ЖЭТФ, 2019, том 155, вып. 1, стр. 32–53

ПРИМЕНЕНИЕ КВАНТОВОГО АЛГОРИТМА ГРОВЕРА В ЗАДАЧЕ ПОИСКА КЛЮЧА БЛОЧНОГО ШИФРА SDES

Д. В. Денисенко**, *М. В. Никитенкова

Московский государственный технический университет им. Н. Э. Баумана
105005, Москва, Россия

Поступила в редакцию 6 июля 2018 г.,
после переработки 6 июля 2018 г.
Принята к публикации 10 июля 2018 г.

Рассмотрена задача поиска ключа Simplified-DES — модели блочного шифра DES, с помощью квантового алгоритма Гровера. Рассмотрены примеры применения алгоритма Гровера. Построена минимальная по количеству кубитов квантовая схема, реализующая поиск ключа Simplified-DES по одной паре открытого и шифрованного текстов, для которой требуется 19 кубитов. Проведена симуляция работы построенной квантовой схемы с использованием квантового симулятора Quipper.

DOI: 10.1134/S0044451019010036

1. ВВЕДЕНИЕ

В настоящее время большое количество исследований направлено на создание квантовых симуляторов и квантовых процессоров: в 2017 г., на конференции ICQT 2017, группа физиков под руководством М. Лукина, сооснователя Российского квантового центра и профессора Гарвардского университета, сообщила о создании программируемого 51-кубитного квантового симулятора [1]. Примерно в это же время группа ученых из университета Мэриленда разработала 53-кубитный симулятор, основанный на ионах в оптических ловушках [2]. В компании IBM успешно испытали прототип квантового процессора из 50 кубитов [3], а в декабре 2017 г. опубликована статья [4], согласно которой представлен проект масштабируемого кремниевого квантового процессора, представляющий собой массив из $24 \times 20 = 480$ кубитов. В январе 2018 г. на выставке CES-2018 компания Intel сообщила о создании сверхпроводящего квантового чипа «Tangle Lake», состоящего из 49 кубитов. Intel ведет разработки квантовых компьютеров по двум направлениям: создание устройств на сверхпроводниках и кремниевых чипов со «спиновыми кубитами». В марте 2018 года компания Google объявила о создании 72-кубитного квантового процессора Bristlecone. В компании надеются, что Bristlecone позволит продемонстрировать «квантовое превосходство» [5, 6].

Квантовые вычисления оказывают непосредственное влияние на защиту информации с помощью современных криптографических алгоритмов и протоколов. Например, возможность применения квантового алгоритма Шора [7] делает криптографическую систему RSA небезопасной, квантовый алгоритм Саймона [8] делает небезопасным использование блочных шифров в режимах CBC-MAC, PMAC, GMAC, GCM и OCB [9]. Квантовый алгоритм Гровера (см. [10–12]) в модели квантовых вычислений является аналогом метода полного перебора ключей алгоритмов шифрования. В работе [13] рассмотрена задача поиска ключа Simplified-DES (SDES) — уменьшенной модели блочного шифра DES — с помощью квантового алгоритма Гровера, представлено описание соответствующей квантовой схемы, реализующей поиск ключа SDES по одной известной паре блоков открытого и шифрованного текстов, использующей 61 кубит. Авторы работы [13] пользовались квантовым симулятором libquantum (см. [14]), однако исходные коды программ не были опубликованы.

В данной работе представлена квантовая схема, реализующая поиск ключа SDES по одной паре открытого и шифрованного текстов на 19 кубитах, верифицированная с помощью квантового симулятора Quipper (см. [15–17]), который, в отличие от libquantum, позволяет автоматически печатать квантовые схемы в PDF-файлы. Показано, что

* E-mail: DenisenkoDV@bmstu.ru

19 кубитов — минимальное количество логических кубитов, достаточное для реализации поиска ключа SDES по одной паре открытого и шифрованного текстов.

Главные цели настоящей работы — демонстрация применения квантового алгоритма Гровера в задаче поиска ключа учебного алгоритма блочного шифрования SDES и оценка минимального количества логических кубитов, необходимого для реализации такого поиска.

В Приложении 1 приведен контрольный пример SDES. Программная реализация применения алгоритма Гровера в задачах поиска одного и двух целевых значений в Wolfram Mathematica представлена в Приложениях 2 и 3. Программная реализация поиска ключа SDES квантовым алгоритмом Гровера в Quipper представлена в Приложении 4.

2. ОПИСАНИЕ АЛГОРИТМА SDES

Блочный шифр SDES — это двухраундовая сеть Фейстеля $E_{SDES} : V_{10} \times V_8 \to V_8$, у которой ключ $K \in V_{10}$, а блоки открытого и шифрованного текста — восьмибитные двоичные векторы (см. рис. 1).

Процедура зашифрования алгоритмом SDES представляет собой композицию отображений:

$$\mathrm{IP}^{-1} \circ f_{k_2} \circ \mathrm{SW} \circ f_{k_1} \circ \mathrm{IP}.$$

Из основного ключа $K \in V_{10}$ формируются два раундовых ключа k_1, $k_2 \in V_8$. Сначала к ключу K применяется перестановка битов P10:

P10 — перестановка	3 5 2 7 4 10 1 9 8 6

Затем к каждой половине P10(K) применяется циклический сдвиг влево на 1 бит (LS-1), после чего из двух половинок снова формируется десятибитный ключ. Далее применяется P8 — выборка восьми битов с соответствующими номерами — и получается раундовый ключ k_1:

P8 — выборка битов с номерами	6 3 7 4 8 5 10 9

Для формирования k_2 после применения LS-1 применяется циклический сдвиг влево на два бита (LS-2), затем применяется P8.

Рассмотрим первый раунд зашифрования блока открытого текста $P \in V_8$ алгоритмом SDES.

