ПРИКЛАДНАЯ ТЕОРИЯ КОДИРОВАНИЯ

УДК 004.056.5

О ПРИМЕНИМОСТИ АЛГЕБРОГЕОМЕТРИЧЕСКИХ КОДОВ L-КОНСТРУКЦИИ КАК КОДОВ ЗАЩИТЫ ОТ КОПИРОВАНИЯ

Д. В. Загуменнов*, В. В. Мкртичян**

**Южный федеральный университет, г. Ростов-на-Дону, Россия*
***ФГАНУ НИИ «Спецвузавтоматика», г. Ростов-на-Дону, Россия*

Схемы широковещательного шифрования используются для защиты легально тиражируемой цифровой продукции от несанкционированного копирования. В схемах распространитель тиражирует данные свободно в зашифрованном виде, а для расшифрования выдаёт каждому легальному пользователю уникальный набор ключей, по которому можно однозначно определить пользователя, его получившего, и далее определить источник несанкционированного распространения. Для организации схем широковещательного шифрования используются методы, основанные на использовании помехоустойчивых кодов с большим кодовым расстоянием и быстрых алгоритмов списочного декодирования. В работе рассматривается возможность использования в этих схемах алгеброгеометрических кодов (АГ-кодов) L-конструкции и списочных декодеров Судана — Гурусвами. Рассмотрена проблема построения таких кодов. В ходе исследования получены достаточные условия применимости АГ-кодов. Представлен алгоритм построения одноточечного АГ-кода, применимого в схемах широковещательного шифрования, обоснована его корректность, приведён пример работы.

Ключевые слова: *помехоустойчивое кодирование, схемы специального широковещательного шифрования, алгеброгеометрические коды.*

DOI 10.17223/20710410/44/6

ON APPLICATION OF ALGEBRAIC GEOMETRY CODES OF L-CONSTRUCTION IN COPY PROTECTION

D. V. Zagumennov*, V. V. Mkrtichyan**

**Southern Federal University, Rostov-on-Don, Russia*
***FGANU NII «Specvuzavtomatika», Rostov-on-Don, Russia*

E-mail: zagumionnov.denis@yandex.ru

Traceability schemes which are applied to the broadcast encryption can prevent unauthorized parties from accessing the distributed data. In a traceability scheme, a distributor encrypts the data and gives each authorized user a unique key suit to decrypt the data. This suit uniquely identifies the recipient and therefore allows the tracing of the source of an unauthorized redistribution. A widely used approach to the constructing good traceability scheme is the use of error-correcting codes with a suitable minimum distance and efficient decoding algorithms. The paper deals with the usage of algebraic geometry codes (AG-codes) of L-construction and Sudan — Guruswami

list decoding algorithms in these schemes. We suggest the problem of constructing traceability AG-codes and obtain sufficient conditions for applying them. Let $C \subset \mathbb{F}_q^n$ be the algebraic geometry code constructed using curve of genus g and divisor of degree α. Firstly, if $c < \sqrt{n/\alpha}$, then C is a traceability code when the number of attackers is a maximum of c. Secondly, if $\alpha \geqslant \log_q N + g - 1$, then C can be used to build traceability schemes maintaining N users. Finally, we obtain several cumbersome bounds on the number of intruders within which it is possible to use Sudan — Guruswami hard- and soft- decision list decoding algorithms for tracing the unauthorized redistribution source. Based on these derived conditions and some other lemmas, the algorithm for suitable one-point AG-code construction is presented. The algorithm can be polynomially reduced to the Riemann — Roch space basis construction problem. Much attention is given to the algorithm validity and its sample execution. Besides, the paper gives a brief description of AG-codes and Sudan — Guruswami hard- and soft- decision list decoding algorithms.

Keywords: *error-correcting codes, traceability schemes, algebraic geometry codes.*

Введение

Рассматривается перспективный способ защиты легально тиражируемой цифровой продукции от несанкционированного копирования, называемый схемой специального широковещательного шифрования (ССШШ) [1]. В ССШШ данные тиражируются свободно в зашифрованном виде, а каждому легальному пользователю выдаётся уникальный набор ключей. В случае обнаружения нелегального использования этого набора его владелец может быть идентифицирован контроллером. В ССШШ допускаются атаки следующего вида: легальные пользователи объединяются в коалиции с целью комбинирования ключей из своих наборов и конструирования новых с целью последующего несанкционированного расшифрования данных. Для борьбы с подобными атаками в [1 – 5] предложен метод обнаружения членов коалиций, основанный на использовании некоторых классов линейных кодов и являющийся эффективным при применении быстрых алгоритмов списочного декодирования. Под эффективностью понимается существование алгоритма, позволяющего определить пользователей из коалиции злоумышленников за полиномиальное (от длины кода) время работы.

Интересной представляется задача определения, являются ли алгеброгеометрические коды (АГ-коды) L-конструкции [6] и списочные методы их декодирования [7, 8] пригодными для эффективного использования в ССШШ. Отметим актуальность рассмотрения АГ-кодов. Во-первых, АГ-коды обобщают многие другие классы линейных помехоустойчивых кодов [9, п. 4.3.1–4.3.3]. Во-вторых, АГ-коды обладают большим минимальным кодовым расстоянием [9, теорема 4.1.1], что может положительно сказаться на выполнении достаточного условия возможности использования линейных кодов в ССШШ (см. п. 1.2). В-третьих, длина АГ-кодов в общем случае не ограничена мощностью поля, в отличие от кодов Рида — Соломона и Рида — Маллера, использование которых исследовано в [4, 5] и [8, Tracing Algorithm, B. Example], а ограничена лишь мощностью кривой, на которой строится код и которая может быть существенно больше, чем мощность поля [9, теорема 3.1.25]. Это может позволить строить более эффективные относительно затрат памяти ССШШ (см. п. 2.4).

Основное внимание в работе уделено формализации построения АГ-кодов, которые можно использовать в ССШШ. Построенные согласно полученным результатам и составленным рекомендациям коды далее можно использовать в различных вариа-

циях схем, например в схеме из [8], основанной на использовании мягкого списочного декодера Судана — Гурусвами.

В работе представлено доказательство утверждения о верхней границе для максимальной мощности коалиции злоумышленников, в пределах которой АГ-коды возможно применять в ССШШ. Получены утверждения о таких границах, в пределах которых возможно применение в ССШШ алгоритмов списочного декодирования [7, 8]. Исследована связь между этими утверждениями. На основе этих границ построен концептуальный алгоритм построения АГ-кода L-конструкции, применимого в ССШШ.

1. Предварительные сведения

Пусть $\mathbb{F}_q$ — конечное поле мощности q, $C \subseteq \mathbb{F}_q^n$ — линейный код, n — длина, k — размерность, d — минимальное кодовое расстояние кода C. Коды с параметрами n, k, d, определённые над полем $\mathbb{F}_q$, будем называть $[n, k, d]_q$-кодами. Для $x, y \in \mathbb{F}_q^n$ будем обозначать $\mathrm{d}(x, y) = |\{i \in \{1, \ldots, n\} : x_i \neq y_i\}|$.

1.1. Схемы специального широковещательного шифрования

В [10] представлены способы защиты легально тиражируемой цифровой продукции от несанкционированного копирования, называемые схемами широковещательного шифрования. В [11] эти подходы описаны и проанализированы более подробно.

ССШШ [1] — это усовершенствованная схема 4.1 из [11], основанная на использовании некоторых классов линейных кодов. Схемы из [11] будем называть классическими схемами широковещательного шифрования (КСШШ). В настоящей работе рассматривается именно ССШШ.

В ССШШ в качестве открытого ключа используется матрица $M = (E_{k_{ij}}(s_i))_{j=1,\ldots,q}^{i=1,\ldots,n}$, где $q = p^r$; p простое; $r, n \in \mathbb{N}$; s_i — разделённые части некоторого секрета s. Каждому пользователю схемы выдаётся уникальный набор ключей $(k_{1,i_1}, \ldots, k_{n,i_n})$ и вектор-номер $(i_1, \ldots, i_n)$, $i_k \in \{1, \ldots, q\}$, $k = 1, \ldots, n$, необходимый для того, чтобы пользователь знал, какие именно части матрицы M он может расшифровать на своих ключах. Проведя расшифрование и получив все части s_i секрета s, пользователь может получить доступ к распространяемой цифровой продукции.

В схемах широковещательного шифрования допускаются атаки следующего вида: легальные пользователи объединяются в коалиции, комбинируют свои вектор-номера и соответственно наборы ключей, получая таким образом новый «пиратский» набор, который может быть использован с целью несанкционированного распространения и последующего расшифрования данных.

Для этого в [1] в качестве вектор-номеров предложено использовать векторы некоторого кода C над полем $\mathbb{F}_q$. В [2, 3] описаны классы кодов, используя которые, возможно бороться с подобными коалиционными атаками, в частности класс c-ТА-кодов.

Далее будем использовать следующие обозначения: N — число легальных пользователей; c — максимальная мощность коалиции злоумышленников; n — длина используемого в ССШШ линейного кода; q — мощность поля, над которым построен используемый код.

Определение 1. ССШШ с N легальными пользователями, корректно функционирующие при атаках коалиции злоумышленников мощности не более c, будем обозначать (N, c)-ССШШ.

В [11] также представлены КСШШ, в которых существует вероятность объявить невиновного пользователя участником коалиции злоумышленников (схемы с секретом, п. 4.3 и 4.4). Далее будем сравнивать ССШШ только с теми КСШШ, в которых,

как и в ССШШ, гарантировано выявление хотя бы одного пользователя из коалиции злоумышленников (открытые схемы, п. 4.1 и 4.2)

В таблице приведено сравнение (теоремы 1 и 2 из [11]) ССШШ с этими КСШШ, а также с тривиальной схемой, когда секрет шифруется на отдельных ключах и рассылается в таком виде всем пользователям. Под длиной ключа понимается длина всей ключевой информации, хранящейся у пользователя, то есть суммарная длина вектор-номера и набора ключей. Под затратами памяти понимается размер матрицы M, под сложностью расшифрования — минимальное число операций расшифрования, которые предстоит выполнить пользователю для получения распространяемой информации в расшифрованном виде.

Через R обозначим отношение числа бит, которые необходимо использовать для представления в памяти элемента поля $\mathbb{F}_q$, к числу бит, которые необходимо использовать для представления в памяти ключа выбранной шифросистемы. Будем полагать, что части s_i разделённого секрета s имеют ту же длину, что и s. Такую схему разделения секрета легко организовать, если секрет s выбран из некоторой аддитивной группы G. В этом случае s можно разделить следующим образом: $s = \bigoplus_{i=1}^{n} s_i$.

Сравнение ССШШ с КСШШ и тривиальной схемой

Параметр	Длина ключевой информации	Затраты памяти	Сложность расшифрования
Единица измерения	Длина ключа в битах	Длина зашифованного секрета в битах	Число операций расшифрования
Тривиальная схема	1	N	1
Открытая одноуровневая схема	$4c^2 \log N$	$8c^4 \log N$	$4c^2 \log N$
Открытая двухуровневая схема	$8/3 \cdot c^2 \log^2 c \log(\mathrm{e}N/c))$	$32/3 \cdot \mathrm{e}c^3 \log^4 c \log(\mathrm{e}N/c)$	$8/3 \cdot c^2 \log^2 c \log(\mathrm{e}N/c))$
ССШШ	$n(1+R)$	nq	n

Единицей измерения в первом столбце выбрана длина ключа шифрсистемы. Для ССШШ пользователю потребуется набор из n таких ключей, а также вектор-номер, состоящий из n элементов поля $\mathbb{F}_q$. Легко проверить, что эта информация займёт $n(1+R)$ длин ключей.

Единицей измерения во втором столбце выбрана длина зашифрованного секрета, так как используемая в качестве открытого ключа матрица M состоит из зашифрованных на различных ключах частей разделённого секрета. Так как части разделённого секрета имеют ту же длину, что и сам секрет, матрица M занимает в памяти nq длин зашифрованного секрета.

Единицей измерения в третьем столбце является число операций расшифрования. Для получения секрета s необходимо выполнить n расшифрований. После i-го расшифрования получим одну из частей s_i разделённого секрета, а после всех расшифрований восстановим s.

Из таблицы видно, что параметры КСШШ и тривиальной схемы, в том числе объём затрат памяти, зависят от величин N и c, в то время как параметры ССШШ зависят от параметров кода: длины кода n и мощности поля q. Таким образом, сразу сделать вывод о большей эффективности той или иной схемы не удастся. Однако из таблицы можно сформулировать следующее утверждение. Оно понадобится при дальнейшем построении ССШШ, более эффективных, чем тривиальная схема и КСШШ.

