ЖЭТФ, 2019, том 155, вып. 4, стр. 645–653

ОЦЕНКА СЛОЖНОСТИ РЕАЛИЗАЦИИ АЛГОРИТМА ГРОВЕРА ДЛЯ ПЕРЕБОРА КЛЮЧЕЙ АЛГОРИТМОВ БЛОЧНОГО ШИФРОВАНИЯ ГОСТ Р 34.12-2015

Д. В. Денисенко [a,b*], ***Г. Б. Маршалко*** [b**], ***М. В. Никитенкова*** [b***], ***В. И. Рудской*** [b****], ***В. А. Шишкин*** [b†]

[a] *Московский государственный технический университет им. Н. Э. Баумана*
105005, Москва, Россия

[b] *Технический комитет по стандартизации ТК 26*
127287, Москва, Россия

Поступила в редакцию 23 октября 2018 г.,
после переработки 23 октября 2018 г.
Принята к публикации 29 октября 2018 г.

В рамках подхода, предложенного в работе [3], исследуется вопрос оценки необходимых ресурсов квантового вычислителя для решения задачи поиска ключей алгоритмов блочного шифрования «Кузнечик» и «Магма», определяемых национальным стандартом ГОСТ Р 34.12-2015, с использованием квантового алгоритма Гровера.

DOI: 10.1134/S0044451019040072

1. ВВЕДЕНИЕ

Теория квантовых вычислений стремительно развивается с конца прошлого века. К настоящему моменту построен ряд формальных моделей квантовых вычислений, для которых показано, что квантовая природа объектов, с использованием которых проводятся вычисления, позволяет эффективно решать определенные задачи, имеющие непосредственное отношение к стойкости криптографических примитивов. К таким задачам относится задача о скрытой подгруппе, возникающая при анализе различных схем асимметричного шифрования и цифровой подписи. Частным случаем этой задачи является факторизация целых чисел, эффективное решение которой предлагает алгоритм Шора [1]. Другим важным приложением квантовых вычислений является задача поиска в неупорядоченном массиве данных, которая эффективно решается с использованием алгоритма Гровера [2] и может быть использована при криптоанализе алгоритмов шифрования [3–5].

В настоящее время фундаментальные исследования направлены на создание квантовых вычислителей, в связи с чем указанные квантовые алгоритмы могут рассматриваться в качестве одной из возможных угроз безопасности современных криптографических алгоритмов и протоколов.

В данной работе на основе подхода, предложенного в [3], получены оценки ресурсов квантового вычислителя, необходимых для поиска ключа определяемых национальным стандартом ГОСТ Р 34.12-2015 [6] алгоритмов блочного шифрования «Кузнечик» и «Магма» с помощью алгоритма Гровера.

2. КВАНТОВЫЙ КОМПЬЮТЕР

Согласно [5], реализация алгоритма на квантовом вычислителе может быть представлена в виде четырехуровневой иерархической модели:

— **уровень математической абстракции**, который формализует математическое описание реализуемого квантового алгоритма;

* E-mail: DenisenkoDV@bmstu.ru
** E-mail: marshalko_gb@tc26.ru
*** E-mail: marina-nic-msc@yandex.ru
**** E-mail: rudskoy_vi@tc26.ru
† E-mail: shishkin_va@tc26.ru

— **логический уровень**, который подразумевает конкретную реализацию алгоритма с расчетом требуемого количества логических кубитов;

— **уровень коррекции ошибок**, который подразумевает реализацию квантовых корректирующих кодов, предназначенных для исправления ошибок измерений и декогерентизации физических кубитов;

— **физический уровень**, подразумевающий использование конкретной технологии реализации кубитов.

Отметим, что последние два уровня непосредственно определяются физическими параметрами используемой технологии реализации кубитов и в дальнейшем изложении рассматриваться не будут.

Нас интересует оценка минимального количества логических кубитов, достаточного для реализации алгоритма Гровера в задаче поиска ключа алгоритмов блочного шифрования, определяемых ГОСТ Р 34.12-2015, по нескольким известным парам открытого и зашифрованного текстов.

3. АЛГОРИТМ ГРОВЕРА

Пусть имеется проиндексированное множество из $N = 2^n$ элементов. Требуется в этом множестве найти индекс элемента, удовлетворяющего некоторому критерию поиска, причем предполагается, что такой элемент существует и единственен. Иными словами, можно считать, что задана некоторая булева функция $f : V_n \to V_1$, причем $f(x) = 1$ тогда и только тогда, когда элемент множества с индексом x удовлетворяет критерию поиска ($x = \omega$). При этом считается, что указанная функция f реализована как черный ящик или оракул. При решении задачи на классическом вычислителе в общем случае необходимо перебрать все возможные значения индекса, что в итоге дает трудоемкость порядка $O(N)$.