К открытому тексту P применяется перестановка битов IP:

IP	2 6 3 1 4 8 5 7

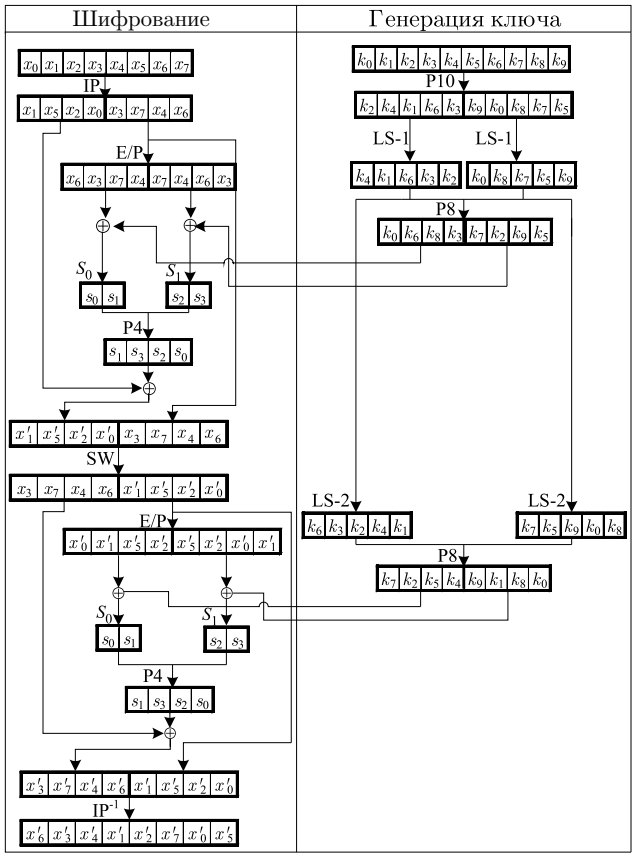

Рис. 1. Схема алгоритма SDES

Преобразование f_k определяется следующим образом. Пусть $P = L||R$, $L, R \in V_4$, тогда $f_k(L, R) = (L \oplus F(R, k), R)$.

Отображение $F(R, k)$: $V_4 \times V_8 \to V_4$ состоит из следующих последовательно применяющихся преобразований.

1. Процедура расширения E/P: $V_4 \to V_8$, выборка битов с соответствующими номерами:

E/P	4 1 2 3 2 3 4 1

2. XOR результата E/P с аргументом k.

3. Применение S-боксов. На каждый S-бокс подается 4-битный вектор, первый и четвертый биты которого образуют номер строки, а второй и третий образуют номера столбцов таблиц замен:

$$S_0 = \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix}, \quad S_1 = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix}.$$

Нумерация строк и столбцов начинается с нуля. Пример: $S_0(1010) = 10$, так как в строке с номером 10 (третья строка) и столбце с номером 01 (второй

столбец) в таблице замен S_0 стоит число 2, которое в двоичной системе счисления записывается как 10.

Координатные функции S_0 имеют вид

$$y_0(x_0,x_1,x_2,x_3) = x_3 \oplus x_1\overline{x}_0 \oplus x_0x_2 \oplus x_0x_1x_2x_3,$$

$$y_1(x_0,x_1,x_2,x_3) = \overline{x}_0\overline{x}_2 \oplus x_0\overline{x}_3 \oplus x_0\overline{x}_1 \oplus x_0x_1\overline{x}_2x_3.$$

Координатные функции S_1 имеют вид

$$y_0(x_0,x_1,x_2,x_3) = x_1 \oplus \overline{x}_2x_3 \oplus x_0\overline{x}_3 \oplus x_0\overline{x}_1x_2\overline{x}_3,$$

$$y_1(x_0,x_1,x_2,x_3) = x_2\overline{x}_3 \oplus x_0\overline{x}_3 \oplus \overline{x}_0x_1x_3 \oplus x_0x_2x_3.$$

4. К 4-битному выходу из S-боксов применяется перестановка битов P4:

P4	2 4 3 1

5. Побитовый XOR с левой половиной $\mathrm{IP}(P)$.

6. Перестановка полубайтов SW: $V_8 \to V_8$, $\mathrm{SW}(L,R) = (R,L)$.

Второй раунд выполняется аналогично первому, только с ключом k_2. После выполнения второго раунда применяется IP^{-1}:

IP^{-1}	4 1 3 5 7 2 8 6

В результате получим блок зашифрованного текста $C = E_{SDES}(K,P)$, $C \in V_8$.

3. АЛГОРИТМ ГРОВЕРА

Пусть имеется пронумерованное множество из $N = 2^n$ элементов и необходимо найти хотя бы один элемент из этого множества, удовлетворяющий некоторому критерию поиска, при этом множество элементов, удовлетворяющих выбранному критерию поиска, не является пустым и состоит из M элементов, $M \leqslant N/2$ (см. [12], с. 318).

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 номером x удоветворяет критерию поиска. При этом считается, что указанная функция f может быть эффективно реализована в виде квантовой схемы.

При решении задачи поиска на классическом вычислителе в общем случае необходимо перебрать все элементы рассматриваемого множества, что в итоге дает трудоемкость порядка $O(N/M)$, в то время как квантовый алгоритм Гровера (см. [10–12]) при решении указанной задачи на квантовом вычислителе имеет трудоемкость $O(\sqrt{N/M})$.

Для произвольного блочного шифра задача поиска ключа с помощью квантового алгоритма Гровера формулируется следующим образом. Рассмотрим блочный шифр с длиной ключа n битов и длиной блока m битов E: $V_n \times V_m \to V_m$. Известно некоторое количество пар открытых и шифрованных текстов, полученных на одном и том же неизвестном ключе $K \in V_n$, $C_i = E(K,P_i)$, $i \in \overline{1,t}$, и решается стандартная задача по восстановлению ключа. Для однозначного восстановления ключа, исходя из расстояния единственности шифра [18], 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 f: $V_n \to V_1$ определяется следующим образом:

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

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 [13] поиск ключа осуществлен только по одной паре открытого и шифрованного текстов (P,C), рассматриваются два случая: в первом — у пары (P,C) нет эквивалентных ключей (т. е. существует ровно один ключ, на котором блок открытого текста P переходит в блок шифрованного текста C), во втором — у пары (P,C) два эквивалентных ключа. В данной работе, следуя работе [13], поиск ключа SDES организован по одной паре открытого и шифрованного текстов, рассмотрены те же примеры, что и в работе [13].

В Приложениях 2 и 3 представлены программные реализации, демонстрирующие преобразование амплитуд квантовых состояний в процессе выполнения алгоритма Гровера в задачах поиска одного и двух целевых значений соответственно, т. е. рассмотрены случаи, когда $M = 1$ и $M = 2$.

Алгоритм 1. 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, где 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 функции f.