Утверждение 1. Пусть $C - [n, k, d]_q$-код, используемый для построения (N, c)-ССШШ. ССШШ является более эффективной относительно затрат по памяти, чем КСШШ и тривиальная схема, если $nq < \min\{N, 8c^4 \log N, 32/3 \cdot \mathrm{e}c^3 \log^4 c \log(\mathrm{e}N/c)\}$.

Доказательство. Прямо следует из таблицы. ■

Определение 2. Нижним порогом затраты памяти для ССШШ будем называть величину $B(N, c) = \min\{N, 8c^4 \log N, 32/3 \cdot \mathrm{e}c^3 \log^4 c \log(\mathrm{e}N/c)\}$.

Будем сравнивать тривиальную схему и КСШШ с ССШШ только относительно затрат памяти, то есть относительно размера матрицы M, и не будем сравнивать схемы относительно длины ключевой информации и сложности расшифрования. Ясно, что с точки зрения двух последних параметров наиболее эффективной является тривиальная схема, где пользователю понадобится только один ключ и одна операция расшифрования. При этом, однако, тривиальная схема показывает большую неэффективность с точки зрения затрат памяти. Большой размер открытого ключа создаёт большую нагрузку на канал распространения данных, что существенно осложняет техническую реализацию этой схемы. Поэтому сравнение схем относительно затрат памяти представляется наиболее интересным.

Построение ССШШ, более эффективных по всем рассмотренным в таблице параметрам, представляется темой отдельного исследования. Для этого нужно сформулировать утверждения об остальных параметрах, аналогичные утверждению 1, и следить за их выполнением при организации схемы.

Утверждение 2. Пусть $C - [n, k, d]_q$-код. Код C возможно использовать для организации (N, c)-ССШШ, если выполнено условие $q^k \geqslant N$.

Доказательство. Для организации (N, c)-ССШШ каждому из N пользователей необходимо выдать вектор-номер, являющийся кодовым словом кода C. Значит, количество кодовых слов не должно быть меньше числа пользователей. У кода C всего q^k кодовых слов. Получаем неравенство $q^k \geqslant N$. ■

1.2. Коды защиты от копирования

Пусть $c \in \mathbb{N} \setminus \{1\}$. Коалицией кода C назовём множество $C_0 = \{u^{(1)}, u^{(2)}, \ldots, u^{(c)}\}$, где $u^{(i)} \in C$. Число c будем называть мощностью коалиции, а множество коалиций кода мощности не больше c будем обозначать $\mathrm{coal}_c(C)$. Множеством потомков коалиции C_0 назовём множество

$$\mathrm{desc}(\mathrm{C}_0) = \{(y_1, y_2, , \ldots, y_n) \in \mathbb{F}_q^n : y_i = u_i^{(j)},\ j \in \{1, \ldots, c\}\}.$$

Определение 3 [2, определение 1.1]. Будем называть линейный код c-ТА-кодом, если выполняется следующее условие:

$$\forall\ C_0 \in \mathrm{coal}_c(C)\ \ \forall\ v \in (C \setminus C_0)\ \ \forall\ y \in \mathrm{desc}(\mathrm{C}_0)\ \ \exists\ \omega \in C_0\ \mathrm{d}(w, y) < \mathrm{d}(v, y).$$

Утверждение 3 [3, теорема 5]. Пусть C — код длины n с минимальным кодовым расстоянием d, $c \in \mathbb{N} \setminus \{1\}$. Если код C удовлетворяет условию $d > n - n/c^2$, то C является c-ТА-кодом, причём если $C_0 \in \mathrm{coal}_c(C)$ и $w \in \mathrm{desc}(C_0)$, то

$$1)\ \exists\ y \in C_0\ (\mathrm{d}(w, y) \leqslant n - n/c);\quad 2)\ \forall\ v \in C \setminus C_0\ (\mathrm{d}(w, v) > n - n/c).$$

Под радиусом работы списочного декодера будем понимать максимальное число исправляемых им ошибок.

Следствие 1. Пусть C — код длины n с минимальным кодовым расстоянием d, $c \in \mathbb{N} \setminus \{1\}$. Рассмотрим списочный декодер этого кода с радиусом работы r. Тогда достаточные условия применимости кода и декодера в ССШШ выглядят следующим образом: $d > n - n/c^2$, $r \geqslant n - n/c$.

Условие c-ТА, наличие подходящего в смысле следствия 1 списочного декодера, а также условия из утверждений 1 и 2 являются достаточными условиями того, что код может быть использован для построения (N, c)-ССШШ. В работе исследована возможность использования АГ-кодов L-конструкции [6, 9, 12] и методов списочного декодирования [7, 8] для построения (N, c)-ССШШ.

1.3. Алгеброгеометрические коды

Рассмотрим концептуальный способ построения АГ-кода L-конструкции над $\mathbb{F}_q$. Пусть $m \in \mathbb{N} \setminus \{1\}$, $\mathbb{F}_q[x_1, \ldots, x_m]$ — кольцо многочленов от m переменных с коэффициентами из поля $\mathbb{F}_q$. Неприводимость многочлена $f \in \mathbb{F}_q[x_1, \ldots, x_m]$ далее рассматривается над любыми алгебраическими расширениями поля $\mathbb{F}_q$ вплоть до его алгебраического замыкания, если это не оговаривается особо.

Определение 4. Пусть $f \in \mathbb{F}_q[x_1, x_2]$ — неприводимый многочлен. Нули этого многочлена являются точками двумерного аффинного пространства $\mathbb{A}^2(\mathbb{F}_q)$ [9, разд. 2.1.1, с. 106]. Их объединение будем называть плоской аффинной кривой [9, разд. 2.1.1; 12, определение 2.2].

Обозначим $\mathbb{F}_q^{\hom}[X_1, X_2, X_3]$ множество однородных многочленов из $\mathbb{F}_q[X_1, X_2, X_3]$.

Определение 5. Пусть $F \in \mathbb{F}_q^{\hom}[X_1, X_2, X_3]$ — неприводимый многочлен. Нули этого многочлена являются точками двумерного проективного пространства $\mathbb{P}^2(\mathbb{F}_q)$ [9, разд. 2.1.1, с. 106]. Их объединение будем называть плоской проективной кривой и обозначать как X_{pr}:

$$X_{\text{pr}} = \{P = (X_1 : X_2 : X_3) \in \mathbb{P}^2(\mathbb{F}_q) : F(X_1, X_2, X_3) = 0\}.$$

Количество точек на кривой X будем обозначать $|X|$. Далее везде будем считать, что кривая X_{pr} задана нулями многочлена F.

Между неприводимыми многочленами f из $\mathbb{F}_q[x_1, x_2]$ и F из $\mathbb{F}_q[X_1, X_2, X_3]$ существует взаимно-однозначное соответствие [12, с. 7–8]. Таким образом, существует также и взаимно-однозначное соответствие между аффинными и проективными кривыми. Процесс построения проективной кривой по соответствующей аффинной называется гомогенизацией.

Определение 6. Пусть $F \in \mathbb{F}_q^{\hom}[X_1, X_2, X_3]$, нули которого задают проективную кривую X_{pr}; (F) — главный идеал в $\mathbb{F}_q[X_1, X_2, X_3]$, порождённый F. Можно проверить, что

$$R = \left\{ \frac{P}{Q} : P, Q \in \mathbb{F}_q^{\hom}[X_1, X_2, X_3], \deg(P) = \deg(Q), Q \notin (F) \right\}$$

с естественными операциями умножения и сложения дробей является кольцом, а

$$I = \left\{ \frac{P}{Q} : P, Q \in \mathbb{F}_q^{\hom}[X_1, X_2, X_3], \deg(P) = \deg(Q), Q \notin (F), P \in (F) \right\}$$

— максимальным идеалом в R [12, с. 7]. Фактор-кольцо R/I в силу максимальности идеала I является полем [13, с. 80], оно называется полем рациональных функций на кривой X_{pr} и обозначается как $\mathbb{F}_q(X_{\text{pr}})$.

Очевидно, что $F \in \mathbb{F}_q[X_1, X_2, X_3]$ можно представить как

$$F = \sum_{i+j+k=\deg(F)} \alpha_{i,j,k} X_1^i X_2^j X_3^k, \quad \alpha_{i,j,k} \in \mathbb{F}_q.$$

Частной производной по переменной X_1 называется многочлен

$$F_{X_1} = \sum_{i+j+k=\deg(F)} i\alpha_{i,j,k} X_1^{i-1} X_2^j X_3^k, \quad \alpha_{i,j,k} \in \mathbb{F}_q.$$

Здесь элемент $i\alpha_{i,j,k}$ нужно понимать как $\alpha_{i,j,k}$, сложенный с самим собой i раз. Аналогично определяются частные производные F_{X_2} и F_{X_3}.

Определение 7. Плоская проективная кривая X_{pr} называется гладкой в точке $M \in X_{\mathrm{pr}}$, если хотя бы одно из чисел $F_{X_1}(M), F_{X_2}(M), F_{X_3}(M)$ не равно нулю.

Далее будем рассматривать только гладкие плоские кривые.

Кривые имеют параметр $g \in \mathbb{N} \cup \{0\}$, называемый родом. В случае плоских гладких кривых род вычисляется по известной формуле [9, следствие 2.2.8]. Пусть F — многочлен, нули которого задают кривую. Тогда

$$g = (\deg(F) - 1)(\deg(F) - 2)/2. \tag{1}$$

Из формулы (1) видно, что гладкие плоские кривые могут иметь в качестве рода число g, только если существуют решения уравнения $(x-1)(x-2) = 2g$.

Утверждение 4 [9, теорема 3.1.25]. Пусть X — аффинная или проективная кривая рода g над полем $\mathbb{F}_q$, тогда $|X| \leqslant q + 1 + g\lceil 2\sqrt{q} \rceil$.

Утверждение 5 [12, определение 2.15 и теорема 2.16]. Пусть X_{pr} — проективная кривая, $\mathbb{F}_q(X_{\mathrm{pr}})$ — поле рациональных функций на ней, $M \in X_{\mathrm{pr}}$. Тогда

$$\exists T \in \mathbb{F}_q(X_{\mathrm{pr}})\ \forall H \in \mathbb{F}_q(X_{\mathrm{pr}})\ \exists U \in \mathbb{F}_q(X_{\mathrm{pr}})(T(M) = 0\ \&\ U(M) \neq 0\ \&\ H = T^m U),$$

где $m \in \mathbb{Z}$, причём значение m не зависит от выбора элемента T.

Определение 8. Порядком $H = T^m U \in \mathbb{F}_q(X_{\mathrm{pr}})$ в точке $M \in X_{\mathrm{pr}}$ назовём значение m и будем обозначать это $\mathrm{ord}_M(H) = m$. Договоримся, что $\mathrm{ord}_M(0) = \infty$.

Замечание 1. Заметим, что в силу определения 8 порядок (в фиксированной точке) произведения функций равен сумме порядков функций-сомножителей, а порядок функции, обратной данной, равен по модулю порядку данной функции, но имеет противоположный знак.

Определение 9. Пусть M — точка на кривой X_{pr}; $L \in \mathbb{F}_q^{\mathrm{hom}}[X, Y, Z]$ — однородный многочлен степени 1, такой, что $L(M) \neq 0$; $G \in \mathbb{F}_q^{\mathrm{hom}}[X, Y, Z]$, $\deg(G) = r$. Назовём кратностью пересечения G и X_{pr} в точке M число $\mathrm{ord}_M(G/L^r)$ и обозначим её как $I(M; X_{\mathrm{pr}}; G)$.

Понятие кратности пересечения является аналогом понятия порядка для однородных многочленов.

Утверждение 6 [12, теорема 2.23]. Пусть $G \in \mathbb{F}_q^{\mathrm{hom}}[X, Y, Z]$ и кривая X_{pr} задана многочленом F. Предположим, что F не делит G. Тогда выполняется равенство

$$\sum_{M \in X_{\mathrm{pr}}} I(M; X_{\mathrm{pr}}; G) = \deg(G)\deg(F).$$

Определение 10. Дивизором D на проективной кривой X_{pr} называется формальная сумма следующего вида: $D = \sum\limits_{M \in X_{\text{pr}}} a_M M$, $a_M \in \mathbb{Z}$.

Так как множество точек на проективной кривой над конечным полем конечно, то эти суммы всегда конечны.

Определение 11. Степенью дивизора D называют число $\deg(D) = \sum a_M$, а носителем дивизора — множество $\operatorname{supp}(D) = \{M \in X_{\text{pr}} : a_M \neq 0\}$.

Определение 12. Говорят, что дивизор $D = \sum a_M M$, $a_M \in \mathbb{Z}$, $M \in X_{\text{pr}}$, эффективен, если все $a_M \geqslant 0$. Этот факт обозначается следующим образом: $D \geqslant 0$.

Дивизоры образуют абелеву группу относительно поточечного сложения.