Реализуемый на квантовом вычислителе алгоритм Гровера (см. [2]) имеет трудоемкость $O(\sqrt{N})$. Прежде всего, считается, что существует квантовый оракул, проверяющий выполнение критерия поиска, который представляет собой унитарный оператор U_ω, действующий следующим образом:

$$\begin{cases} U_\omega |x\rangle = -|x\rangle, & \text{если } x = \omega, \text{ т. е. } f(x) = 1, \\ U_\omega |x\rangle = |x\rangle, & \text{если } x \neq \omega, \text{ т. е. } f(x) = 0. \end{cases} \quad (1)$$

Чтобы связать квантовый оракул с классической булевой функцией f, используют эквивалентное задание со вспомогательным кубитом $|y\rangle$:

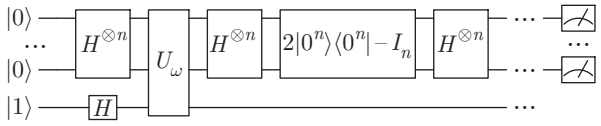

Рис. 1. Алгоритм Гровера

$$\begin{cases} U_\omega |x\rangle |y\rangle = |x\rangle |y \oplus 1\rangle, & \text{если } x = \omega, \\ U_\omega |x\rangle |y\rangle = |x\rangle |y\rangle, & \text{если } x \neq \omega, \end{cases} \quad (2)$$

или кратко $U_\omega |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$. Легко убедиться, что если вспомогательный кубит приведен в состояние

$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = H|1\rangle,$$

то действие оператора U_ω в форме (2) эквивалентно заданию в форме (1):

$$U_\omega(|x\rangle \otimes |-\rangle) = \begin{cases} -|x\rangle \otimes |-\rangle, & \text{если } x = \omega, \\ |x\rangle \otimes |-\rangle, & \text{если } x \neq \omega. \end{cases}$$

Далее алгоритм действует следующим образом. Формируется состояние, являющееся равномерной суперпозицией всех $N = 2^n$ значений аргумента x (для чего потребуется n кубитов):

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle.$$

К этому состоянию (или к состоянию $|s\rangle \otimes |-\rangle$, в зависимости от способа задания оператора оракула) применяется оператор «итерации Гровера», состоящий из последовательного применения оператора оракула U_ω и оператора «рассеивания Гровера» U_s, который задается следующим образом:

$$U_s = 2|s\rangle\langle s| - I,$$

где I — единичный оператор (единичная матрица размера $2^n \times 2^n$).

После определенного количества итераций Гровера, порядка $O(\sqrt{N})$, выполняется измерение всех (или первых n) кубитов, а результат измерения с большой вероятностью будет давать искомое значение ω. Алгоритм проиллюстрирован на рис. 1.

Алгоритм I. Алгоритм Гровера

Вход. Множество $\{a_1, a_2, \ldots, a_N\}$ из $N = 2^n$ элементов, $f : V_n \to V_1$, $f(x) = 1$ тогда и только тогда, когда a_x удовлетворяет некоторому критерию поиска, причем количество таких x, на которых $f(x) = 1$, в общем случае равно M (в нашем случае полагаем $M = 1$), x — номер элемента a_x в двоичной системе счисления, т. е. $x \in V_n$.

Выход. С вероятностью $p > 1/2$ произвольный $a_{x'} : f(x') = 1$.

1. Инициализация $n+1$ кубитов в состояние $|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle$ (дополнительные рабочие кубиты инициализируются в зависимости от вида функции f).

2. Применение вентилей Адамара H:

$$|\psi_1\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

3. Применение «итерации Гровера» $(\pi/4)\sqrt{N/M}$ раз:

а) Изменение знака у амплитуды целевого состояния для всех $i \in \overline{0, N-1}$ (в терминах монографии [7] — применение оракула O)

$$|i\rangle \xrightarrow{O} (-1)^{f(i)}|i\rangle;$$

б) Инверсия относительно среднего (в терминах монографии [7] — применение оператора $2|\psi\rangle\langle\psi| - I$, где I — единичная матрица размером $2^n \times 2^n$):

— применить оператор $H^{\otimes n}$;

— применить оператор $2|0\rangle\langle 0| - I$;

— применить оператор $H^{\otimes n}$.

4. Измерение кубитов, с вероятностью $p > 1/2$ получим произвольный $a_{x'} : f(x') = 1$.

Рассмотрим произвольный алгоритм блочного шифрования с длиной ключа n битов и длиной блока m битов $E : V_n \times V_m \to V_m$. Пусть известно некоторое количество пар открытых и шифрованных текстов, полученных при шифровании на одном и том же ключе $K \in V_n$, $C_i = E(P_i, K)$, $i \in \overline{1,t}$, и решается стандартная задача по восстановлению ключа. Для применения алгоритма Гровера перед нами стоит задача построения оператора U_ω. Прежде всего заметим, что мы требовали, чтобы решение задачи было единственным, т. е. ключ шифрования по заданному набору входов и выходов определялся однозначным образом. С учетом расстояния единственности шифра [8] количество пар текстов должно быть не менее $t = \lceil n/m \rceil$.

В этом случае с большой вероятностью ключ будет единственным, а соответствующая булева функция оракула $f : V_n \to V_1$ определяется следующим образом:

$$f(x) = \bigwedge_{i=1}^{t} z\left(E(P_i, x) \oplus C_i\right),$$

где $z : V_m \to V_1$, причем $z(x) = 1$, если $x = 0^m$, и $z(x) = 0$ в противном случае.