2. Применение гейтов Адамара H, получим

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

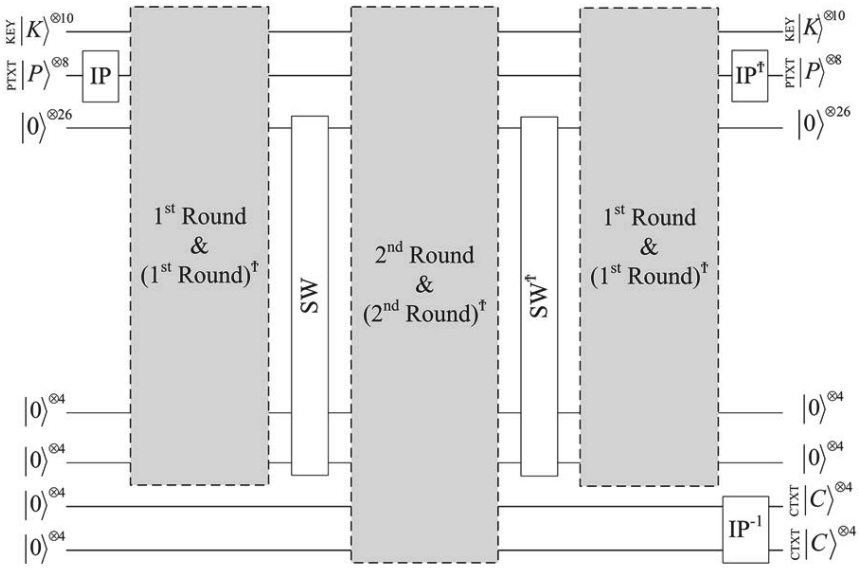

Рис. 2. Реализация SDES в виде квантовой схемы

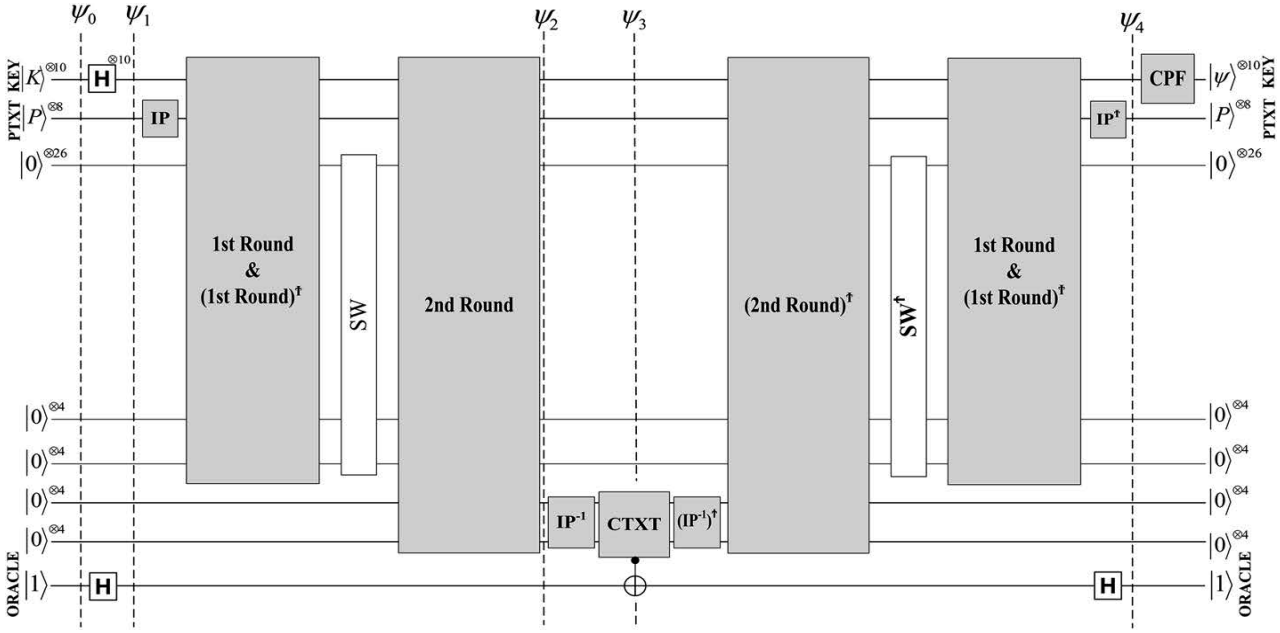

Рис. 3. Квантовая схема одной итерации Гровера

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 книге [12] — применение оракула O)

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

б) инверсия относительно среднего (увеличение вероятности получить одно из целевых значений, в [12] — 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 $H^{\otimes n}$.