Определение 13. Пусть $G \in \mathbb{F}_q^{\text{hom}}[X, Y, Z]$, $\deg(G) = r$. Назовём дивизором пересечения G и X_{pr} дивизор вида

$$X_{\text{pr}} \cdot G = \sum_{M \in X_{\text{pr}}} I(M; X_{\text{pr}}; G) M.$$

Определение 14. Пусть $H = P/Q \in \mathbb{F}_q(X_{\text{pr}})$, где $P, Q \in \mathbb{F}_q^{\text{hom}}[X, Y, Z]$. Дивизором H на проективной кривой X_{pr} называется дивизор

$$(H) = \sum_{M \in X_{\text{pr}}} \operatorname{ord}_M(H) M.$$

Замечание 2. Учитывая определения 8 и 9, можно проверить, что

$$(H) = X_{\text{pr}} \cdot P - X_{\text{pr}} \cdot Q.$$

Действительно, используя замечание 1, получаем

$$\operatorname{ord}_M(H) = \operatorname{ord}_M\left(\frac{P}{Q}\right) = \operatorname{ord}_M\left(\frac{P}{L^r} \cdot \frac{L^r}{Q}\right) = \operatorname{ord}_M\left(\frac{P}{L^r}\right) + \operatorname{ord}_M\left(\frac{L^r}{Q}\right) =$$
$$= \operatorname{ord}_M\left(\frac{P}{L^r}\right) - \operatorname{ord}_M\left(\frac{Q}{L^r}\right) = I(M; X_{\text{pr}}; P) - I(M; X_{\text{pr}}; Q),$$

где $L \in \mathbb{F}_q^{\text{hom}}[X, Y, Z]$; $\deg(L) = 1$; $L(M) \neq 0$; $r = \deg(P) = \deg(Q)$.

Определение 15. Зафиксируем дивизор $D = \sum a_M M$ и рассмотрим множество

$$L(D) = \{H \in \mathbb{F}_q(X_{\text{pr}}) : (H) + D \geqslant 0\}.$$

Множество $L(D)$ называется пространством функций Римана — Роха, ассоциированным с дивизором D. Пространство Римана — Роха является конечномерным векторным пространством [12, определение 2.36, теорема 2.37].

Пусть $X_{\text{pr}} \neq \varnothing$ — гладкая плоская проективная кривая, $Points = \{P_1, \ldots, P_n\} \subset X_{\text{pr}}$, $D = \sum\limits_{M \in X_{\text{pr}}} a_M M$, $a_M \in \mathbb{Z}$, — дивизор на X_{pr}, такой, что $\operatorname{supp}(D) \cap Points = \varnothing$. Построим отображение

$$Ev_{Points} : L(D) \to \mathbb{F}_q^n, \quad Ev_{Points}(h) = (H(P_1), H(P_2), \ldots, H(P_n)). \tag{2}$$

Образ этого отображения $C = \operatorname{Im}(Ev_{Points}(L(D)))$ называется АГ-кодом L-конструкции. Как видно, АГ-код зависит от выбора кривой X_{pr}, множества $Points$ и дивизора D. Дивизор D будем называть дивизором кода C.

Заметим, что при вычислении функции $H \in L(D)$ на точках из $Points$ невозможно деление на ноль. Действительно, если существует $P_i \in Points$, такая, что при вычислении H в P_i проявляется деление на ноль, это означает, что $\mathrm{ord}_{P_i}(H) < 0$. Но $P_i \notin \mathrm{supp}(D)$, так как $\mathrm{supp}(D) \cap Points = \varnothing$. Значит, в дивизоре $(H) + D$ есть слагаемое, соответствующее P_i, при котором коэффициент отрицателен. Но это невозможно, так как $(H) + D \geqslant 0$. Таким образом, предположение неверно, и $\mathrm{ord}_{P_i}(H) \geqslant 0$ для любой точки $P_i \in Points$, то есть деление на ноль невозможно.

Утверждение 7 [9, теорема 4.1.1]. Пусть X_{pr} — кривая рода g и D — дивизор, такой, что $0 < \deg(D) = \alpha < n$. Тогда АГ-код $C = \mathrm{Im}(Ev_{Points}(L(D)))$ является $[n, k, d]_q$-кодом, где $k \geqslant \alpha - g + 1$ и $d \geqslant n - \alpha$. Если $\alpha > 2g - 2$, то $k = \alpha - g + 1$.

Замечание 3. Величина $d^* = n - \alpha$ называется конструктивным расстоянием алгеброгеометрического кода C. Если $\alpha > 2g - 2$, то $d^* = n - k - g + 1$. Отсюда видно, что C является кодом с максимально допустимым расстоянием тогда и только тогда, когда $g = 0$.

Замечание 4. Естественно встаёт вопрос о вычислении базиса пространства Римана — Роха $L(D)$, являющегося, как видно по построению АГ-кода, пространством кодируемых сообщений.

При малых параметрах m, q, g базис можно попытаться вычислить вручную. Сначала необходимо найти все точки кривой. Далее, так как пространство $L(D)$ является пространством кодируемых сообщений, размерность этого пространства равна размерности k соответствующего АГ-кода. Предположим, что $\deg(D) = \alpha > 2g - 2$, тогда $k = \alpha - g + 1$. Таким образом, нужно найти k линейно независимых функций ϕ_i, таких, что $(\phi_i) + D \geqslant 0$. Для проверки этих условий нужно знать величины $\mathrm{ord}_M(\phi_i)$, где $M \in X_{\mathrm{pr}}$. Для этого можно использовать замечание 2, в частности то, что порядок функции в точке равен разности кратностей пересечений числителя и знаменателя. Кратность пересечения $I(M; X_{\mathrm{pr}}; G)$ многочлена G в точке M с кривой X_{pr} равна нулю, если $G(M) \neq 0$ (определение 9). Для определения кратности пересечения в случае $G(M) = 0$ необходимо использовать различные свойства. Например, если известны кратности пересечений для всех точек, помимо M, кратность пересечения можно найти из теоремы 6. В силу замечания 1, если известна некоторая факторизация $G = G_1 \cdot G_2 \cdot \ldots \cdot G_s$, то $I(M; X_{\mathrm{pr}}; G) = I(M; X_{\mathrm{pr}}; G_1) + I(M; X_{\mathrm{pr}}; G_2) + \ldots + I(M; X_{\mathrm{pr}}; G_s)$. Кроме того, если $\deg(G) = 1$ и $G(M) = 0$, то если G не является касательной в точке M к X_{pr}, можно утверждать, что $I(M; X_{\mathrm{pr}}; G) = 1$. Это часто используется в [12], примеры 2.33 и 2.38, 2.34 и 2.76. Уравнение касательной к кривой X_{pr}, заданной многочленом F, в точке M в проективном пространстве выглядит следующим образом [12, определение 2.11]:

$$F_{X_1}(M)X_1 + F_{X_2}(M)X_2 + F_{X_3}(M)X_3 = 0.$$

Для обработки больших параметров могут понадобиться такие алгоритмы вычисления базиса, как, например, алгоритм Хесса [14]. Алгоритм Хесса используется, например, в пакете компьютерной алгебры Magma [15].

Далее будем считать, что $0 < \deg(D) = \alpha < n$.

Замечание 5. Если дивизор D имеет вид $D = \alpha Q$, то построенный по такому дивизору код называется одноточечным АГ-кодом. В таком случае в качестве точки Q обычно берут точку вида $(M_1 : M_2 : 0), M_i \in \mathbb{F}_q$ [9, пример 4.1.5; 12, примеры 2.67 и 2.75].

1.4. Классический декодер Судана — Гурусвами

Утверждение 8 [7, теорема 27]. Пусть C — АГ-код длины n, D — дивизор кода C, $\deg(D) = \alpha$, конструктивное расстояние кода $d^* = n - k - g + 1$. Тогда существует алгоритм декодирования (классический списочный декодер Судана — Гурусвами — классический СДСГ) этого кода со сложностью, полиномиальной по n, исправляющий $r = \lfloor n - \sqrt{n(k+g-1)} \rfloor$ ошибок.

Замечание 6. Если $\alpha > 2g - 2$, то из утверждения 7 видно, что $\alpha = k + g - 1$ и $d^* = n - k - g + 1$. Таким образом, в силу утверждения 8 при выполнении условия $\alpha > 2g - 2$ существует возможность использования СДСГ.

1.5. Мягкий декодер Судана — Гурусвами

Утверждение 9 [8, с. 2 (5)]. Пусть C — АГ-код длины n над полем $\mathbb{F}_q$, d^* — конструктивное расстояние кода и для некоторых $m, l \in \mathbb{N} \cup \{0\}$ выполнено неравенство

$$\frac{l^2}{m} + \frac{(m-l)^2}{m(q-1)} + \frac{n-m}{q} \geqslant n - d^* + \varepsilon,$$

где $\varepsilon \in \mathbb{R}$ — заданный заранее безразмерный параметр допустимого отклонения, $\varepsilon > 0$. Тогда существует алгоритм декодирования (мягкий списочный декодер Судана — Гурусвами — мягкий СДСГ) этого кода со сложностью, полиномиальной по n, исправляющий $n - m$ стираний и $m - l$ ошибок.

2. Результаты

2.1. Достаточные условия применимости АГ-кодов L-конструкции и декодера Судана — Гурусвами в ССШШ

Представим границы, в пределах которых АГ-коды и мягкий и классический СДСГ могут быть применены для построения (N, c)-ССШШ.

Пусть C — АГ-код L-конструкции над $\mathbb{F}_q$, n — его длина, k — размерность, g — род кривой, на которой определён код C, D — дивизор кода C, $\deg(D) = \alpha$. Рассмотрим неравенство

$$\alpha > 2g - 2. \tag{3}$$

По утверждению 7 при выполнении условия (3) $\alpha = k + g - 1$.

Теорема 1. Код C является c-ТА-кодом, если выполняется условие

$$c < \sqrt{n/\alpha}. \tag{4}$$

Доказательство. Согласно следствию 1, для того чтобы код C являлся c-ТА-кодом, достаточно выполнения следующего условия:

$$d > n - n/c^2. \tag{5}$$

Согласно утверждению 7 и замечанию 3, $d \geqslant d^* = n - \alpha$. Решив неравенство для конструктивного расстояния $d^* > n - n/c^2$ относительно c, получим значения c, которые удовлетворяют (5). Решая, получаем

$$c < \sqrt{n/(n-d^*)} = \sqrt{n/\alpha}.$$

Таким образом, для выполнения c-ТА достаточно выполнения (4). ■

Частный случай теоремы 1 приведён в [3, теорема 6, часть 2] в случае, когда выполнено условие (3). Доказательство этого частного случая можно получить, используя доказательство теоремы 1, имея в виду, что при выполнении (3) $\alpha = k + g - 1$.

Лемма 1. Максимально возможный радиус классического СДСГ равен r, где

$$r = \begin{cases} n - \left\lceil \sqrt{n(k+g-1)} \right\rceil, & \sqrt{n(k+g-1)} \notin \mathbb{N}, \\ n - \sqrt{n(k+g-1)} - 1, & \sqrt{n(k+g-1)} \in \mathbb{N}. \end{cases}$$

Доказательство. В силу утверждения 8 максимально возможный радиус при условии $\sqrt{n(k+g-1)} \notin \mathbb{N}$ равен $r = \lfloor n - \sqrt{n(k+g-1)} \rfloor$, так как в этом случае это максимальное из натуральных чисел r, удовлетворяющих неравенству $r < n - \sqrt{n(k+g-1)}$. С учётом того, что $\lfloor -x \rfloor = -\lceil x \rceil$ и $\forall k \in \mathbb{Z}\ \forall x \in \mathbb{R}\ (\lfloor k + x \rfloor = k + \lfloor x \rfloor)$, получаем

$$\begin{aligned} r &= \left\lfloor n - \sqrt{n(k+g-1)} \right\rfloor = \left\lfloor n + -(\sqrt{n(k+g-1)}) \right\rfloor = \\ &= n + \left\lfloor -(\sqrt{n(k+g-1)}) \right\rfloor = n - \left\lceil \sqrt{n(k+g-1)} \right\rceil. \end{aligned}$$

В силу утверждения 8 максимально возможный радиус при условии $\sqrt{n(k+g-1)} \in \mathbb{N}$ равен $r = n - \sqrt{n(k+g-1)} - 1$, так как в этом случае это максимальное из натуральных чисел r, удовлетворяющее неравенству $r < n - \sqrt{n(k+g-1)}$. ■

Замечание 7. Работу декодера всегда можно организовать с максимальным радиусом декодирования, выбирая соответствующим образом управляемые параметры алгоритма классического СДСГ [7, п. 6.3.3].

Теорема 2.