Отметим, что в работе [3] используется завышенная оценка для требуемого числа пар (P_i, C_i), $i \in \overline{1,t}$, что приводит к некорректной результирующей оценке требуемых для реализации алгоритма Гровера ресурсов.

Таким образом, требуемые для реализации алгоритма Гровера ресурсы (количество кубитов и квантовых вентилей) могут быть оценены как произведение соответствующих оценок для ресурсов одной итерации на среднее число итераций алгоритма Гровера (этапом измерения мы в данном случае пренебрегаем). При этом каждая итерация, как уже отмечалось, представляет собой последовательное применение оператора оракула U_ω и оператора «рассеивания Гровера» U_s. Для оператора «рассеивания Гровера» количество квантовых операторов t_{U_ω} будем оценивать в соответствии с работами [3, 4], а оценке количества квантовых операторов t_{U_s}, требуемых для реализации оракула U_ω, посвящены следующие разделы настоящей статьи.

Итоговая оценка требуемого количества квантовых операторов для алгоритма Гровера составляет

$$\frac{\pi}{4}\sqrt{N}(t_{U_\omega} + t_{U_s}). \tag{3}$$

4. РЕАЛИЗАЦИЯ ОДНОЙ ИТЕРАЦИИ АЛГОРИТМОВ ШИФРОВАНИЯ «КУЗНЕЧИК» И «МАГМА»

Для того чтобы реализовать унитарный оператор U_ω, соответствующий произвольному блочному шифру $E : V_n \times V_m \to V_m$, необходимо представить его функцию зашифровывания в виде квантовой схемы. Отображение $E : V_n \times V_m \to V_m$ состоит из m булевых координатных функций $f_i(x_1, \ldots, x_{n+m})$, $i \in \overline{1,m}$. Реализация булевой функции $f(x_1, \ldots, x_n)$ в виде квантовой схемы в общем случае приведена на рис. 2.

Полином Жегалкина булевой функции $f_i(x_1, \ldots, x_{m+n})$ можно реализовать с помощью обобщенных вентилей $\mathrm{CNOT}(C|t)$ (см. [7, 9, 10]), в которых управляемый кубит t контролируется некоторым множеством кубитов C. Для реализации обобщенных вентилей $\mathrm{CNOT}(C|t)$ не требуется использование дополнительных кубитов

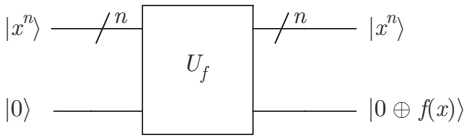

Рис. 2. Общий вид квантовой схемы, реализующей булеву функцию $f(x_1, \ldots, x_n)$

(см. [7], с. 184). Следовательно, для реализации $f_i(x_1, \ldots, x_{n+m})$ требуется $n + m + 1$ кубитов, а количество обобщенных вентилей CNOT$(C|t)$ не превосходит количества слагаемых в полиноме Жегалкина рассматриваемой булевой функции.

Таким образом, если координатные функции шифра $E : V_n \times V_m \to V_m$ легко представимы в виде полинома Жегалкина, то для реализации соответствующей квантовой схемы алгоритма шифрования достаточно $n + m + m$ кубитов.

Алгоритмы блочного шифрования строят таким образом, что получить аналитические выражения для координатных функций $E : V_n \times V_m \to V_m$ в виде полиномов Жегалкина с практической точки зрения невозможно. В связи с этим, для того чтобы представить $E : V_n \times V_m \to V_m$ в виде квантовой схемы, необходимо представить блочный шифр в виде композиции базовых преобразований, а затем каждое базовое преобразование блочного шифра представить в виде квантовой схемы.

Таблица 1. Количество дополнительных кубитов и квантовых вентилей при выполнении базовых преобразований алгоритмов блочного шифрования

Операция	Кол-во доп. кубитов	Кол-во вентилей
$P \oplus \mathrm{Key}$	0	n
$P \boxplus \mathrm{Key} \bmod 2^n$	1	$\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{25}{6}n + 8$
Подстановка	m	N_s
Линейное преобр.	m	$\leq m(m-1)$
Цикл. сдвиг	0	0

Традиционно блочные шифры строятся на основе композиции элементарных преобразований, к которым относятся сложение вектора внутреннего состояния P с ключом K (модульное или побитовый XOR), нелинейное преобразование — подстановка S, линейные преобразования — умножения на матрицу L, сдвиги. В табл. 1 представлены оценки достаточного количества ресурсов (см. [7, 11, 12]) для реализации таких элементарных преобразований.

Отметим, что N_s — количество вентилей для реализации S-боксов, как было показано выше, можно оценить по количеству слагаемых в полиномах Жегалкина координатных функций S-боксов. Существуют другие способы реализации подстановок в виде квантовых схем. Например, в работе [3] удалось найти квантовую схему, задающую байтовый S-бокс алгоритма AES с помощью 9 кубитов. Однако в данной работе будем считать, что подстановки $s \in S(V_m)$ реализуются путем представления координатных функций $f_i(x_1, \ldots, x_m), i \in \overline{1, m}$, в виде полиномов Жегалкина.