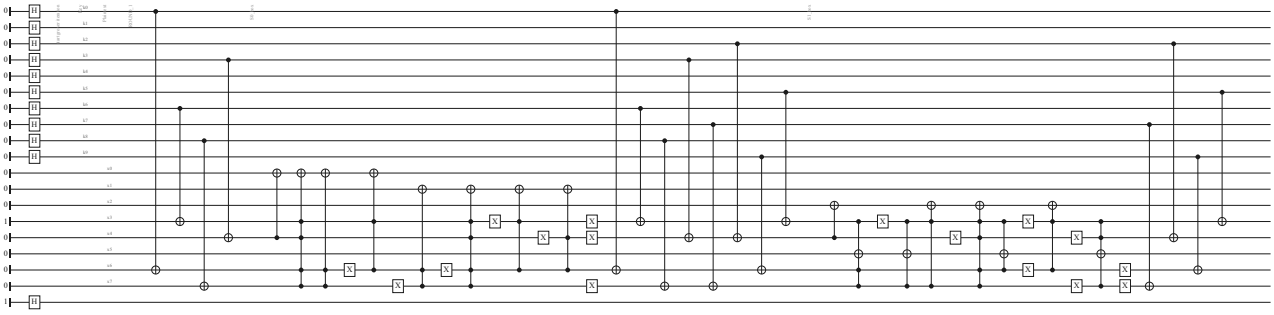

Рис. 4. Первая часть схемы: первый раунд SDES

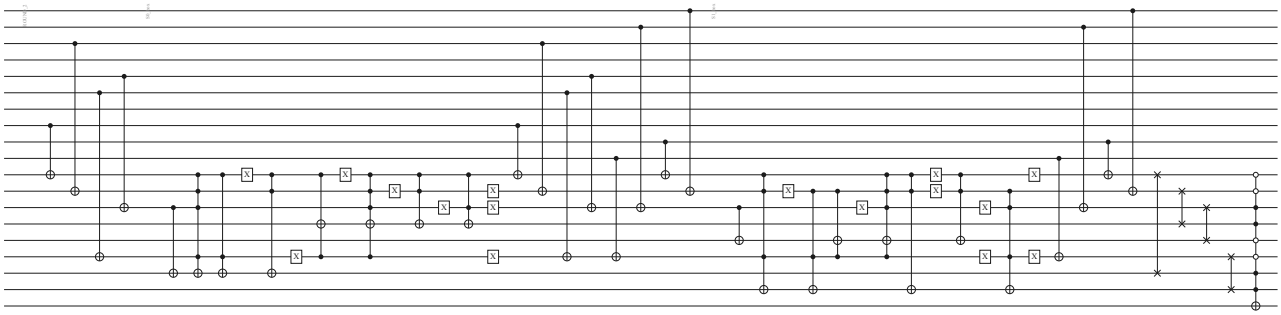

Рис. 5. Вторая часть схемы: второй раунд SDES, инвертирование нижнего кубита

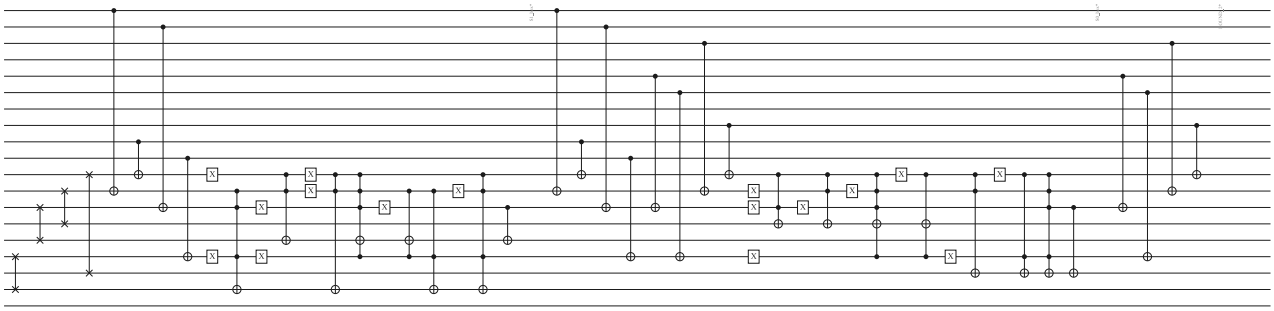

Рис. 6. Третья часть схемы: обратное преобразование второго раунда SDES

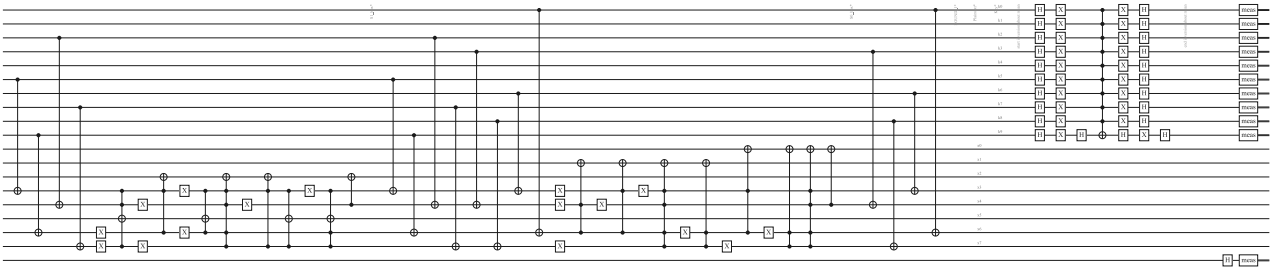

Рис. 7. Четвертая часть схемы: обратное преобразование первого раунда SDES, рассеивание Гровера, измерение кубитов

Таблица 1. Распределение вероятностей ключей после 25 итераций Гровера при поиске ровно одного целевого значения

Ключ SDES	Вероятность получить в результате измерения кубитов соответствующий ключ
00000 00000	$5.26642 \cdot 10^{-7}$
00000 00001	$5.26642 \cdot 10^{-7}$
⋮	⋮
11000 10011	0.99946124
⋮	⋮
11111 11110	$5.26642 \cdot 10^{-7}$
11111 11111	$5.26642 \cdot 10^{-7}$

Таблица 2. Распределение вероятностей ключей после 18 итераций Гровера при поиске одного из двух целевых значений

Ключ SDES	Вероятность получить в результате измерения кубитов соответствующий ключ
00000 00000	$4.118199 \cdot 10^{-6}$
00000 00001	$4.118199 \cdot 10^{-6}$
⋮	⋮
00100 10111	0.4978955
⋮	⋮
00110 11111	0.4978955
⋮	⋮
11111 11110	$4.118199 \cdot 10^{-6}$
11111 11111	$4.118199 \cdot 10^{-6}$

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

4. КВАНТОВАЯ СХЕМА SDES И АЛГОРИТМА ГРОВЕРА

В работе [13] представлена реализация SDES в виде квантовой схемы на 60 кубитах и квантовая схема алгоритма Гровера на 61 кубите (рис. 2, 3):

1) 10 кубитов для записи ключа;

2) 8 кубитов для записи блока открытого текста;

3) 8 кубитов для записи блока зашифрованного текста;

4) 34 кубита — рабочее пространство для записи результатов промежуточных вычислений.

На рис. 4–7 представлена более эффективная квантовая схема итерации Гровера (по сравнению с квантовой схемой, описанной в [13]), реализующая поиск ключа SDES по одной паре открытого и шифрованного текста, финальное измерение кубитов. Программная реализация поиска ключа SDES в квантовом симуляторе Quipper приведена в Приложении 4.

Рассмотрим два случая.

В первом случае выберем в качестве блока открытого текста $P = 00010000$, в качестве блока шифрованного текста $C = 00110011$. Для такой пары (P, C) существует единственный ключ $K = 1100010011$, на котором $E_{SDES}(K, P) = C$.

Во втором случае выберем $P = 10100101$ и $C = 00110110$. Для такой пары (P, C) существуют два ключа

$$K_1 = 0010010111 \text{ и } K_2 = 0011011111,$$

на которых $E_{SDES}(K_i, P) = C$, $i \in \{1, 2\}$.

В соответствии с теоретической оценкой количества итераций Гровера, в первом случае оптимальное количество итераций Гровера составляет

$$R = \frac{\pi}{4}\sqrt{\frac{N}{M}} = \frac{\pi}{4}\sqrt{\frac{1024}{1}} \approx [25.1327] = 25,$$

а во втором случае —

$$R = \frac{\pi}{4}\sqrt{\frac{N}{M}} = \frac{\pi}{4}\sqrt{\frac{1024}{2}} \approx [17.7715] = 18.$$

В табл. 1 представлено распределение вероятностей ключей после 25 итераций Гровера при $P = 00010000$ и $C = 00110011$ (первый случай), полученное с помощью квантового симулятора Quipper.