1) Если $\sqrt{n\alpha} \notin \mathbb{N}$, то классический СДСГ для кода C применим для построения (N, c)-ССШШ, если выполняются условие (3) и условие

$$c \leqslant \frac{n}{\lceil \sqrt{n\alpha}\ \rceil}. \tag{6}$$

2) Если $\sqrt{n\alpha} \in \mathbb{N}$, то классический СДСГ для кода C применим для построения (N, c)-ССШШ, если выполняются условие (3) и условие

$$c \leqslant \frac{n}{\sqrt{n\alpha} + 1}. \tag{7}$$

При выполнении условия (6) выполняется и условие (4), а при выполнении (7) выполняется (6).

Доказательство. Условие (3) обосновывается замечанием 6. По утверждению 7 при выполнении этого условия $\alpha = k + g - 1$. Из леммы 1 возьмём выражение для радиуса работы классического СДСГ: если $\sqrt{n\alpha} \notin \mathbb{N}$, то $r = n - \lceil \sqrt{n\alpha} \rceil$, иначе $r = n - \sqrt{n\alpha} - 1$. По следствию 1, радиус декодера должен удовлетворять условию

$$r = n - \lceil \sqrt{n\alpha} \rceil \geqslant n - n/c,$$

что эквивалентно

$$c \leqslant \frac{n}{n - r}. \tag{8}$$

Подставив в (8) значение r при $\sqrt{n\alpha} \notin \mathbb{N}$, получим (6), а подставив r при $\sqrt{n\alpha} \in \mathbb{N}$, получим (7).

Докажем, что из (6) следует (4). Условие (4) можно переписать в виде $c \leqslant n/\sqrt{n\alpha}$. В случае $\sqrt{n\alpha} \notin \mathbb{N}$ выполняется строгое неравенство $\lceil \sqrt{n\alpha} \rceil > \sqrt{n\alpha}$, и тогда $c \leqslant \leqslant n/\lceil \sqrt{n\alpha} \rceil < n/\sqrt{n\alpha}$. Значит, (6) накладывает более сильное условие на c, чем (4), следовательно, выполнение (6) влечёт за собой и выполнение (4).

Докажем, что из (7) следует (6). При $\sqrt{n\alpha} \in \mathbb{N}$ выполняется неравенство $\sqrt{n\alpha}+1 > > \lceil \sqrt{n\alpha} \rceil$, тогда $c \leqslant n/\sqrt{n\alpha}+1 < n/\lceil \sqrt{n\alpha} \rceil$. Значит, выполнение условия (7) влечёт за собой и выполнение (6). ■

Теорема 3. Предположим, что слово-потомок коалиции злоумышленников получено без стираний. Пусть

$$D = \sqrt{n(q-1)(n(q-1) - q(n-\alpha-\varepsilon))},$$

где ε — параметр допустимого отклонения для мягкого СДСГ ($\varepsilon \in \mathbb{R}$, $\varepsilon > 0$). Тогда мягкий СДСГ для кода C применим для построения (N, c)-ССШШ, то есть гарантируется определение как минимум одного члена коалиции злоумышленников, если выполняются условие (3) и условие

$$c \geqslant \frac{n - \lfloor (n(q-1) - D)/q \rfloor - 1}{\alpha}. \tag{9}$$

Доказательство. Пусть r_{soft} — радиус декодирования мягкого СДСГ. Из теоремы 9, учитывая, что $m = n$ (так как считаем, что стираний нет), получаем неравенство

$$\frac{(n - r_{\text{soft}})^2}{n} + \frac{r_{\text{soft}}^2}{n(q-1)} \geqslant n - d^* + \varepsilon.$$

Решая это неравенство относительно r_{soft}, получаем условия на радиус декодирования для мягкого декодера. Пусть $D = \sqrt{n(q-1)(n(q-1) - q(d^* - \varepsilon))}$. Тогда

$$r_{\text{soft}} \leqslant R_1 = \frac{n(q-1) - D}{q}, \quad r_{\text{soft}} \geqslant R_2 = \frac{n(q-1) + D}{q}.$$

При достаточно больших n, в частности при $n > (d^* - \varepsilon)(q-1)/(q-2)$, величина R_2 превосходит n, следовательно, второе ограничение на r_{soft} перестаёт иметь смысл, так как $r_{\text{soft}} < n$. Таким образом, существенным остаётся только первое условие. Тогда максимально возможное значение радиуса работы мягкого декодера равно

$$r_{\text{soft}} = \lfloor R_1 \rfloor = \left\lfloor \frac{n(q-1) - D}{q} \right\rfloor.$$

На шаге 4 на первой итерации алгоритма определения злоумышленников в [8, п. «Example»], то есть на итерации, где потенциально может быть определён первый пользователь из коалиции злоумышленников, накладывается ещё одно ограничение на радиус мягкого декодера — $r_{\text{soft}} \geqslant C = n - c(k+g-1) - 1$, так как на этом шаге из полученного списочным декодером списка пользователей выбираются те слова, что имеют одинаковые значения с текущим потомком хотя бы в $c(k+g-1)+1$ позициях.

По условию теоремы выполняется условие (3), то есть $\alpha = k + g - 1$. Получаем

$$\lfloor (n(q-1) - D)/q \rfloor \geqslant n - c(k+g-1) - 1,$$
$$c \geqslant \frac{n - \lfloor (n(q-1) - D)/q \rfloor - 1}{k+g-1},$$
$$c \geqslant \frac{n - \lfloor (n(q-1) - D)/q \rfloor - 1}{\alpha}.$$

Учитывая, что для АГ-кодов $d^* = n - \alpha$ (замечание 3), величину D можно переписать в виде $D = \sqrt{n(q-1)(n(q-1) - q(n-\alpha-\varepsilon))}$. ■

Теорема 4. Код C возможно использовать для организации (N, c)-ССШШ, если выполнено условие

$$\alpha \geqslant \log_q N + g - 1. \tag{10}$$

Доказательство. Из утверждения 7 и замечания 3 видно, что если $\alpha \geqslant 2g-1$, то $k = \alpha - g + 1$. Тогда условие $N \leqslant q^k$ возможности применения кода C в (N, c)-ССШШ (утверждение 2) переписывается в виде $N \leqslant q^{\alpha-g+1}$, откуда получаем условие (10). ■

Лемма 2. Пусть C — c-ТА-АГ-код над полем $\mathbb{F}_q$, где $q = 2^m$ для некоторого натурального m; n — длина кода, D — дивизор кода C, $\deg(D) = \alpha$, конструктивное расстояние кода d^* — нечётное число. Тогда код $\hat{C}$, являющийся удлинением кода C с помощью добавления проверки чётности [18, п. 1.9(I)], также является c-ТА-кодом.

Доказательство. По замечанию 3, для кода C конструктивное расстояние $d^* = n - \alpha$. Согласно [18, п. 1.9(I)], при добавлении позиции проверки чётности к двоичному $[n, k, d]_q$-коду с нечётным расстоянием d получим $[n+1, k, d+1]_q$-код. Значит, код $\hat{C}$ как удлинение $[n, k, \geqslant n-\alpha]_q$-кода C является $[n+1, k, \geqslant n-\alpha+1]_q$-кодом. Действуя аналогично доказательству теоремы 1 и решив неравенство для конструктивного расстояния

$$d^* = n - \alpha + 1 > n + 1 - (n+1)/c^2$$

относительно c, получим значения c, которые удовлетворяют достаточному условию c-ТА (5) для кода $\hat{C}$:

$$c < \sqrt{n + 1/\alpha}. \tag{11}$$

Так как C — c-ТА-код, выполнено условие (4). Очевидно, что из (4) вытекает и справедливость условия (11). Значит, $\hat{C}$ также является c-ТА-кодом. ■

Лемма 3. Пусть стоит задача построения (N, c)-ССШШ. Рассмотрим кривые рода g над $\mathbb{F}_q$. Пусть D — дивизор на некоторой кривой, $\deg(D) = \alpha$. Тогда АГ-код C с дивизором D возможно использовать в ССШШ, если $\alpha \geqslant \max\{\lceil \log_q N \rceil + g - 1, 2g-1\}$.

Доказательство. Для использования в (N, c)-ССШШ кода C должны выполняться условия $\alpha \geqslant \log_q N + g - 1$ (теорема 4) и $\alpha \geqslant 2g - 1$ (теоремы 2 и 3). Значит, $\alpha \geqslant \max\{\lceil \log_q N + g - 1 \rceil, 2g-1\}$; заметим, что $\lceil \log_q N + g - 1 \rceil = \lceil \log_q N \rceil + g - 1$. ■

Напомним, что $B(N, c)$ – нижний порог затрат памяти для ССШШ (определение 2).

Теорема 5. Рассмотрим следующие четыре неравенства:

Неравенство 1 имеет вид

— если $c^2 > \lfloor 2\sqrt{q} \rfloor$, то

$$0 \leqslant g \leqslant \min\left\{\lceil \log_q N \rceil, \frac{\lfloor B(N,c)/q + 1 \rfloor - q - 1}{\lfloor 2\sqrt{q} \rfloor}, \frac{q - c^2(\lceil \log_q N \rceil - 1)}{c^2 - \lfloor 2\sqrt{q} \rfloor}\right\};$$

— если $c^2 < \lfloor 2\sqrt{q} \rfloor$ и $q < c^2(\lceil \log_q N \rceil - 1)$, то

$$\max\left\{0, \frac{c^2(\lceil \log_q N \rceil - 1) - q}{\lfloor 2\sqrt{q} \rfloor - c^2}\right\} \leqslant g \leqslant \min\left\{\lceil \log_q N \rceil, \frac{\lfloor B(N,c)/q + 1 \rfloor - q - 1}{\lfloor 2\sqrt{q} \rfloor}\right\};$$

— иначе

$$0 \leqslant g \leqslant \min\left\{\lceil \log_q N \rceil, \frac{\lfloor B(N,c)/q + 1 \rfloor - q - 1}{\lfloor 2\sqrt{q} \rfloor}\right\}.$$

Неравенство 2 имеет вид

$$\lceil \log_q N \rceil \leqslant g \leqslant \min\left\{ \frac{\lfloor B(N,c)/q+1 \rfloor - q - 1}{\lfloor 2\sqrt{q} \rfloor}, \frac{q+c^2}{2c^2 - \lfloor 2\sqrt{q} \rfloor} \right\}.$$

Неравенство 3 имеет вид

$$\max\left\{0, \frac{\lfloor B(N,c)/q+1 \rfloor - q - 1}{\lfloor 2\sqrt{q} \rfloor}\right\} \leqslant g \leqslant \min\left\{ \lceil \log_q N \rceil, \frac{\lfloor B(N,c)/q+1 \rfloor - q - 1}{\lfloor 2\sqrt{q} \rfloor}\right\}.$$

Неравенство 4 имеет вид

$$\max\left\{ \lceil \log_q N \rceil, \frac{\lfloor B(N,c)/q+1 \rfloor - q - 1}{\lfloor 2\sqrt{q} \rfloor}\right\} \leqslant g \leqslant \frac{\lfloor B(N,c)/q+1 \rfloor - c^2 - 1}{2c^2}.$$

Если род кривой g удовлетворяет этим неравенствам, то на кривой рода g возможно построить c-ТА-код, причём

1) длина этого кода минимальна, при которой ещё выполняется достаточное условие c-ТА (4);
2) этот код возможно использовать для построения (N,c)-ССШШ, более эффективных относительно затрачиваемой памяти, чем КСШШ.

Доказательство. Для того чтобы на кривой рода g над полем $\mathbb{F}_q$ можно было бы построить c-ТА-код длины n с дивизором степени α, достаточно выполнения условия (4). Минимальная величина n, при которой возможно его выполнение, равна $c^2\alpha + 1$, причём, в соответствии с леммой 3, минимальное значение α равно $\max\{\lceil \log_q N \rceil + g - 1, 2g-1\}$. Таким образом, минимально возможное n, при которой выполнено (4), равно $c^2 \max\{\lceil \log_q N \rceil + g - 1, 2g-1\} + 1$.