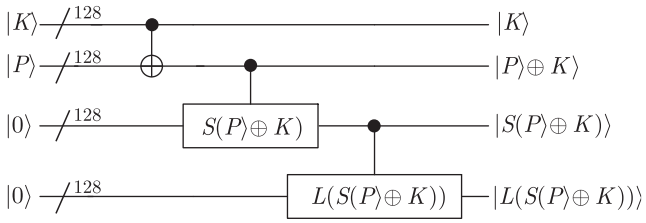

Рис. 3. Квантовая схема одной итерации алгоритма «Кузнечик»

4.1. Реализация одной итерации алгоритма «Кузнечик»

Для реализации одной итерации функции зашифровывания $E : V_{128} \times V_{128} \to V_{128}$ алгоритма ГОСТ Р 34.12-2015 «Кузнечик», которая состоит из сложения внутреннего вектора внутреннего состояния со 128-битным ключом K, параллельного применения шестнадцати 8-битных подстановок и применения линейного преобразования $L : V_{128} \to V_{128}$, в виде квантовой схемы требуется $128 + 128 + 128 + 128 = 512$ кубитов (рис. 3).

При побитовом наложении итерационного ключа в алгоритме «Кузнечик» потребуется 128 вентилей CNOT.

Количество слагаемых в полиномах Жегалкина координатных функций подстановки $\pi : \mathbb{Z}_{2^8} \to \mathbb{Z}_{2^8}$ алгоритма «Кузнечик» равно 133, 130, 103, 125, 108, 123, 132, 123 соответственно порядковым номерам координатных функций, т. е. для реализации $\pi : \mathbb{Z}_{2^8} \to \mathbb{Z}_{2^8}$ требуется 977 обобщенных вентилей CNOT$(C|t)$. Следовательно, для реализации $S : V_{128} \to V_{128}$ требуется $16 \cdot 977 = 15632$ вентиля CNOT$(C|t)$.

Для реализации линейного преобразования алгоритма «Кузнечик» в виде умножения 128-битного вектора на матрицу размером 128×128 бит потребуется 7879 вентилей CNOT. Однако данное линейное преобразование может также быть реализовано с помощью 16 итераций линейного регистра сдви-

га. В этом случае для реализации одной итерации потребуется 415 вентилей, а для всего преобразования — 6640. Оценка количества ресурсов для реализации одного раунда алгоритма «Кузнечик» в виде квантовой схемы приведена в табл. 2.

Таблица 2. Количество кубитов и вентилей при реализации одного раунда алгоритма «Кузнечик»

Кол-во кубитов	Кол-во вентилей
512	$128 + 15632 + 6640 = 22400$

4.2. Реализация одной итерации алгоритма «Магма»

Каждая итерация функции зашифровывания $E : V_{32} \times V_{64} \to V_{64}$ алгоритма ГОСТ Р 34.12-2015 «Магма», являющегося так называемой сетью Фейстеля, преобразует 32-битный подвектор 64-битного вектора внутреннего состояния. Для реализации этого преобразования, которое состоит из наложения ключа по модулю 2^{32}, применения восьми 4-битных подстановок, циклического сдвига на 11 и побитового сложения со вторым подвектором, в виде квантовой схемы потребуется $32 + 32 + 32 + 32 + 32 + 1 = 161$ кубит (см. рис. 4).

Оценим количество обобщенных вентилей $\mathrm{CNOT}(C|t)$.

При наложении раундового ключа в алгоритме «Магма» потребуется (см. [11])

$$\frac{2}{3}32^3 + \frac{3}{2}32^2 - \frac{25}{6}32 + 8 = 23256 \text{ вентилей CNOT.}$$

Количество слагаемых в полиномах Жегалкина координатных функций подстановок $\pi_i : \mathbb{Z}_{2^4} \to \mathbb{Z}_{2^4}$, $i = 0, 1, \ldots, 7$, алгоритма «Магма» равно 7,5,7,4; 4,10,6,8; 7,9,7,11; 6,9,8,10; 2,8,7,9; 7,7,5,5; 7,8,6,8; 5,7,6,11 соответственно порядковым номерам координатных функций, т. е. для реализации S-боксов алгоритма «Магма» потребуется всего 226 обобщенных вентилей $\mathrm{CNOT}(C|t)$.

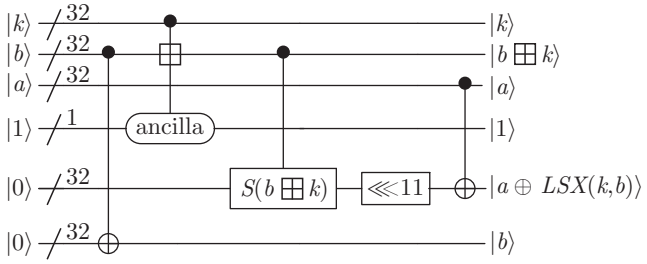

Рис. 4. Квантовая схема одной итерации алгоритма «Магма»

Циклический сдвиг влево на 11 бит (линейного преобразования алгоритма «Магма») может быть реализован с помощью программной перенумерации кубитов (см. табл. 1).