В табл. 2 представлено распределение вероятностей ключей после 18 итераций Гровера при $P = 10100101$ и $C = 00110110$ (второй случай), также полученное с помощью квантового симулятора Quipper. Заметим, что ключи $0010010111_{10} = 151$ и $0011011111_{10} = 223$, т. е. соответствуют TargetValues $\in \{151, 223\}$ в задаче, рассмотренной в Приложении 3. Сумма вероятностей

$$P(K = 151_2) + P(K = 223_2) = 2 \cdot 0.4978955 \equiv 0.995791$$

Таблица 3

Количество итераций Гровера	Характеристики квантовых схем, время их симуляции в Quipper на процессоре Intel Core i7-4470K 3.50 ГГц
Одна итерация Гровера	Quantum exhaustive key search (1 key): 34: "H, arity 1" 17: "Init0" 2: "Init1" 11: "Meas" 84: "X, arity 1" 72: "not, arity 1 controls 1" 36: "not, arity 1 controls 2" 8: "not, arity 1 controls 3" 12: "not, arity 1 controls 4" 1: "not, arity 1 controls 4+4" 1: "not, arity 1 controls 9" 8: "swap, arity 2" Total gates: 286 Inputs: 0 Outputs: 19 Qubits in circuit: 19 Время работы: 290.288319 с
25 итераций Гровера (поиск одного ключа)	Quantum exhaustive key search (1 key): 562: "H, arity 1" 17: "Init0" 2: "Init1" 11: "Meas" 2100: "X, arity 1" 1800: "not, arity 1 controls 1" 900: "not, arity 1 controls 2" 200: "not, arity 1 controls 3" 300: "not, arity 1 controls 4" 25: "not, arity 1 controls 4+4" 25: "not, arity 1 controls 9" 200: "swap, arity 2" Total gates: 6142 Inputs: 0 Outputs: 19 Qubits in circuit: 19 Время работы: 7665.869522 с

Продолжение таблицы 3

Количество итераций Гровера	Характеристики квантовых схем, время их симуляции в Quipper на процессоре Intel Core i7-4470K 3.50 ГГц
18 итераций Гровера (поиск одного из двух ключей)	Quantum exhaustive key search (2 keys): 408: "H, arity 1" 14: "Init0" 5: "Init1" 11: "Meas" 1512: "X, arity 1" 1296: "not, arity 1 controls 1" 648: "not, arity 1 controls 2" 144: "not, arity 1 controls 3" 216: "not, arity 1 controls 4" 18: "not, arity 1 controls 4+4" 18: "not, arity 1 controls 9" 144: "swap, arity 2" Total gates: 4434 Inputs: 0 Outputs: 19 Qubits in circuit: 19 Время работы = 6593.336957 с

Таблица 4. Распределение квантовых вентилей по операциям SDES из работы [13]

E/P	8 CNOT
XOR полубайтов	4 CNOT
P4	4 CNOT
XOR с раундовым ключом	8 CNOT
SWAP	12 CNOT
Для каждого S-бокса	2×32 вентиля X 2×48 Тоффоли 2×32 CNOT

Таблица 5. Сравнение количества вентилей в одной итерации Гровера

Операция	В работе [13]	На рис. 4–7
X	404	84
H	34	34
CNOT	936	72+24
Тоффоли	576	36
Обобщенный CNOT	2	22

совпадает с вероятностью успеха алгоритма Гровера после 18 итераций в табл. 7, рассчитанной в Wolfram Mathematica (см. Приложение 3).

Сводные данные по характеристикам построенных квантовых схем представлены в табл. 3.

В предложенной квантовой схеме используются так называемые обобщенные вентили (гейты) CNOT(n) — вентили с одним контролируемым кубитом и n контролирующими кубитами, для реализации которых не требуются дополнительные рабочие кубиты (см. [12], с. 236).

Нас интересует минимальная оценка количества логических кубитов для реализации алгоритма Гровера в задаче поиска ключа SDES. Отображение E_{SDES}: $V_{10} \times V_8 \to V_8$ состоит из 8 булевых координатных функций $f_i(x_1, \ldots, x_{10+8})$, $i \in \overline{1,8}$, зависящих от 18 переменных. Для того чтобы построить квантовую схему, реализующую изменение знака амплитуды искомого состояния (см. алгоритм Гровера), требуется не менее 18 кубитов для реализации SDES и еще один флаговый кубит. Таким образом, на рис. 4–7 представлена минимальная по количеству логических кубитов квантовая схема, реализующая поиск ключа SDES с помощью алгоритма Гровера по одной паре открытого и шифрованного текстов.

Сравним общее количество квантовых вентилей в одной итерации Гровера. В работе [13] подсчету квантовых вентилей посвящен раздел «Complexity analysis», но их общее количество явно не приведено. Согласно [13], процедура выработки раундовых ключей SDES интегрирована в один шаг и требует 8 вентилей CNOT, распределение вентилей по остальным операциям SDES приведено в табл. 4.

В соответствии с алгоритмом зашифровывания SDES и табл. 4, пользуясь рис. 2 и 3, можем посчитать общее количество вентилей в квантовой схеме одной итерации Гровера из работы [13]. При подсчете необходимо учитывать процедуру обращения раундовых преобразований (количество вентилей, участвующих в раундовых преобразованиях, умножается на два), инвертирование флагового кубита (требуется один обобщенный CNOT), рассеивание Гровера (еще один обобщенный CNOT), а также тот факт, что перестановка двух кубитов (SWAP) выполняется с помощью трех CNOT.

В квантовой схеме на рис. 4–7 операции IP, P10, P8, E/P, LS-1, LS-2, P4 реализованы без использования каких-либо вентилей, что достигается простой перенумерацией кубитов, которую можно рассчитать заранее.

В табл. 5 представлено сравнение количества вентилей в одной итерации Гровера.

5. ЗАКЛЮЧЕНИЕ

В работе представлена минимальная по количеству кубитов (19 кубитов) квантовая схема, реализующая поиск ключа SDES по одной паре открытого и шифрованного текстов квантовым алгоритмом Гровера, в то время как в работе [13] соответствующая квантовая схема построена на 61 кубите.