Кроме того, на кривой должно быть достаточное количество точек для построения кода. В частности, учитывая возможность дальнейшего использования проверки чётности и утверждение 4, их должно быть не менее $q + g\lfloor 2\sqrt{q} \rfloor + 1$. Так получаем условие

$$n = c^2 \max\{\lceil \log_q N \rceil + g - 1, 2g-1\} + 1 \leqslant q + g\lfloor 2\sqrt{q} \rfloor + 1.$$

Для возможности использования такого кода в ССШШ, более эффективной, чем КССШ и тривиальная схема, по утверждению 1 должно выполняться условие

$$n = c^2 \max\{\lceil \log_q N \rceil + g - 1, 2g-1\} + 1 \leqslant q + g\lfloor 2\sqrt{q} \rfloor + 1 < B(N,c)/q.$$

Два последних неравенства можно объединить в одно:

$$c^2 \max\{\lceil \log_q N \rceil + g - 1, 2g-1\} + 1 \leqslant \min\{q + g\lfloor 2\sqrt{q} \rfloor + 1, \lfloor B(N,c)/q+1 \rfloor\}. \qquad (12)$$

Решим это неравенство относительно g. Составим четыре системы:

$$\begin{cases} \lceil \log_q N \rceil \geqslant g, \\ q + g\lfloor 2\sqrt{q} \rfloor + 1 \leqslant \lfloor B(N,c)/q+1 \rfloor, \\ c^2(\lceil \log_q N \rceil + g - 1) \leqslant q + g\lfloor 2\sqrt{q} \rfloor, \end{cases} \quad \begin{cases} \lceil \log_q N \rceil \leqslant g, \\ q + g\lfloor 2\sqrt{q} \rfloor + 1 \leqslant \lfloor B(N,c)/q+1 \rfloor, \\ c^2(2g-1) \leqslant q + g\lfloor 2\sqrt{q} \rfloor, \end{cases}$$

$$\begin{cases} \lceil \log_q N \rceil \geqslant g, \\ q + g\lfloor 2\sqrt{q} \rfloor + 1 \geqslant \lfloor B(N,c)/q+1 \rfloor, \\ c^2(\lceil \log_q N \rceil + g - 1) + 1 \leqslant \lfloor B(N,c)/q+1 \rfloor, \end{cases} \quad \begin{cases} \lceil \log_q N \rceil \leqslant g, \\ q + g\lfloor 2\sqrt{q} \rfloor + 1 \geqslant \lfloor B(N,c)/q+1 \rfloor, \\ c^2(2g-1) + 1 \leqslant \lfloor B(N,c)/q+1 \rfloor. \end{cases}$$

Первая система соответствует случаю, когда в (12) слева максимумом является $\lceil \log_q N \rceil + g - 1$, а справа минимумом $q + g\lfloor 2\sqrt{q} \rfloor + 1$. Три остальные системы соответствуют трём оставшимся случаям. Решения первой системы совпадают с решениями неравенства 1, второй — с решениями неравенства 2 и т. д. ■

2.2. Алгоритм построения АГ-кода L-конструкции, применимого в ССШШ

Предложим концептуальный алгоритм построения АГ-кода L-конструкции, который может быть применён для построения (N, c)-ССШШ. Он основан на утверждениях п. 2.1 и на способе построения АГ-кода п. 1.3. Полученный по алгоритму АГ-код является одноточечным АГ-кодом (см. замечание 5).

Алгоритм 1. Построение одноточечного АГ-кода, применимого в ССШШ

Вход: $c \in \mathbb{N} \setminus \{1\}$ — максимальная мощность коалиции злоумышленников;
$N \in \mathbb{N} \setminus \{1\}$ — число легальных пользователей в ССШШ;
q — мощность конечного поля, над которым строится код;
L — список кривых над полем $\mathbb{F}_q$;
InputDecoders $= \{\text{ClassicDecoder}, \text{SoftDecoder}, \dots\}$ — список известных списочных декодеров для АГ-кодов.

Выход: порождающая матрица q-ичного АГ-кода L-конструкции, который вместе с алгоритмом СДСГ может быть применён для построения (N, c)-ССШШ; список наиболее приоритетных для использования списочных декодеров OutputDecoders.

1: Составить множество $G = \{g_1, g_2, \dots, g_{|G|}\}$ допустимых значений рода кривой, решив четыре неравенства относительно g из теоремы 5; $i := 1$, $g := g_i$.
2: **Пока** $i \leqslant |G|$:
3: **Если** уравнение $(x-1)(x-2) = 2g$ не имеет натуральных корней, **то**
4: $i := i + 1$, $g := g_i$;
5: перейти на следующую итерацию цикла на шаг 3.
6: $\alpha := \max\{\lceil \log_q N \rceil + g - 1, 2g - 1\}$;
7: $\hat{n} := c^2\alpha + 1$.
8: **Пока** $\hat{n} \leqslant \min\{q + g\lfloor 2\sqrt{q} \rfloor, \lfloor B(N, c)/q + 1 \rfloor\}$:
9: **Если** в L есть гладкая плоская проективная кривая X, $|X| \geqslant \hat{n}$, рода g над полем $\mathbb{F}_q$, **то**
10: выбрать кривую X,
11: **иначе**
12: выйти из цикла, перейти на шаг 38.
13: **Если** $|X| \geqslant \hat{n} + 1$, **то**
14: ParityCheck := `false`;
15: $n := \hat{n}$;
16: **иначе**
17: **если** $|X| < \hat{n} + 1 \wedge \exists m \in \mathbb{N}\ (q = 2^m) \wedge \exists k \in \mathbb{N}\ (\hat{n} - \alpha = 2k + 1)$, **то**
ParityCheck := `true`;
18: $n := \hat{n} + 1$,
19: **иначе**
20: ParityCheck := `false`;
21: $\hat{n} := \hat{n} + 1$, перейти на новую итерацию цикла (шаг 8).
22: OutputDecoders := $\varnothing$.
23: **Если** SoftDecoder $\in$ InputDecoders, **то**
24: **Если** для n выполняется условие (9), **то**
25: к OutputDecoders добавить метку мягкого СДСГ SoftDecoder.
26: **Если** ClassicDecoder $\in$ InputDecoders, **то**
27: **Если** $\sqrt{n\alpha} \notin \mathbb{N}$, **то**
28: **Если** для n выполняется условие (6), **то**

29: к OutputDecoders добавить метку классического СДСГ ClassicDecoder,
30: **иначе** // $\sqrt{n\alpha} \in \mathbb{N}$
31: **Если** для n выполняется условие (7), **то**
32: к OutputDecoders добавить метку классического СДСГ ClassicDecoder,
33: // ... проверка условий на иные алгоритмы списочного декодирования.
34: **Если** OutputDecoders $= \varnothing$, **то**
35: $\hat{n} := \hat{n} + 1$, перейти на новую итерацию цикла на шаге 8,
36: **иначе**
37: выйти из цикла на шаг 41.
38: **Если** $\hat{n}q > B(N, c)$, **то**
39: **Вернуть** «Ошибка. Эффективнее использовать КСШШ».
40: $g := g + 1$.
41: Объявить точку $Q \in X$ вида $(X_1 : X_2 : 0)$ (см. замечание 5).
42: **Если** ParityCheck = `true`, **то**
43: объявить множество $Points := \{P_1, \ldots, P_{n-1}\} \subset X \setminus \{Q\}$,
44: **иначе**
45: объявить множество $Points := \{P_1, \ldots, P_n\} \subset X \setminus \{Q\}$.
46: Объявить дивизор $D := \alpha Q$ (определение 10).
47: Присвоить $k := \alpha - g + 1$.
48: Рассмотреть пространство Римана — Роха $L(D)$ (определение 15). Построить его базис из k линейно независимых элементов. Обозначить его $B = \{\phi_1, \ldots, \phi_l\}$.
49: Построить матрицу

$$G = \begin{bmatrix} \phi_1(P_1) & \phi_1(P_2) & \ldots & \phi_1(P_{|Points|}) \\ \phi_2(P_1) & \phi_2(P_2) & \ldots & \phi_2(P_{|Points|}) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_l(P_1) & \phi_l(P_2) & \ldots & \phi_l(P_{|Points|}) \end{bmatrix}.$$

50: **Если** ParityCheck = `true`, **то**
51: добавить к G столбец проверки на чётность, то есть в i-ю строку добавить элемент $\sum\limits_{j=1}^{|Points|} \phi_i(P_j)$.
52: **Вернуть** G и DecoderList.

Замечание 8. В силу требований, накладываемых утверждением 1, в алгоритме минимизируется значение длины кода n. Минимизация n при заданном q делает проектируемую ССШШ более эффективной.

Теорема 6. Алгоритм 1 является корректным, то есть с его помощью можно либо построить q-ичный АГ-код L-конструкции, который вместе с предложенными списочными декодерами OutputDecoders может быть применён для построения (N, c)-ССШШ, либо убедиться в неэффективности ССШШ по отношению к КСШШ.

Доказательство. На шаге 1 включаем в множество G возможные значения рода g, для которых выполняется свойство c-ТА (теорема 5).

На шаге 2 запускаем цикл по G, в результате которого подбираются кривая X подходящих мощности и рода, а также степень дивизора, такие, что эти кривые и дивизор могут быть использованы для построения искомого АГ-кода.

Коды строятся только на плоских кривых, поэтому сначала в цикле проверяем (шаги 3–5), возможно ли рассмотреть гладкую плоскую проективную кривую текущего

рода g. Согласно формуле (1), если уравнение $(x-1)(x-2) = 2g$ не имеет натуральных корней, то таких кривых не существует. Тогда увеличиваем g и осуществляем проверку, пока не будет выбран подходящий род g.

Затем, согласно лемме 3, выбираем число α как $\max(\lceil \log_q N \rceil + g - 1, 2g - 1)$ (шаг 6). Заметим, что α — минимальное из удовлетворяющих лемме 3 чисел, оно гарантирует дальнейшую минимизацию длины кода (замечание 8).

На шаге 7 выбираем константу $\hat{n}$, которая будет использована в качестве значения длины кода по умолчанию. Инициализируем её значением $c^2\alpha + 1$, так как это наименьшее (замечание 8) значение, при котором искомый АГ-код является c-ТА-кодом (теорема 1, условие (4)).

На шаге 8 запускается цикл по $\hat{n}$, пока $\hat{n}$ не достигнет минимального из значений $q+g\lfloor 2\sqrt{q} \rfloor$, $\lfloor B(N,c)/q+1 \rfloor$. Первое значение является максимальной мощностью кривой рода g над полем $\mathbb{F}_q$, уменьшенной на единицу (утверждение 4). Это значение также совпадает с максимально возможной длиной АГ-кода на кривой рода g над полем $\mathbb{F}_q$ (пока без учёта возможного добавления проверки чётности). Параметр $\hat{n}$ не может быть больше $\lfloor B(N,c)/q + 1 \rfloor$ по утверждению 1. Значит, цикл перебирает возможные значения $\hat{n}$ как потенциальной длины АГ-кода.

В цикле на шагах 9–10 выбираем гладкую плоскую кривую X рода g, такую, что $|X| \geqslant \hat{n}$. Это ограничение достаточно для того, чтобы утверждать, что при первом входе в цикл построенный на кривой X одноточечный АГ-код (замечание 5) может быть c-ТА-кодом (с учётом добавления проверки чётности, лемма 11), а при последующих итерациях — чтобы обеспечивать также выполнение условий на некоторые списочные декодеры (например, на классический СДСГ на шагах 28 и 31). Если такие кривые выбрать невозможно, то есть не существует X, такого, что $|X| \geqslant \hat{n}$, то выходим из цикла по $\hat{n}$, переходим на шаг 40 и увеличиваем род g.

Далее определяем, необходимо и возможно ли использование проверки чётности [18, п. 1.9(I)]. Если на кривой уже более $\hat{n}$ точек, то использовать проверку чётности не нужно, количества точек на кривой хватит для построения одноточечного АГ-кода (замечание 5), поэтому мы не удлиняем код, присваиваем ParityCheck значение `false`, а константе n — значение $\hat{n}$ (шаги 13–15). Если на кривой не больше $\hat{n}$ точек, код строится над двоичным полем и конструктивное расстояние строящегося кода нечётно, то возможно использование проверки чётности (лемма 11). В этом случае мы удлиняем код, присваиваем ParityCheck значение `true`, а n — значение $\hat{n} + 1$. Таким образом, мы сначала построим АГ-код с длиной $\hat{n}$, а затем удлиним его до длины n.

В противном случае, когда на кривой не больше $\hat{n}$ точек, но невозможно использование проверки чётности, переходим на новую итерацию цикла на шаге 8, так как количества точек на кривой не хватит для построения искомого кода. Далее в качестве длины кода используется константа n.

После этого начинаем проверку выполнения условий использования методов списочного декодирования. Инициализируем список OutputDecoders пустым списком. Если во входном списке декодеров есть метка мягкого СДСГ, то проверяем выполнение условия его использования (9) на шаге 24. Если условие выполнено, то помещаем его метку в выходной список декодеров.

Аналогично, если во входном списке декодеров есть метка классического СДСГ, то проверяем выполнение условия его использования (6) или (7) на шагах 27–32. Для этого сначала анализируем выполнение условия $\sqrt{n\alpha} \notin \mathbb{N}$. Если $\sqrt{n\alpha} \notin \mathbb{N}$, то необходимо подобрать параметры к условиям теоремы 2, п. 1. Если на шаге 28 выполняется (6), то возможно построение АГ-кода, который вместе с классическим СДСГ может быть

использован для построения (N, c)-ССШШ. В этом случае помещаем метку классического СДСГ в OutputDecoders. Если $\sqrt{n\alpha} \in \mathbb{N}$, то необходимо подобрать параметры к условиям теоремы 2, п. 2. Если на шаге 31 выполняется (7), то также помещаем метку классического СДСГ в OutputDecoders.