Побитовый XOR двух 32-битных полублоков требует 32 вентиля CNOT. Столько же вентилей понадобится для перезаписи полублока, значение которого было изменено при наложении раундового ключа. Необходимое количество ресурсов для реализации одного раунда алгоритма «Магма» приведено в табл. 3.

Таблица 3. Количество кубитов и вентилей при реализации одного раунда алгоритма «Магма»

Кол-во кубитов	Кол-во вентилей
161	$32 + 23256 + 226 + 32 = 23546$

5. ОЦЕНКА КОЛИЧЕСТВА РЕСУРСОВ ДЛЯ РЕАЛИЗАЦИИ АЛГОРИТМОВ «КУЗНЕЧИК» И «МАГМА»

В предыдущем разделе получены оценки достаточного количества кубитов и обобщенных вентилей $\mathrm{CNOT}(C|t)$ для реализации одной итерации алгоритма «Кузнечик» и одной итерации алгоритма «Магма». В данном разделе рассмотрим два подхода к реализации алгоритмов «Кузнечик» и «Магма». Основная проблема, возникающая при реализации схем рассматриваемого типа, – рост числа кубитов вследствие необходимости приведения используемых преобразований к унитарному виду. В связи с этим возникает два подхода к реализации.

Первый подход к реализации алгоритмов — без повторного использования кубитов. Общая схе-

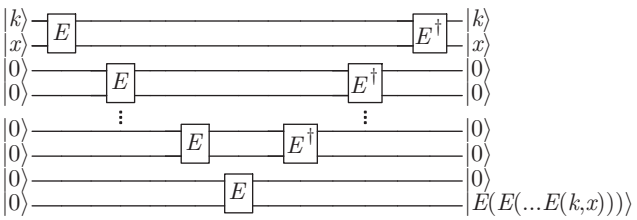

Рис. 5. Композиция раундовых преобразований без экономии кубитов

ма вычисления композиции нескольких итераций условного алгоритма $E : V_n \times V_m \to V_m$ приведена на рис. 5.

При такой реализации количество вентилей и кубитов можно оценить путем перемножения количества итераций на соответствующее количество ресурсов, необходимое на один раунд блочного шифра. Отметим, что количество вентилей необходимо еще умножить на 2, чтобы учесть применение обратных преобразований $E^\dagger$, необходимых для того, чтобы после измерения получить не только какой-то зашифрованный текст $E(E(\dots E(k,x)))$, но и пару (k,x), на которой этот зашифрованный текст получен, а также для того, чтобы избавиться от «запутанных» (см. [7]) состояний системы.

Второй подход — это использованная в работе [3] реализация с повторным использованием кубитов. Идея данного подхода заключается в том, что, выполнив первоначально несколько преобразований, мы можем вернуть в начальное состояние часть задействованных кубитов, применив обратные преобразования, с тем чтобы использовать данные кубиты в последующих вычислениях (см. рис. 6). При этом должны быть выполнены следующие условия:

— обратные преобразования должны применяться в обратном порядке по отношению к прямым преобразованиям с тем, чтобы иметь возможность провести преобразования над кубитами, находящимися в соответствующем состоянии;

— обратное преобразование должно применяться по крайней мере после одного прямого преобразования с тем, чтобы сохранить состояние, полученное применением соответствующего прямого преобразования.

5.1. Алгоритм «Кузнечик»

В алгоритме ГОСТ Р 34.12-2015 «Кузнечик» применяется 9 полных итераций, а в 10-й итерации применяется только наложение ключа. Квантовая схема, реализующая 10 раундов алгоритма «Кузнечик» (см. описание алгоритма [6]) с повторным использованием кубитов, продемонстрирована на рис. 6.

Для реализации квантовой схемы, приведенной на рис. 6, требуется $8 \cdot 128 = 1024$ кубита, причем алгоритм развертывания раундовых ключей уже учтен. Для простоты будем считать, что для вычисления S и $S^\dagger$, а также для вычисления L и $L^\dagger$ требуется одинаковое количество вентилей. На рис. 6 побитовый XOR 128-битных блоков выполняется 16 раз, 16 раз применяется S и 14 раз вычисляется L, после обращения первого раунда с помощью вентилей Паули происходит обнуление 128 кубитов, в которые записан блок открытого текста P (мы восстанавливаем ключ по известным парам открытого и шифрованного текстов). Следовательно, для реализации схемы без учета алгоритма развертывания ключа требуется не более $16 \cdot 128 + 16 \cdot 15632 + 14 \cdot 6640 + 128 = 345248$ обобщенных вентилей.

Алгоритм развертывания ключа представляет собой 32-раундовую сеть Фейстеля, функция усложнения которой совпадает с итерационным преобразованием алгоритма «Кузнечик», где вместо добавления ключа используется добавление фиксированных констант. Квантовая схема одной итерации представлена на рис. 7, для ее реализации требуется $128 + 128 = 256$ вспомогательных кубитов и $2 \cdot (128 + 15632 + 6640) + 128 + (128 \cdot 3) = 45312$ обобщенных вентилей. Для 32 итераций требуется $32 \cdot 45312 = 1449984$ вентилей.

