С. 24 – 30.

10. Patterson N. The algebraic decoding of Goppa codes// IEEE Transactions on Information Theory – 1975. – Vol. 21, No. 2 – P. 203 – 207.

11. P. Eugene. The use of information sets in decoding cyclic codes// IRE Transactions on Information Theory – 1962.

12. Hamdaoui Y., Sendrier N. A non asymptotic analysis of information set decoding// IACR Cryptology ePrint Archive – 2013.

13. Repka M., Zajac P. OVERVIEW OF THE MCELIECE CRYPTOSYSTEM AND ITS SECURITY// Tatra Mountains Math. Publications – 2014. – No. 60 – P. 57 – 75.

14. Галлагер Р. Дж. Коды с малой плотностью проверок на четность// Мир – 1966. – С. 90.

15. Baldi M., Bianchi M., Chiaraluce F. Security and complexity of the McEliece

cryptosystem based on QC-LDPC codes// IET Information Security – 2013.

16. Peters C. Information-set decoding for linear codes over Fq// Post-Quantum Cryptography/ ser. Lecture Notes in Computer Science – 2010. – No. 6061, P. 81 – 94.

17. Misoczki R., Tillich J-P., Sendrier N., Barreto P. MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes – 2012.

18. Блейхут Р. Теория и практика кодов, контролирующих ошибки – М: Мир, 1986 . – 576 с.

19. Gao S., A new algorythm for decoding Reed-Solomon codes// Communications, Information and Network Security – 2003. – No. 6061, P. 58 – 68.

20. Baldi. M. QC-LDPC Code-Based Cryptography. – Research Gate, 2014 . – 137 с.

21. Yamada A., Eaton E., Kalach K., Lafrance P., Parent A. QC-MDPC KEM: A Key Encapsulation Mechanism Based on the QC-MDPC McEliece Encryption Scheme// NIST Submission – 2017. – URL: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions. (дата обращения 2021-03-01).

22. Гурский С.С., Могилевская Н.С. О модификации декодера bit-flipping кодов с низкой плотностью проверки на чётность // Advanced Engeneering Research – 2021. – Т. 21, N 1 – С. 96 – 104.