На шаге 33 можно проверить выполнение условий для иных методов списочного декодирования.

На шаге 34 если список OutputDecoders не пуст, то существует списочный декодер, который применим для построения (N, c)-ССШШ совместно с одноточечным АГ-кодом на кривой X. Тогда выходим из циклов на шаг 41. Если список пуст, то увеличиваем $\hat{n}$ и продолжаем поиски.

Если поиски окажутся неудачными, то n станет так велико, что мы покинем вложенный цикл (проверка на шаге 8), затем пройдём проверку на эффективность проектируемой ССШШ (шаг 38) и, если она будет пройдена, увеличим род кривой g (шаг 40). Далее опять начнём искать кривые, но уже большего рода, а значит, с бóльшим количеством точек, что видно из утверждения 4 (при $g \to \infty$ количество точек на кривой стремится к ∞). В конце концов, увеличивая род и мощность кривых, мы можем найти кривую, для которой выполнятся все необходимые условия. Если этого не произойдёт, то мы либо достигнем для длины кода значения $\lfloor B(N, c)/q + 1 \rfloor$, либо исчерпаем все возможные значения рода g из G. Это будет означать, что рассматриваемые достаточные условия не гарантируют существования искомого АГ-кода.

Отметим, что в цикле мы найдём минимальное подходящее значение n (замечание 8), так как n на каждой итерации изменяется максимум на единицу и увеличивается ровно настолько, насколько это необходимо для выполнения накладываемых условий.

После выбора параметров выбираем точку Q (шаг 41). В замечаниии 5 указаны рекомендации к выбору этой точки. Затем выбираем множество точек кривой X, отличных от Q. Обозначим его как $Points$ (шаги 42–45). Мощность этого множества равна $n - 1$, если ParityCheck=`true`, и n в противном случае. Такое множество всегда существует, так как в случае ParityCheck=`true` кривая X выбрана так, чтобы $|X| \geqslant n$ (шаги 10, 16), в противном случае $|X| \geqslant n + 1$ (шаги 10, 13).

Затем на шаге 46 объявляем дивизор $D = \alpha Q$, который будет использован в качестве дивизора искомого АГ-кода.

Так как по построению $\alpha > 2g - 2$, размерность пространства Римана — Роха $L(D)$, ассоциированного с дивизором D, равна $\alpha - g + 1$ (утверждение 7). В алгоритме размерность $L(D)$ обозначена как k (шаг 47). Зная размерность, вычисляем базис пространства $L(D)$ — набор из k линейно независимых функций из $\mathbb{F}_q(X)$. Зная базис, легко написать порождающую матрицу АГ-кода: i-я строка матрицы есть результат действия отображения Ev_{Points} (2) на i-й элемент базиса (шаг 49). Длина такого кода равна $|Points|$. Поэтому если ParityCheck=`true`, то на шаге 51 в матрицу G добавляем столбец проверки чётности. Таким образом, длина кода станет равна выбранному параметру n.

На последнем шаге возвращаем матрицу G и выходной список декодеров. ■

Теорема 7. Алгоритм 1 полиномиально сводится к алгоритму вычисления базиса пространства Римана — Роха.

Доказательство. Решение неравенств 1–4 из теоремы 5 выполняется за константное число шагов. Цикл по g, начинающийся на шаге 2, работает полиномиальное от входных параметров c, N, q время, так как в нём перебираются значения мно-

жества G, мощность которого ограничена значениями многочленов от c, N, q (теорема 5). После шагов 3–7 начинается цикл по $\hat{n}$ на шаге 8, который также имеет полиномиальное от c, N, q число шагов — не более $\min\{q + g\lfloor 2\sqrt{q}\rfloor, \lfloor B(N,c)/q + 1\rfloor\}$, где $g \leqslant \max\{x : x \in G\}$. Внутри этого цикла производится поиск кривых в списке L, длина которого не зависит от входных параметров, проверка необходимости и возможности использования проверки чётности, а также проверка условий для декодеров из списка InputDecoders, длина которого является константой. После выхода из циклов или ошибки на шаге 38 производятся только тривиальные присвоения и объявления переменных, за исключением шага 41, где необходимо построить базис пространства Римана — Роха (замечание 4). ■

Для построения АГ-кода L-конструкции необходимо работать с проективной кривой. Однако при выборе кривой в алгоритме возможно рассмотрение сначала аффинных кривых, а затем использование их проективизаций.

Замечание 9. Выделим некоторые частные случаи АГ-кодов, которые могут быть получены по алгоритму. Известно, что все АГ-коды L-конструкции на кривых рода ноль являются или кодами Рида — Соломона, или их удлинениями [9, с. 314]. Количество точек на кривой рода 0 равно $q+1$ (утверждение 4). Это влечёт ограничение на длину АГ-кода, построенного по алгоритму 1: $n \leqslant q$. В этом случае достаточное условие того, что одноточечный АГ-код является c-ТА-кодом, выглядит следующим образом:

$$c < \sqrt{q/\alpha}.$$

Учитывая утверждение 4, получим, что достаточное условие того, что одноточечный АГ-код на кривой рода 1 с максимальным числом точек является c-ТА-кодом, имеет вид

$$c < \sqrt{(q + \lfloor 2\sqrt{q}\rfloor)/\alpha}.$$

Известно, что все АГ-коды L-конструкции на кривых рода 1 являются эллиптическими кодами, то есть кодами, построенными по эллиптическим кривым [9, п. 4.4.1, разд. «Коды рода один»].

Данная работа преимущественно посвящена построению АГ-кодов безотносительно методов их списочного декодирования. Несмотря на это, для более полного исследования получены также достаточные условия применимости списочных декодеров [7, 8] (теоремы 2, 3). Достаточность выполнения этих условий отражена в алгоритме 1 на шагах 23–32.

При применении классического СДСГ обычно используется следующая модель канала: отправитель посылает $x \in \mathbb{F}_q$, получатель получает некоторый $y \in \mathbb{F}_q$, и вероятность того, что $x = y$, равна $1-p$, где p — вероятность ошибки в канале. Классический СДСГ выбран для исследования, потому что он, во-первых, достигает границы Джонсона, которая является нижним порогом списочной «декодируемости» кода, то есть классический СДСГ удовлетворяет минимальным требованиям к списочному декодеру; во-вторых, применение классического СДСГ достаточно для построения ССШШ, более эффективных с точки зрения затрат памяти (утверждение 1), чем тривиальные схемы широковещательного шифрования или чем КСШШ (см. п. 2.3).

Мягкий СДСГ является расширением классического СДСГ. При его применении используется следующая модель канала: отправитель посылает $x \in \mathbb{F}_q$, получатель на своём конце имеет функции плотности вероятности $f_x(\cdot)$, где $f_x(a)$ — вероятность того,

что получатель получит на выходе значение a в случае, когда отправитель послал x. Построение таких функций в случае декодирования АГ-кодов описано в [8].

На шаге 33 алгоритма 1 могут быть размещены условия на иные методы списочного декодирования для АГ-кодов. При использовании этих методов они должны быть включены в список InputDecoders на входе алгоритма. Например, возможно использование списочного декодера, представленного в [16]. В данную работу рассмотрение этого алгоритма, имеющего радиус декодирования $n-\lceil\sqrt{2n(n-d)}+g-1\rceil-1$, не включено, так как он меньше аналогичного радиуса для классического СДСГ [7, введение в разд. 4].

Сравнение различных методов списочного декодирования относительно использования в ССШШ, в том числе сравнение характеристик классического и мягкого СДСГ, представляется темой отдельного исследования.

Отметим, что одноточечные АГ-коды на фиксированной кривой имеют максимально возможную длину. Это условие является важным с точки зрения выполнения достаточного условия c-ТА (4).

2.3. Пример работы алгоритма

Пусть на вход алгоритма поданы следующие параметры:

$$c = 2,\ N = 512,\ q = 8.$$

В качестве списка кривых L будем использовать таблицы кривых из [17]. В список декодеров занесём метки классического и мягкого СДСГ, рассматриваемых в работе.

Применим алгоритм 1. Сначала покажем, что достаточные условия c-ТА и условия применимости СДСГ не гарантируют существование q-ичных кодов Рида — Соломона, применимых для построения $(512, 2)$-ССШШ. Затем построим одноточечный АГ-код на кривой рода 1, который вместе с мягким СДСГ может быть применён для организации $(512, 2)$-ССШШ, и построим одноточечный АГ-код на кривой рода 3, который может быть применён вместе с классическим СДСГ.

На шаге 1 сначала вычисляем величину $B(N, c) = N = 512$. Затем определяем множество возможных значений рода кривой g. После решения неравенств получаем

$$G = \{1, 2, 3\} \cup \{3\} \cup \varnothing \cup \varnothing = \{1, 2, 3\}.$$

Так как значение $g = 0$ не входит в множество G, можно сделать вывод, что при поданных на вход параметрах достаточные условия c-ТА не гарантируют существования искомого АГ-кода на кривой рода 0. В силу замечания 9, это означает, что достаточные условия не гарантируют существование кодов Рида — Соломона, которые вместе с СДСГ могут быть применены для организации $(512, 2)$-ССШШ.

Полагаем $g = 1$ и заходим в цикл (шаг 2). Уравнение $(x-1)(x-2) = 2$ имеет натуральный корень 3, поэтому можем рассматривать плоские кривые рода 1. На шагах 6 и 7 присваиваем $\alpha := \max\{\lceil\log_8 512+1-1\rceil, 2\cdot 1-1\} = 3$ и $\hat{n} := 2^2\cdot 3+1 = 13$. Затем заходим в цикл по $\hat{n}$ на шаге 8, так как выполняются условия $\hat{n} = 13 \leqslant q+g\lfloor 2\sqrt{q}\rfloor+1 = 14$ и $\hat{n} = 13 \leqslant \lfloor B(N, c)/q + 1\rfloor = 65$. В цикле на шаге 10 в качестве кривой X выберем кривую, заданную следующим многочленом:

$$X_2^2X_3 + X_1X_2X_3 + X_2X_3^2 - X_1^3 - X_3^3 = 0. \tag{13}$$

Можно проверить, что на этой кривой находится 14 точек над полем $\mathbb{F}_8$, таким образом, она удовлетворяет всем условиям, наложенным на шаге 9. Так как $|X| \geqslant \hat{n} = 13$,

присваиваем ParityCheck:= `false` и $n := \hat{n} = 13$. На шаге 22 инициализируем список возможных для использования декодеров. Затем, так как на шаге 23 видим, что SoftDecoder $\in$ InputDecoders, переходим на шаг 24, где проверяем условие (9) и убеждаемся в том, что оно выполнено: $c = 2 \geqslant (13 - 7 - 1)/3 \geqslant 5/3$. На шаге 26 видим, что ClassicDecoder $\in$ InputDecoders; анализируя на следующем шаге значение величины $\sqrt{n\alpha} = \sqrt{13 \cdot 3}$, убеждаемся, что оно не принадлежит множеству $\mathbb{N}$, и на шаге 27 проходим по первой ветке. На шаге 28 видим, что условие (6) не выполнено. На шаге 37, так как на шаге 34 обнаружено, что список возможных декодеров не пуст, выходим из цикла на шаг 41. Таким образом, все необходимые параметры для построения кода подобраны.

Рассмотрим кривую (13). Поле $\mathbb{F}_8$ будем рассматривать как $\mathbb{F}_2[\xi]/(\xi^3 + \xi + 1)$. Выпишем все точки кривой:

$$Q = (0:1:0), P_1 = (1:0:1), P_2 = (\xi^4 : \xi^4 : 1), P_3 = (\xi^2 : \xi^2 : 1),$$
$$P_4 = (\xi : \xi : 1), P_5 = (\xi^3 : \xi^4 : 1), P_6 = (\xi^6 : \xi : 1), P_7 = (\xi^5 : \xi : 1), P_8 = (\xi^4 : 1 : 1),$$
$$P_9 = (\xi^5 : \xi^2 : 1), P_{10} = (\xi^6 : \xi^4 : 1), P_{11} = (\xi^2 : 1 : 1), P_{12} = (\xi^3 : \xi^2 : 1), P_{13} = (\xi : 1 : 1).$$

В качестве точки Q выберем точку $(0 : 1 : 0)$ (шаг 41), а в качестве $Points$ — множество $\{P_i : 1 \leqslant i \leqslant 13\}$ (шаг 45). Объявим дивизор $D = 3Q$ (шаг 46) и $l = 3 - 1 + 1 = 3$ (шаг 47).

На шаге 48 необходимо вычислить базис пространства Римана — Роха. Вычислим сначала дивизоры пересечения $X_{\text{pr}} \cdot X_1$, $X_{\text{pr}} \cdot X_2$, $X_{\text{pr}} \cdot X_3$, используя соображения, изложенные в замечании 4.