На рис. 6 в блоках Ki $(i = 1, 2, 3, 4)$ формируются раундовые ключи алгоритма шифрования «Кузнечик» (см. описание алгоритма [6]). Каждый такой блок включает 8 итераций квантовой схемы, изображенной на рис. 7.

Квантовая схема одной итерации без повторного использования кубитов представлена на рис. 8, для ее реализации требуется $128 \cdot 3 = 384$ вспомогательных кубита и $128 + 128 + 15632 + 6640 + 128 = 22656$ обобщенных вентилей. Для 32 итераций требуется $256 + 32 \cdot 384 = 12544$ кубита и $32 \cdot 22656 = 724992$ вентиля.

Оценки количества кубитов и вентилей, необходимых для реализации в виде квантовой схемы процедуры зашифровывания одного блока открытого текста блочным шифром «Кузнечик», приведены в табл. 4 и 5.

Таблица 4. Реализация алгоритма «Кузнечик» с повторным использованием кубитов

Кол-во кубитов	Кол-во CNOT$(C\|t)$
1024	1795232

Таблица 5. Реализация алгоритма «Кузнечик» без повторного использования кубитов

Кол-во кубитов	Кол-во CNOT$(C\|t)$
$128 + 256 \cdot 9 + 12544 =$ $= 14976$	$9 \cdot 22400 + 128 + 724992 =$ $= 926720$

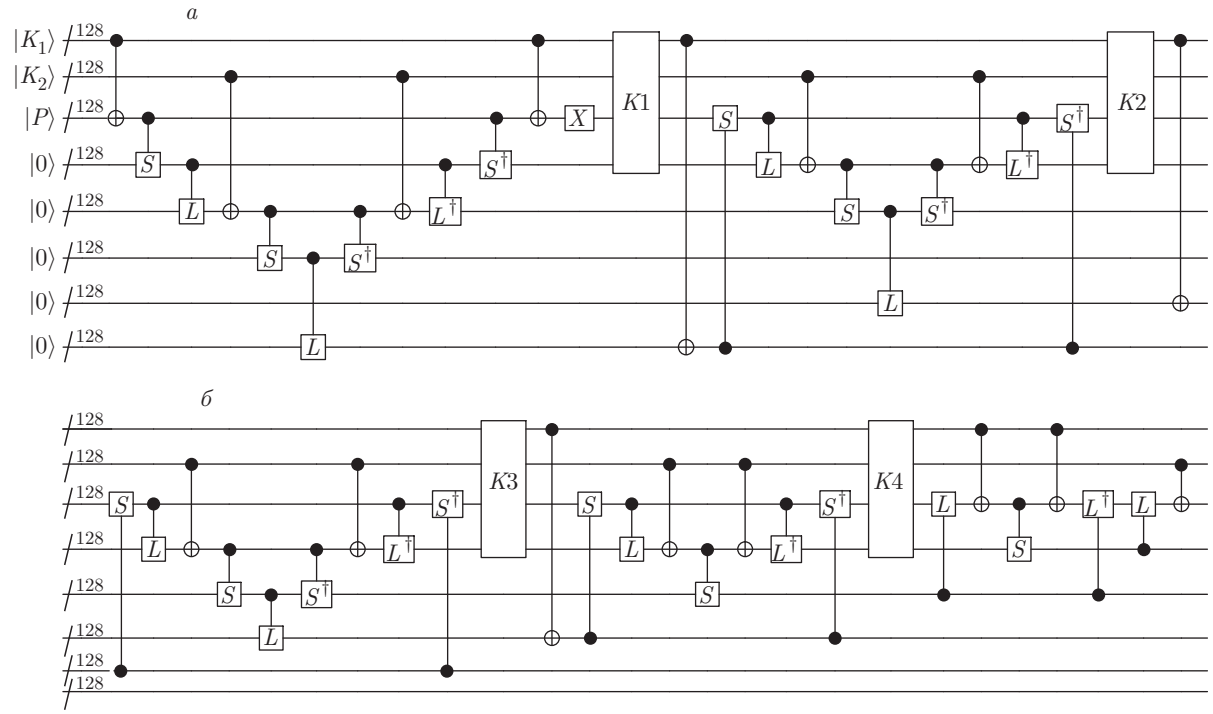

Рис. 6. Квантовая схема 10 раундов алгоритма «Кузнечик» с повторным использованием кубитов ($а$ — первые четыре итерации алгоритма, $б$ — оставшиеся пять полных итераций и одна неполная)

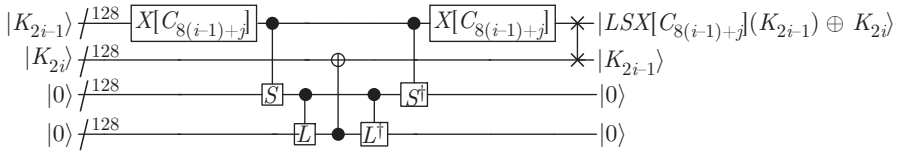

Рис. 7. Квантовая схема для одной итерации алгоритма развертывания ключа алгоритма шифрования «Кузнечик» с повторным использованием кубитов, $i = 1, 2, 3, 4$, $j = 1, 2, \ldots, 8$