С помощью квантового симулятора Quipper проведена симуляция работы построенных квантовых схем для 18 и 25 итераций Гровера, получены соответствующие распределения вероятностей успеха алгоритма Гровера в задаче поиска ключа SDES для случаев, когда для известной пары блоков открытого и шифрованного текстов (P, C) существует ровно один ключ $K \in V_{10}$, на котором $E_{SDES}(K, P) = C$, и два ключа $K_1, K_2 \in V_{10}$, на которых $E_{SDES}(K_1, P) = C$ и $E_{SDES}(K_2, P) = C$.

Контрольный пример учебного алгоритма блочного шифрования SDES приведен в Приложении 1, программная реализация алгоритма Гровера в задачах поиска одного и двух целевых значений представлена в Приложениях 2 и 3. Программная реализация поиска ключа SDES квантовым алгоритмом Гровера в квантовом симуляторе Quipper представлена в Приложении 4.

ПРИЛОЖЕНИЕ 1

Контрольный пример SDES

Выберем блок открытого текста P= 00101000 и K = 1100011110. Формирование раундового ключа k_1:

Номера битов	1	2	3	4	5	6	7	8	9	10
K	1	1	0	0	0	1	1	1	1	0
P10(K)	0	0	1	1	0	0	1	1	1	1
LS-1(P10(K))	0	1	1	0	0	1	1	1	1	1
k_1 = P8(LS-1(P10(K)))	1	1	1	0	1	0	0	1		

Формирование раундового ключа k_2:

Номера битов	1	2	3	4	5	6	7	8	9	10
K	1	1	0	0	0	1	1	1	1	0
P10(K)	0	0	1	1	0	0	1	1	1	1
LS-3(P10(K))	1	0	0	0	1	1	1	0	1	1
k_2 = P8(LS-3(P10(K)))	1	0	1	0	0	1	1	1		

В рассматриваемом примере $P = 00101000$, $\mathrm{IP}(P) = 00100010$.

1. Выберем $P = 00101000$ и $K = 1100011110$.
2. $\mathrm{IP}(P) = 00100010$.
3. $f_{k_1}(L, R) = f_{11101001}(00100010) = (0010 \oplus F(0010, 11101001), 0010)$.
4. $F(0010, 11101001) = \mathrm{P4} \circ \mathrm{Sboxes} \circ (11101001 \oplus \mathrm{E/P}(0010))$:

Номера битов	1	2	3	4	5	6	7	8
R	0	0	1	0				
E/P(R)	0	0	0	1	0	1	0	0
k_1	1	1	1	0	1	0	0	1
E/P(R) $\oplus$ k_1	1	1	1	1	1	1	0	1
Sboxes(E/P(R) $\oplus$ k_1)	1	0	0	0				
P4(Sboxes(E/P)$\oplus k_1$))	0	0	0	1				

5. Вычислили $F(0010, 11101001) = 0001$, получили $f_{k_1}(L, R) = (0011, 0010)$.
6. $\mathrm{SW}(0011, 0010) = (0010, 0011)$.
7. $f_{k_2}(L, R) = f_{10100111}(00100011) = (0010 \oplus F(0011, 10100111), 0011)$:

Номера битов	1	2	3	4	5	6	7	8
R	0	0	1	1				
E/P(R)	1	0	0	1	0	1	1	0
k_2	1	0	1	0	0	1	1	1
E/P(R) $\oplus$ k_2	0	0	1	1	0	0	0	1
Sboxes(E/P(R) $\oplus$ k_2)	1	0	1	0				
P4(Sboxes(E/P(R) $\oplus$ k_2))	0	0	1	1				

8. Вычислили $F(0011, 10100111) = 0011$, получили $f_{k_2}(L, R) = (0001, 0011)$.
9. Применили IP^{-1}:

Номера битов	1	2	3	4	5	6	7	8
L, R	0	0	0	1	0	0	1	1
$\mathrm{IP}^{-1}(L, R)$	1	0	0	0	1	0	1	0

Примеры результатов зашифровывания на ключе $K = 1100011110$:

$$E_{SDES}(K, 00101000) = 10001010,$$
$$E_{SDES}(K, 10001101) = 11010000,$$
$$E_{SDES}(K, 11110010) = 11011010,$$
$$E_{SDES}(K, 01010111) = 01100000.$$

ПРИЛОЖЕНИЕ 2

Поиск одного целевого значения с помощью алгоритма Гровера

Пусть $N = 2^3$, $M = 1$, целевое значение TargetValue=7. Определим вероятности успеха поиска целевого значения в зависимости от количества итераций алгоритма Гровера.

Листинг 1. Программная реализация в пакете Wolfram Mathematica

```
1 TargetValue=7; (*Зададим номер элемента, который хотим получить в результате
  измерения кубитов*)
2 NumberOfQubits=3; (*определили количество кубитов*)
3 H= HadamardMatrix[2^NumberOfQubits];
4 (*Инициализировали матрицу Адамара*)
5 Numb=2^NumberOfQubits; (*ввели дополнительную переменную для более короткой записи*)
6 matrixD=ConstantArray[ConstantArray[2/Numb,Numb],Numb]-IdentityMatrix[Numb];
7 (*Инициализировали матрицу D (рассеивание Гровера)*)
8 Print["Матрица Адамара: \n ",MatrixForm[H]];
```

$$H=\begin{pmatrix}
\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\
\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} \\
\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\
\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} \\
\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\
\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} \\
\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\
\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}}
\end{pmatrix}$$

```
9 Print["Матрица D (рассеивание Гровера): \n ",MatrixForm[matrixD]];
```

$$D=\begin{pmatrix}
-\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4}
\end{pmatrix}$$

```
10 FirstState=ConstantArray[0,2^NumberOfQubits];
11 FirstState[[1]]=1;
12 (*Инициализировали начальное состояние*)
13 Print["Инициализировали начальное состояние:",FirstState];
```

$$\text{FirstState} \;=\; \{1,0,0,0,0,0,0,0\}$$

```
14 State=FirstState.H; (*Применили гейты Адамара к каждому кубиту,
   т.е. умножили вектор FirstState на матрицу H*)
```

$$State{=}\left\{\frac{1}{2\sqrt{2}},\frac{1}{2\sqrt{2}},\frac{1}{2\sqrt{2}},\frac{1}{2\sqrt{2}},\frac{1}{2\sqrt{2}},\frac{1}{2\sqrt{2}},\frac{1}{2\sqrt{2}},\frac{1}{2\sqrt{2}}\right\}$$

```
15 i=0;
16 Print["Итерация №", i ," вероятность успеха = ", N[(State[[TargetValue]]*
   State[[TargetValue]])],"  вероятность неудачи = ",
   N[(1-State[[TargetValue]]*State[[TargetValue]])]  ];
```