Многочлен X_1 равен нулю на кривой X_{pr} только в точке Q. Тогда, в силу теоремы 6, $I(P_1; X_{\text{pr}}; X_1) = 3$. Значит, $X_{\text{pr}} \cdot X_1 = 3Q$.

Многочлен X_2 равен нулю на кривой X_{pr} только в точке P_1. Тогда, в силу теоремы 6, $I(P_2; X_{\text{pr}}; X_1) = 3$. Значит, $X_{\text{pr}} \cdot X_2 = 3P_1$.

Многочлен X_3 равен нулю на кривой X_{pr} только в точке Q. Тогда, в силу теоремы 6, $I(Q; X_{\text{pr}}; X_3) = 3$. Значит, $X_{\text{pr}} \cdot X_3 = 3Q$.

Рассмотрим функции 1, X_1/X_3, X_2/X_3. Их количество совпадает с $l = 3$. В силу замечаний 1 и 2 имеем

$$(X_1/X_3) = 3Q - 3Q = 0, \ (X_2/X_3) = 3P_1 - 3Q.$$

Кроме того,

$$(1) + D = 3Q \geqslant 0, \quad (X_1/X_3) + D = 0 + 3Q = 3Q \geqslant 0,$$
$$(X_2/X_3) + D = 3P_1 - 3Q + 3Q = 3P_1 \geqslant 0,$$

то есть все эти функции лежат в пространстве $L(D)$. Можно проверить, что они линейно независимы в пространстве Римана — Роха $L(D)$. Значит, $\{1, X_1/X_3, X_2/X_3\}$ — полный линейно независимый набор функций из $L(D)$, то есть искомый базис. Построим порождающую матрицу G:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \xi^4 & \xi^2 & \xi & \xi^3 & \xi^6 & \xi^5 & \xi^4 & \xi^5 & \xi^6 & \xi^2 & \xi^3 & \xi \\ 0 & \xi^4 & \xi^2 & \xi & \xi^4 & \xi & \xi & 1 & \xi^2 & \xi^4 & 1 & \xi^2 & 1 \end{bmatrix}.$$

Так как ParityCheck = `false`, то на шагах 50–52 не выполняем никаких действий. Матрица G и список OutputDecoders = {SoftDecoder} с меткой мягкого СДСГ являются

результатом работы алгоритма. Получен $[13, 3, \geqslant 10]_8$-код, который совместно с мягким СДСГ может быть применён для организации $(512, 2)$-ССШШ.

Рассмотрим, какой код был бы получен в результате работы алгоритма, если бы на входе в список декодеров не был включен только жёсткий СДСГ. В этом случае мы бы не прошли проверку на шагах 23, 27 и 28, на шаге 34 получили пустой список OutputDecoders, увеличили $\hat{n}$ и перешли на шаг 8. При $g = 1$ на шаге 8 при $\hat{n} = 14$ не заходим на следующую итерацию, так как условие $\hat{n} = 14 \leqslant q + g\lfloor 2\sqrt{q} \rfloor = 13$ неверно. Поэтому переходим на проверку на шаге 38, проходим её и на шаге 40 увеличиваем значение рода: $g = 2$. При $g = 2$ уравнение на шаге 3 не имеет корней, поэтому увеличиваем g до 3.

Затем на шаге 6 присваиваем $\alpha := \max\{\lceil \log_8 512 + 3 - 1 \rceil, 2 \cdot 3 - 1\} = 5$, а на шаге 7 $\hat{n} := 2^2 \cdot 5 + 1 = 21$. Далее попадаем на начало цикла по $\hat{n}$ (шаг 8). Заходим в цикл, так как выполняются условия $\hat{n} = 21 \leqslant q + g\lfloor 2\sqrt{q} \rfloor = 23$ и $\hat{n} = 21 \leqslant \lfloor B(N, c)/q + 1 \rfloor = 65$. На шаге 10 в качестве кривой X выберем кривую, заданную следующим многочленом:

$$X_1^3 X_2 + X_2^3 X_3 + X_3^3 X_1 = 0. \tag{14}$$

На этой кривой находится 24 точки над полем $\mathbb{F}_8$, таким образом, она удовлетворяет всем условиям, наложенным на шаге 9.

Так как $|X| \geqslant 22$, присваиваем ParityCheck := `false` и $n := \hat{n} = 21$. Пропускаем проверку на шаге 23, так как SoftDecoder $\notin$ InputDecoders. На шаге 27 анализируем значение величины $\sqrt{n\alpha} = \sqrt{21 \cdot 5}$, убеждаемся, что она не принадлежит множеству $\mathbb{N}$. Значит, проходим по первой ветке на шаг 28 и видим, что условие (6) не выполнено. Тогда на шаге 35 увеличиваем значение $\hat{n}$ до 22 и переходим на новую итерацию цикла.

На шаге 10 можем выбрать ту же самую кривую. Снова присваиваем ParityCheck := := `false` и $n := \hat{n} = 22$. Пройдя все проверки, попадаем на шаг 27 и видим, что снова $\sqrt{n\alpha} = \sqrt{22 \cdot 5} \notin \mathbb{N}$. Значит, снова проходим по первой ветке на шаг 28. Там убеждаемся, что условие (6) выполнено, добавляем в OutputDecoders метку классического СДСГ ClassicDecoder и на шаге 37 выходим из циклов, попадая на шаг 41. Таким образом, все необходимые параметры для построения кода подобраны.

Рассмотрим кривую (14). Поле $\mathbb{F}_8$, как и ранее, будем рассматривать как $\mathbb{F}_2[\xi]/(\xi^3 + \xi + 1)$. Выпишем все точки кривой:

$$P_1 = (1 : 0 : 0),\ Q = (0 : 1 : 0),\ P_2 = (0 : 0 : 1),\ P_3 = (1 : \xi : 1),\ P_4 = (1 : \xi^2 : 1),$$
$$P_5 = (1 : \xi^4 : 1),\ P_6 = (\xi : 1 : 1),\ P_7 = (\xi : \xi^2 : 1),\ P_8 = (\xi : \xi^6 : 1),\ P_9 = (\xi^2 : 1 : 1),$$
$$P_{10} = (\xi^2 : \xi^4 : 1),\ P_{11} = (\xi^2 : \xi^5 : 1),\ P_{12} = (\xi^3 : \xi^2 : 1),\ P_{13} = (\xi^3 : \xi^3 : 1),\ P_{14} = (\xi^3 : \xi^5 : 1),$$
$$P_{15} = (\xi^4 : 1 : 1),\ P_{16} = (\xi^4 : \xi : 1),\ P_{17} = (\xi^4 : \xi^3 : 1),\ P_{18} = (\xi^5 : \xi : 1),\ P_{19} = (\xi^5 : \xi^5 : 1),$$
$$P_{20} = (\xi^5 : \xi^6 : 1),\ P_{21} = (\xi^6 : \xi^3 : 1),\ P_{22} = (\xi^6 : \xi^4 : 1),\ P_{23} = (\xi^6 : \xi^6 : 1).$$

В качестве Q выберем точку $(0 : 1 : 0)$, в качестве $Points$ — множество $\{P_i : 2 \leqslant i \leqslant 23\}$. Объявим дивизор $D = 5Q$ и $l = 5 - 3 + 1 = 3$.

Затем необходимо вычислить базис пространства Римана — Роха. Вычисляем дивизоры пересечения $X_{\text{pr}} \cdot X_1$, $X_{\text{pr}} \cdot X_2$, $X_{\text{pr}} \cdot X_3$.

Многочлен X_1 равен нулю на кривой X_{pr} только в точках P_2 и Q. Так как $X_1 = 0$ не является касательной к кривой в точке Q, то $I(Q; X_{\text{pr}}; X_1) = 1$. Тогда, в силу теоремы 6, $I(P_2; X_{\text{pr}}; X_1) = 3$. Значит, $X_{\text{pr}} \cdot X_1 = 3P_2 + Q$.

Многочлен X_2 равен нулю на кривой X_{pr} только в точках P_1 и P_2. Так как $X_2 = 0$ не является касательной к кривой в точке P_2, то $I(P_2; X_{\text{pr}}; X_2) = 1$. Тогда, в силу теоремы 6, $I(P_1; X_{\text{pr}}; X_1) = 3$. Значит, $X_{\text{pr}} \cdot X_2 = 3P_1 + P_2$.

Многочлен X_3 равен нулю на кривой X_{pr} только в точках P_1 и Q. Так как $X_3 = 0$ не является касательной к кривой в точке P_1, то $I(P_1; X_{\mathrm{pr}}; X_3) = 1$. Тогда, в силу теоремы 6, $I(Q; X_{\mathrm{pr}}; X_3) = 3$. Значит, $X_{\mathrm{pr}} \cdot X_3 = 3Q + P_1$.

Рассмотрим функции 1, X_2/X_3, X_1X_2/X_3^2. Их количество совпадает с $l = 3$. В силу замечаний 1 и 2 имеем

$$(X_1/X_3) = 3P_2 - P_1 - 2Q, \quad (X_2/X_3) = 2P_1 + P_2 - 3Q$$

и, кроме того,

$$(1) + D = 5Q \geqslant 0, \quad (X_2/X_3) + D = 2P_1 + P_2 - 3Q + 5Q = 2P_1 + P_2 + 2Q \geqslant 0,$$
$$\left(X_1X_2/X_3^2\right) + D = (X_1/X_2) + (X_2/X_3) + D =$$
$$= 3P_2 - P_1 - 2Q + 2P_1 + P_2 - 3Q + 5Q = P_1 + 4P_2 \geqslant 0,$$

то есть все эти функции лежат в пространстве $L(D)$. Можно проверить, что они линейно независимы в пространстве Римана — Роха $L(D)$. Значит, $\{1, X_2/X_3, X_1X_2/X_3^2\}$ — полный линейно независимый набор функций из $L(D)$, то есть искомый базис.

Построим порождающую матрицу G:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & \xi & \xi^2 & \xi^4 & 1 & \xi^2 & \xi^6 & 1 & \xi^4 & \xi^5 & \xi^2 & \xi^3 & \xi^5 & 1 & \xi & \xi^3 & \xi & \xi^5 & \xi^6 & \xi^3 & \xi^4 & \xi^6 \\ 0 & \xi & \xi^2 & \xi^4 & \xi & \xi^3 & 1 & \xi^2 & \xi^6 & 1 & \xi^5 & \xi^6 & \xi & \xi^4 & \xi^5 & 1 & \xi^6 & \xi^3 & \xi^4 & \xi^2 & \xi^3 & \xi^5 \end{bmatrix}.$$

Так как ParityCheck = `false`, на шагах 50–52 не выполняем никаких действий. Матрица G является результатом работы алгоритма в случае, если в списке InputDecoders только классический СДСГ. Получен $[22, 3, \geqslant 17]_8$-код, который совместно с классическим СДСГ может быть применён для организации $(512, 2)$-ССШШ.

2.4. Сравнение АГ-кодов с другими классами кодов

Выгода в затратах памяти при использовании АГ-кодов на кривых высоких родов обсуловлена тем, что, в отличие от классических кодов Рида — Соломона и Рида — Маллера, длина АГ-кода ограничена не мощностью поля, а лишь мощностью кривой, на которой строится код и которая может быть существенно больше, чем мощность поля [9, теорема 3.1.25]. Таким образом, можно строить ССШШ над полями меньшей мощности, незначительно увеличивая n, в то время как при использовании классических кодов приходится увеличивать мощность поля. Наиболее наглядно это можно видеть в случаях, когда $q = 2^m$ для некоторого m. Отметим, что именно такие поля наиболее часто используются в технических и программных реализациях схем защиты.

Рассмотрим некоторые примеры ССШШ, на которых сравним классические коды Рида — Соломона и Рида — Маллера с АГ-кодами. Покажем, что существуют как параметры ССШШ, при которых более выгодно использование исключительно классических кодов, так и параметры, при которых более выгодно использование АГ-кодов на кривых рода больше 0.

Пример 1. Как показано в рассмотренном в п. 2.3 примере, достаточные условия не гарантируют построения кода Рида — Соломона над полем $\mathbb{F}_8$, применимого в ССШШ с параметрами $c = 2$, $N = 512$. Для обслуживания такой ССШШ построены $[13, 3, \geqslant 10]_8$- и $[22, 3, \geqslant 17]_8$-коды на кривых рода 1 и 3 соответственно, которые должны применяться с мягким СДСГ в первом случае и с классическим СДСГ во втором. Заметим, что для этих кодов величины nq, важные с точки зрения затрачиваемой в ССШШ памяти (см. таблицу на с. 70), равны 104 и 176 соответственно.

Попробуем построить классические коды, применимые для организации такой схемы. Так, для этих параметров возможно построение $[16, 4, 13]_{16}$-кода Рида — Соломона. Однако для этого кода $nq = 256$, что больше, чем при использовании АГ-кодов на кривых рода 1 и 3, следовательно, хуже с точки зрения объёма используемой памяти.