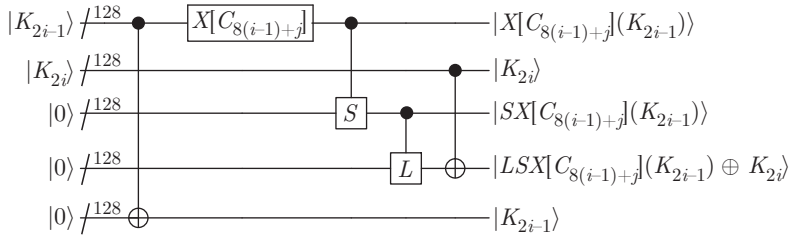

Рис. 8. Квантовая схема для одной итерации алгоритма развертывания ключа алгоритма шифрования «Кузнечик» без повторного использования кубитов, $i = 1, 2, 3, 4$, $j = 1, 2, \ldots, 8$

5.2. Алгоритм «Магма»

Квантовая схема, реализующая процедуру зашифровывания одного блока открытого текста с помощью алгоритма «Магма» с повторным использованием кубитов, состоит из 32 последовательных применений квантовой схемы на рис. 9.

В силу простоты алгоритма формирования итерационных ключей алгоритма «Магма», для записи всех раундовых ключей требуется 256 кубитов. Для записи блока открытого текста требуется 64 кубита, а для хранения промежуточных значений — еще 33 дополнительных кубита, т. е. общее количество кубитов равно $256 + 64 + 32 + 1 = 353$. Для оценки

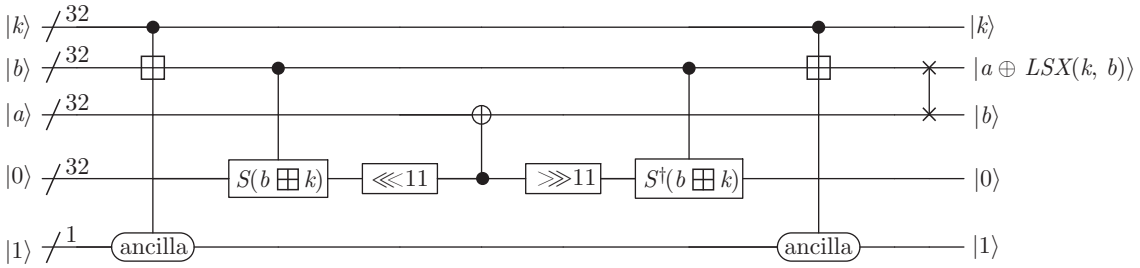

Рис. 9. Квантовая схема одной итерации алгоритма «Магма» с повторным использованием кубитов

количества обобщенных вентилей воспользуемся результатами, полученными в разд. 4.2. Для наложения раундового ключа необходимо 23256 вентилей, для реализации S-боксов потребуется 226 обобщенных вентилей, побитовый XOR двух 32-битных полублоков требует 32 вентилей. С учетом процедуры обращения кубитов, показанной на рис. 9, количество обобщенных вентилей равно $23256 + 226 + 32 + 226 + 23256 + 32 \cdot 3 = 47092$.

Оценки количества кубитов и вентилей, необходимых для реализации блочного шифра «Магма» в виде квантовой схемы (применительно к одному блоку открытого текста), приведены в табл. 6 и 7.

Таблица 6. Реализация алгоритма «Магма» с повторным использованием кубитов

Кол-во кубитов	Кол-во CNOT($C\|t$)
353	47092·32=1506944

Таблица 7. Реализация алгоритма «Магма» без повторного использования кубитов

Кол-во кубитов	Кол-во CNOT($C\|t$)
256+64·32+1+64 = 2369	$23546 \cdot 32 = 753472$

5.3. Общая оценка количества квантовых ресурсов для применения алгоритма Гровера в задаче поиска ключей алгоритмов блочного шифрования ГОСТ Р 34.12-2015

Для поиска ключа алгоритма «Кузнечик» требуется $\lceil \frac{256}{128} \rceil = 2$ пары блоков открытого и шифрованного текстов, поэтому для применения алгоритма Гровера к алгоритму «Кузнечик» требуется

— $1024 + (1024 - 256) + 1 = 1793$ кубита;

— не менее $\frac{\pi}{4} \cdot 2^{128}(t_{U_\omega} + t_{U_s})$ вентилей.

Для поиска ключа алгоритма «Магма» требуется $\lceil \frac{256}{64} \rceil = 4$ пары блоков открытого и шифрованного текстов, поэтому для применения алгоритма Гровера к алгоритму «Магма» требуется

— $256 + 64 \cdot 4 + 32 + 1 + 1 = 546$ кубитов;

— не менее $\frac{\pi}{4} \cdot 2^{128}(t_{U_\omega} + t_{U_s})$ вентилей.