Итерация №0, вероятность успеха = 0.125, вероятность неудачи = 0.875;

```
17 (*Изменение знака амплитуды целевого значения*)
18 State[[TargetValue]]=(-1)*State[[TargetValue]];
19 Print["изменили знак у целевого значения: \n",State];
```

$$State{=}\left\{\frac{1}{2\sqrt{2}},\frac{1}{2\sqrt{2}},\frac{1}{2\sqrt{2}},\frac{1}{2\sqrt{2}},\frac{1}{2\sqrt{2}},\frac{1}{2\sqrt{2}},-\frac{1}{2\sqrt{2}},\frac{1}{2\sqrt{2}}\right\}$$

```
20 State=State.matrixD; (*Применили рассеивание Гровера, умножили State на матрицу D*)
21 Print["Применили рассеивание Гровера, умножили State на матрицу D: \n",State];
```

$$State{=}\left\{\frac{1}{4\sqrt{2}},\frac{1}{4\sqrt{2}},\frac{1}{4\sqrt{2}},\frac{1}{4\sqrt{2}},\frac{1}{4\sqrt{2}},\frac{1}{4\sqrt{2}},\frac{5}{4\sqrt{2}},\frac{1}{4\sqrt{2}}\right\}$$

```
22 i=1;
23 Print["Итерация №", i ," вероятность успеха = ", N[(State[[TargetValue]]*
   State[[TargetValue]])],"  вероятность неудачи = ", N[(1-State[[TargetValue]]*
   State[[TargetValue]])]     ];
```

Итерация №1, вероятность успеха = 0.78125, вероятность неудачи = 0.21875

```
24 NumberOfGroverIterations=30;
25 (*задали количество итераций, например 30*)
26 For[i=2,i<=NumberOfGroverIterations,i++,
27 State[[TargetValue]]=(-1)*State[[TargetValue]];
   (*изменили знак у целевого значения*)
28 State=State.matrixD; (*Применили рассеивание Гровера, умножили State на матрицу D*)
29 p=State[[TargetValue]]*State[[TargetValue]];
30 Print["Итерация №", i ," вероятность успеха = ", N[p],"  вероятность неудачи = ", N[1-p]];
31 ]
```

Рассчитаем вероятности успешного поиска целевого значения (TargetValue=7) в зависимости от количества итераций алгоритма Гровера на трех кубитах (см. табл. 6). Оптимальное количество итераций Гровера в рассматриваемом примере оценивается величиной $\left[\frac{\pi}{4}\sqrt{\frac{2^3}{1}}\right] = [2.221441469079183] = 2.$

Таблица 6

Номер итерации Гровера	Вероятность успеха алгоритма Гровера, т. е. вероятность получить в результате измерения кубитов TargetValue=7	Вероятность неудачи алгоритма Гровера, т. е. вероятность получить в результате измерения кубитов TargetValue≠7
1	0.78125	0.21875
2	**0.945313**	**0.0546875**
3	0.330078	0.669922
4	0.012207	0.987793
5	0.547974	0.452026
6	**0.999786**	**0.000213623**
7	0.576973	0.423027
8	0.0194569	0.980543
9	0.302891	0.697109
10	**0.931266**	**0.068734**
11	0.804925	0.195075
12	0.144965	0.855035
13	0.106316	0.893684
14	0.756614	0.243386
15	**0.957837**	**0.0421627**
16	0.357846	0.642154
17	0.0066241	0.993376
18	0.51881	0.48119
19	**0.998078**	**0.00192151**
20	0.605709	0.394291
21	0.0283488	0.971651
22	0.276378	0.723622
23	**0.915746**	**0.0842543**
24	0.827558	0.172442
25	0.166144	0.833856
26	0.0889775	0.911022
27	0.7311	0.2689
28	**0.968798**	**0.0312024**

ПРИЛОЖЕНИЕ 3

Поиск одного из двух целевых значений с помощью алгоритма Гровера

Пусть $N = 2^{10}$, $M = 2$, целевое значение TargetValue $\in \{151, 223\}$. Определим вероятности успеха поиска целевого значения в зависимости от количества итераций алгоритма Гровера (см. табл. 7). Оптимальное количество итераций Гровера в рассматриваемом примере оценивается величиной $\left[\frac{\pi}{4}\sqrt{\frac{2^{10}}{2}}\right] = [17.771531752633464] = 18.$

Таблица 7

Номер итерации Гровера	Вероятность успеха алгоритма Гровера, т. е. вероятность получить в результате измерения кубитов одно из TargetValues = {151,223}	Вероятность неудачи алгоритма Гровера, т. е. вероятность получить в результате измерения кубитов значение не из TargetValues
1	0.0174867	0.982513
2	0.0480693	0.951931
3	0.0927473	0.907253
4	0.150127	0.849873
5	0.218419	0.781581
6	0.295493	0.704507
7	0.378945	0.621055
8	0.466173	0.533827
9	0.554456	0.445544
10	0.641041	0.358959
11	0.723227	0.276773
12	0.79845	0.20155
13	0.864365	0.135635
14	0.918916	0.0810837
15	0.960402	0.0395983
16	0.987528	0.0124724
17	*0.999448*	*0.000551974*
18	**0.995791**	**0.0042088**
19	0.976671	0.0233288
20	0.942684	0.0573158
21	0.89489	0.10511
22	0.83478	0.16522
23	0.764229	0.235771
24	0.685436	0.314564
25	0.60086	0.39914
26	0.513139	0.486861
27	0.425007	0.574993
28	0.339214	0.660786

Листинг 2. Программная реализация в пакете Wolfram Mathematica

```
1 (*Случай когда несколько искомых номеров*)
2 TargetValues={151,223};  (*искомые номера*)
3 NumberOfQubits=10;   (*определили количество кубитов*)
4 H= HadamardMatrix[2^NumberOfQubits]; (*Инициализировали матрицу Адамара*)
5 Numb=2^NumberOfQubits; (*ввели доп. переменную для более короткой записи*)
```

```
6 matrixD=ConstantArray[ConstantArray[2/Numb,Numb],Numb]-IdentityMatrix[Numb];
7 (*Инициализировали матрицу D (рассеивание Гровера)*)
8 FirstState=ConstantArray[0,2^NumberOfQubits];
9 FirstState[[1]]=1; (*Инициализировали начальное состояние*)
10 State=FirstState.H;
11 (*Применили гейты Адамара к каждому кубиту, т.е. умножили вектор FirstState на матрицу H*)
```