При заданных параметрах можно построить 2-ТА-код Рида — Маллера с $q = 8$, $r = 1$, $m = 2$. Для этого кода $n = 8^2 = 64$, $k = 3$, $d = 8^2 - 1 \cdot 8 = 58$, поэтому он имеет самое большое значение $nq = 512$.

Таким образом, для рассмотренных в примере значений c и N использование АГ-кодов более выгодно, чем классических кодов Рида — Соломона и Рида — Маллера.

Пример 2. В [8, п. «Example»] рассмотрен пример ССШШ с параметрами $c = 3$, $N = 2^{20} = 1\,048\,576$. Для неё в [8] построен $[31, 4, 28]_{32}$-код Рида — Соломона, который применим в таких ССШШ совместно как с классическим, так и с мягким СДСГ. Заметим, что алгоритм 1 также выдаст для таких значений c, N, q этот код. Действительно, для этих параметров выбор такого кода представляется наиболее удачным, так как построение АГ-кода над полем $\mathbb{F}_{32}$ на кривых более высоких родов с целью оптимизации памяти не имеет смысла: увеличение рода кривых приведёт только к увеличению n, а уменьшение величины q до 16, например, не удастся, так как для кодов, построенных при параметрах $c = 3$, $N = 2^{20}$, $q = 16$, перестанет выполняться достаточное условие c-ТА (4).

Для обслуживания такой ССШШ также возможно построение 3-ТА $(1, 2)$-кода Рида — Маллера над полем $\mathbb{F}_{16}$, но такой код показывает большую неэффективность в использовании памяти: учитывая, что $n = q^m$, для него $nq = 256 \cdot 16 = 4096$.

Значит, для рассмотренных в примере параметров использование АГ-кодов представляется менее выгодным, чем использование классического кода Рида — Соломона.

Пример 3. Пусть поставлена задача построения ССШШ с параметрами $N = 2^{20} + 1$, $c = 3$. В этом случае код Рида — Соломона над полем $\mathbb{F}_{32}$ построить уже не удастся. Действительно, для обслуживания такого числа пользователей с помощью кода Рида — Соломона над полем $\mathbb{F}_{32}$ величина $q^k = 32^k$ должна быть больше, чем 2^{20} (утверждение 2). Это накладывает ограничение на размерность кода: $k > 4$. Но уже при $k = 5$ получаем, что $d = n - 4$. Это ограничение совместно с достаточным условием c-ТА (4) налагает на длину кода Рида — Соломона условие $n > 36$. Но код Рида — Соломона не может иметь такой длины над полем $\mathbb{F}_{32}$. Таким образом, для такой ССШШ возможно построение лишь $[64, 4, 61]_{16}$-кода Рида — Соломона, для которого $nq = 4096$. Ситуация аналогична для всех N, таких, что $2^{20} < N < 2^{24}$.

Легко проверить, что с помощью алгоритма 1 для таких параметров АГ-код над полем $\mathbb{F}_{32}$ можно построить только при условии существования над $\mathbb{F}_{32}$ кривой рода 3 с максимальным числом точек, в частности с 65 точками. Тогда возможно построение $[64, 5, \geqslant 57]_{32}$-АГ-кода, который может быть применён в такой ССШШ совместно с мягким СДСГ. Однако в ходе исследования такой кривой найти не удалось. Известные кривые рода 3 над полем $\mathbb{F}_{32}$ имеют максимум 64 точки [17], что недостаточно для построения искомого АГ-кода даже с использованием столбца проверки чётности.

Значит, для заданных параметров использование АГ-кодов менее выгодно, чем классических кодов.

Пример 4. Пусть поставлена задача построения ССШШ с параметрами

$$2^{35} < N < 2^{36},\ c = 2.$$

Для обслуживания такого числа пользователей с помощью классического кода Рида — Соломона необходимо использовать поле $\mathbb{F}_{32}$. Действительно, для поля $\mathbb{F}_{16}$ получаем,

что величина $q^k = 16^k$ должна быть больше, чем 2^{35} (утверждение 2). Это накладывает ограничение на размерность кода: $k > 8$. Уже при $k = 9$ получаем, что $d = n - 8$. Это ограничение совместно с достаточным условием c-ТА (4) налагает на длину кода Рида — Соломона условие $n > 32$. Но код Рида — Соломона не может иметь такой длины над полем $\mathbb{F}_{16}$. Таким образом, для такой ССШШ возможно построение лишь $[29, 8, 22]_{32}$- кода Рида — Соломона, для которого $nq = 928$. Он может быть применён в этой ССШШ совместно с мягким СДСГ (при параметре допустимого отклонения не более 0,1).

С помощью алгоритма 1 для таких же параметров возможно построить АГ-код над полем $\mathbb{F}_{16}$ на кривой рода 6 с 65 точками [17]. Получим $[57, 9, \geqslant 43]_{16}$-АГ-код, который может быть применён в такой ССШШ совместно с мягким СДСГ (при параметре допустимого отклонения не более 0,1). Для такого кода $nq = 912$.

Для применения классического СДСГ можно использовать $[30, 8, 23]_{32}$-код Рида — Соломона с $nq = 960$ или $[58, 9, \geqslant 44]_{16}$-АГ-код с $nq = 928$.

Возможно также построение 2-ТА $(2, 3)$-кода Рида — Маллера над полем $\mathbb{F}_q$, но, учитывая, что для него длина равна $n = q^m = 16^3$, можно заключить, что его использование является наименее эффективным относительно затрат памяти.

Таким образом, для заданных параметров использование АГ-кодов более выгодно, чем классических.

Заключение

В [3] представлены достаточные условия эффективного использования линейных кодов и методов списочного декодирования в ССШШ. В работе эти результаты применены для АГ-кодов L-конструкции [6, 9, 12] и классического и мягкого алгоритмов СДСГ [7, 8]. Таким образом, получены точные достаточные условия применения этих кодов и декодеров в ССШШ, исследована связь между ними.

На основе полученных утверждений представлен концептуальный алгоритм построения АГ-кода L-конструкции, который, совместно с алгоритмами СДСГ, может быть эффективно использован в ССШШ.

ЛИТЕРАТУРА

1. *Stinson D. R. and Wei R.* Combinatorial properties and constructions of traceability schemes and frameproof codes // SIAM J. Discr. Math. 1998. V. 11. No. 1. P. 41–53.
2. *Staddon J. N., Stinson D. R., and Wei R.* Combinatorial properties of frameproof and traceability codes // IEEE Trans. Inform. Theory. 2001. V. 47. No. 3. P. 1042–1049.
3. *Silverberg A., Staddon J., and Walker J.* Applications of list decoding to tracing traitors // IEEE Trans. Inform. Theory. 2003. V. 49. No. 5. P. 1312–1318.
4. *Деундяк В. М., Мкртичян В. В.* Исследование границ применения схемы защиты информации, основанной на РС-кодах // Дискретный анализ и исследование операций. 2011. Т. 18. № 3. С. 21–38.
5. *Деундяк В. М., Евпак С. А., Мкртичян В. В.* Исследование свойств q-ичных помехоустойчивых кодов Рида — Маллера как кодов для защиты от копирования // Проблемы передачи информации. 2015. Т. 51. № 4. С. 99–111.
6. *Гоппа В. Д.* Алгебраико-геометрические коды // Известия АН СССР. Сер. матем. 1982. Т. 46. № 4. С. 762–781.
7. *Guruswami V. and Sudan M.* Improved decoding of Reed — Solomon and algebraic-geometric codes // Foundations of Computer Science. Palo Alto: IEEE, 1998. P. 28–37.

8. *Fernandez M. and Soriano M.* Identification of traitors in algebraic-geometric traceability codes // IEEE Trans. Signal Proc. 2004. V. 52. Iss. 10. P. 3073–3077.
9. *Влэдуц С. Г., Ногин Д. Ю., Цфасман М. А.* Алгеброгеометрические коды. Основные понятия. М.: МЦНМО, 2003. 504 с.
10. *Fiat A. and Naor M.* Broadcast encryption // LNCS. 1994. V. 773. P. 480–491.
11. *Chor B., Fiat A., and Naor M.* Tracing traitors // LNCS. 1994. V. 839. P. 257–270.
12. *Hoholdt T., van Lindt J., and Pellikaan R.* Algebraic geometry codes // Handbook of Coding Theory / eds. V. S. Pless, W. C. Huffman, and R. A. Brualdi. V. 1. Amsterdam: Elsevier, 1998. P. 871–961.
13. *Лэнг С.* Алгебра. М.: Мир, 1968. 564 с.
14. *Hess F.* Computing Riemann — Roch spaces in algebraic function fields and related topics // J. Symbolic Comput. 2002. V. 33. No. 4. P. 425–445.
15. `http://magma.maths.usyd.edu.au/magma/` — Magma Computational Algebra System.
16. *Shokrollahi A. and Wasserman H.* List decoding of algebraic-geometric codes // IEEE Trans. Inform. Theory. 1999. V. 45. No. 2. P. 432–437.
17. *Van Der Geer G. and Van Der Vlugt M.* Tables of curves with many points // Mathematics of Computation. 2000. V. 69. No. 230. P. 797–810.
18. *Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А.* Теория кодов, исправляющих ошибки. М.: Связь, 1979. 744 с.

REFERENCES

1. *Stinson D. R. and Wei R.* Combinatorial properties and constructions of traceability schemes and frameproof codes. SIAM J. Discr. Math., 1998, vol. 11, no. 1, pp. 41–53.
2. *Staddon J. N., Stinson D. R., and Wei R.* Combinatorial properties of frameproof and traceability codes. IEEE Trans. Inform. Theory, 2001, vol. 47, no. 3, pp. 1042–1049.
3. *Silverberg A., Staddon J., and Walker J.* Applications of list decoding to tracing traitors. IEEE Trans. Inform. Theory, 2003, vol. 49, no. 5, pp. 1312–1318.
4. *Deundyak V. M. and Mkrtichyan V. V.* Issledovaniye granits primeneniya skhemy zashchity informatsii, osnovannoy na RS-kodakh [Research of applying bounds of the information protection scheme based on RS-codes]. Diskretn. Anal. Issled. Oper., 2011, vol. 18, no. 3, pp. 21–38. (in Russian)
5. *Deundyak V. M., Evpak S. A., and Mkrtichyan V. V.* Analysis of properties of q-ary Reed — Muller error-correcting codes viewed as codes for copyright protection. Probl. Inform. Transm., 2015, vol. 51, no. 4, pp. 398–408.
6. *Goppa V. D.* Algebraiko-geometricheskiye kody [Algebraic-geometric codes]. Izv. Akad. Nauk SSSR. Ser. Mat., 1982, vol. 46, no. 4, pp. 762–781. (in Russian)
7. *Guruswami V. and Sudan M.* Improved decoding of Reed — Solomon and algebraic-geometric codes. Foundations of Computer Science, Palo Alto, IEEE, 1998, pp. 28–37.
8. *Fernandez M. and Soriano M.* Identification of traitors in algebraic-geometric traceability codes. IEEE Trans. Signal Proc., 2004, vol. 52, iss. 10, pp. 3073–3077.
9. *Vladut S. G., Nogin D. Y., and Tsfasman M. A.* Algebrogeometricheskie kody. Osnovnye ponyatiya. [Algebraic Geometric Codes. Basic Notions]. Moscow, MCCME Publ., 2003. 504 p. (in Russian)
10. *Fiat A. and Naor M.* Broadcast encryption. LNCS, 1994, vol. 773, pp. 480–491.
11. *Chor B., Fiat A., and Naor M.* Tracing traitors. LNCS, 1994, vol. 839, pp. 257–270.
12. *Hoholdt T., van Lindt J., and Pellikaan R.* Algebraic geometry codes. Handbook of Coding Theory, eds. V. S. Pless, W. C. Huffman, and R. A. Brualdi, vol. 1. Amsterdam, Elsevier, 1998, pp. 871–961.

13. *Lang S.* Algebra. Springer, 2002.
14. *Hess F.* Computing Riemann — Roch spaces in algebraic function fields and related topics. J. Symbolic Computation, 2002, vol. 33, no. 4, pp. 425–445.
15. `http://magma.maths.usyd.edu.au/magma/` — Magma Computational Algebra System.
16. *Shokrollahi A. and Wasserman H.* List decoding of algebraic-geometric codes. IEEE Trans. Inform. Theory, 1999, vol. 45, no. 2, pp. 432–437.
17. *Van Der Geer G. and Van Der Vlugt M.* Tables of curves with many points. Mathematics of Computation, 2000, vol. 69, no. 230, pp. 797–810.
18. *MacWilliams F. J. and Sloane N. J. A.* The Theory of Error-Correcting Codes. Elsevier, 1977. 744 p.