Значение t_{U_ω} оцениваем количеством операторов CNOT($C|t$) по табл. 4 и 6. При этом количество вентилей в данных таблицах необходимо умножить на 2 для обращения схем.

Для того чтобы понять, как точно посчитать количество вентилей t_{U_s}, воспользуемся результатами работы [4]. Для обоих алгоритмов ГОСТ Р 34.12-2015 в процедуре «рассеивания Гровера» к ключу последовательно применяются 256 вентилей Адамара H, 256 вентилей Паули X, один вентиль Адамара H, один обобщенный CCNOT, один вентиль Адамара H, 256 вентилей Паули X и еще 256 вентилей Адамара H, т. е. всего 1027 квантовых вентилей. Кроме того, при изменении флагового кубита происходит проверка на совпадение с известными блоками шифрованного текста, для которой дополнительно требуется не более 256 вентилей Паули X, один обобщенный CCNOT и еще не более 256 вентилей Паули X для возврата соответствующих кубитов в исходное состояние перед проверкой, т. е.

$$1027 + 1 \leq t_{U_s} \leq 1027 + 512 + 1.$$

Перед процедурой измерения переводить флаговый кубит в первоначальное состояние $|1\rangle = H|-\rangle$ не обязательно, поэтому осталось учесть еще один вентиль Адамара, применяемый при формировании состояния $|-\rangle = H|1\rangle$ на флаговом кубите.

6. ЗАКЛЮЧЕНИЕ

Полученные результаты позволяют оценить число логических кубитов, необходимых для реализации рассмотренных алгоритмов. Для реализации

логических кубитов вследствие процессов декогерентизации потребуется использование кодов коррекции квантовых ошибок, что, в свою очередь, подразумевает примерно на порядок большее число физических кубитов. Например, код Шора позволяет кодировать один логический кубит девятью физическими кубитами. Коды Стина (CSS-коды) позволяют кодировать один логический кубит пятью физическими кубитами — это минимальное количество физических кубитов для исправления одной ошибки (см. [7]).

Работы [13, 14] посвящены теоретическим оценкам минимального времени выполнения одного квантового вентиля на одном кубите. В случае реализации кубита с помощью иона Ca^+ в ловушке это время составляет порядка $6.62 \cdot 10^{-16}$ с, что значительно меньше, чем время выполнения одного вентиля на практике: 10^{-9} с.

Грубые оценки времени потери когерентности (декогерентизации), времени выполнения одной операции над кубитом и максимального числа операций, последовательно выполняемых над кубитом (максимальной длины квантовой схемы), приведены в работе [7], табл. 7.1, с. 278.

В марте 2018 г. компания Google представила 72-кубитный квантовый процессор Bristlecone на базе сверхпроводников (см. [15]). Согласно [16, 17], время когерентности систем на базе сверхпроводников составляет около 100 мкс, а минимальное время применения однокубитового вентиля, согласно [18] (см. табл. 2 на с. 20), составляет 5 нс. Таким образом, к кубитам на базе сверхпроводников можно успеть применить не более $\frac{100 \cdot 10^{-6}}{5 \cdot 10^{-9}} = 20000$ однокубитовых вентилей, после чего система перейдет в некоторое основное состояние, и состояние суперпозиции кубитов будет разрушено. Тем не менее, согласно [18] количество вентилей, которые на современном этапе развития квантовых технологий могут быть применены к кубитам на базе сверхпроводников, составляет порядка 10^3.

ЛИТЕРАТУРА

1. P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).

2. L. K. Grover, in *Proc. STOC1996*, pp. 212–219, ACM (1996).

3. M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, arXiv:1512.04965v1.

4. Д. В. Денисенко, М. В. Никитенкова, ЖЭТФ **155**, 32 (2019).

5. M. Amy, O. Di Matteo, V. Gheorghiu, M. Mosca, A. Parent, and J. Schanck, ePrint Report 2016/992.

6. ГОСТ Р 34.12-2015, *Информационные технологии. Защита информации. Блочные шифры.*

7. M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge Univ. Press, Cambridge (2010).

8. К. Шеннон, в кн. *Работы по теории связи и кибернетике*, Изд-во иностр. лит., Москва (1963).

9. A. Younes and J. Miller, arXiv:quant-ph/0304099v1.

10. S. Ding and Z. Jin, ICCCAS 2007, Beijing, China, DOI:10.1109/ICCCAS.2007.4348267 (2007).

11. P. Kaye, arXiv:quant-ph/0408173.

12. D. Beckman, A. N. Chari, and J. Preskill, arXiv:org/abs/quant-ph/9602016.

13. S. Ashhab, P. C. Groot, and F. Nori, arXiv:1202.5872v2 [quant-ph].

14. L. B. Levitin, T. Toffoli, and Z. Walton, arXiv:quant-ph/0211167v3.

15. research.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html?m=1.

16. J. M. Martinis and A. Megrant, arXiv:1410.5793v1.

17. J. Portes, *Decoherence, Superconducting Qubits, and the Possibility of Quantum Computing*, Columbia Univ. (2015).

18. G. Wendin, arXiv:1610.02208v2.