$$\mathrm{p} = \sum_{k=1}^{\mathrm{Length[TargetValues]}} \mathrm{State}\ [[\mathrm{TargetValues}\ [[k]]]] * \mathrm{State}\ [[\mathrm{TargetValues}[[k]]]];$$

```
12 i=0;Print["Итерация №", i ,", вероятность успеха = ", N[p],",
   вероятность неудачи = ", N[1-p]];
13 (*Изменение знака у амплитуды целевого значения*)
14 For[i=1,i<=Length[TargetValues],i++,
15 State[[ TargetValues[[i]] ]]=(-1)*State[[ TargetValues[[i]] ]];   ];
16 State=State.matrixD;
17 (*Применили рассеивание Гровера, умножили вектор State на матрицу D*)
```

$$\mathrm{p} = \sum_{k=1}^{\mathrm{Length[TargetValues]}} \mathrm{State}\ [[\mathrm{TargetValues}\ [[k]]]] * \mathrm{State}\ [[\mathrm{TargetValues}[[k]]]];$$

```
18 Print["Итерация №", i ,", вероятность успеха = ", N[p],",  вероятность неудачи = ", N[1-p]];
19 NumberOfGroverIterations=30;
20 (*задали количество итераций Гровера*)
21 For[i=2,i<=NumberOfGroverIterations,i++,
22 For[j=1,j<=Length[TargetValues],j++,
23 State[[TargetValues[[j]]]]=(-1)*State[[TargetValues[[j]] ]];];
24 State=State.matrixD;
```

$$\mathrm{p} = \sum_{k=1}^{\mathrm{Length[TargetValues]}} \mathrm{State}\ [[\mathrm{TargetValues}\ [[k]]]] * \mathrm{State}\ [[\mathrm{TargetValues}[[k]]]];$$

```
25 Print["Итерация №", i ,", вероятность успеха = ", N[p],",  вероятность неудачи = ",
N[1-p]];]
```

Вероятность успеха после 17 итераций Гровера оказалась чуть больше, чем вероятность успеха после 18 итераций Гровера. Это ничему не противоречит, так как $\left[\frac{\pi}{4}\sqrt{\frac{N}{M}}\right]$ — верхняя оценка числа итераций (см. [12], с. 318).

ПРИЛОЖЕНИЕ 4

Программная реализация алгоритма Гровера для поиска ключа SDES по паре открытого и шифрованного текстов на квантовом симуляторе Quipper

Программная реализация представляет собой три файла:

1) QSDES.hs — реализация алгоритма SDES,

2) Grover.hs — реализация алгоритма Гровера на базе QSDES,

3) Main.hs — запуск симуляции квантовой схемы.

Приведем содержимое указанных файлов.

QSDES.HS

```
1 module QSDES where
2 -- Модуль квантовой реализации схемы шифрования SDES
3
4 import Quipper
```

```
5 import QuipperLib.Simulation
6 import System.Random
7 import Quipper.Printing
8 import Quipper.QData
9
10 --Суммирование с ключом
11 sum_qubit :: ([Qubit], [Qubit]) -> Circ [Qubit]
12 sum_qubit ([k0, k1, k2, k3], [x0, x1, x2, x3]) = do
13  qnot_at x0 `controlled` [k0]
14  qnot_at x1 `controlled` [k1]
15  qnot_at x2 `controlled` [k2]
16  qnot_at x3 `controlled` [k3]
17  return [x0, x1, x2, x3]


19 -- S0
20 s0_box :: ([Qubit], [Qubit]) -> Circ [Qubit]
21 s0_box ([x0, x1, x2, x3], [s0, s1])  = do
22  comment "S0_box"
23  qnot_at s0 `controlled` [x3]
24  qnot_at s0 `controlled` [x0, x1, x2, x3]
25  qnot_at s0 `controlled` [x0, x2]
26  x0 <- gate_X x0
27  qnot_at s0 `controlled` [x0, x1]
28  x2 <- gate_X x2
29  qnot_at s1 `controlled` [x0, x2]
30  x0 <- gate_X x0
31  qnot_at s1 `controlled` [x0, x1, x2, x3]
32  x1 <- gate_X x1
33  qnot_at s1 `controlled` [x0, x1]
34  x3 <- gate_X x3
35  qnot_at s1 `controlled` [x0, x3]
36  x1 <- gate_X x1
37  x2 <- gate_X x2
38  x3 <- gate_X x3
39  return [s0, s1]
40
41 -- S1
42 s1_box :: ([Qubit], [Qubit]) -> Circ [Qubit]
43 s1_box ([x0, x1, x2, x3], [s2, s3])  = do
44  comment "S1_box"
45  qnot_at s2 `controlled` [x1]
46  qnot_at s3 `controlled` [x0, x2, x3]
47  x3 <- gate_X x3
48  qnot_at s3 `controlled` [x0, x3]
49  qnot_at s2 `controlled` [x0, x3]
50  x1 <- gate_X x1
51  qnot_at s2 `controlled` [x0, x1, x2, x3]
52  qnot_at s3 `controlled` [x2, x3]
53  x2 <- gate_X x2
54  x3 <- gate_X x3
55  qnot_at s2 `controlled` [x2, x3]
```

```
56  x0 <- gate_X x0
57  x1 <- gate_X x1
58  qnot_at s3 ‘controlled‘ [x0, x1, x3]
59  x0 <- gate_X x0
60  x2 <- gate_X x2
61  return [s2, s3]
62
63 ----------------------1 раунд----------------------
64
65 round1 :: ([Qubit], [Qubit]) -> Circ [Qubit]
66 round1 ([k0, k1, k2, k3, k4, k5, k6, k7, k8, k9], [x0, x1, x2, x3, x4, x5, x6, x7]) = do
67  comment "ROUND_1"
68
69  [x6, x3, x7, x4] <- sum_qubit ([k0, k6, k8, k3], [x6, x3, x7, x4])
70  [x0, x1] <- s0_box ([x6, x3, x7, x4], [x0, x1])
71  [x6, x3, x7, x4] <- sum_qubit ([k0, k6, k8, k3], [x6, x3, x7, x4])
72
73  [x7, x4, x6, x3] <- sum_qubit ([k7, k2, k9, k5], [x7, x4, x6, x3])
74  [x2, x5] <- s1_box ([x7, x4, x6, x3], [x2, x5])
75  [x7, x4, x6, x3] <- sum_qubit ([k7, k2, k9, k5], [x7, x4, x6, x3])
76
77  return [x0, x1, x2, x3, x4, x5, x6, x7]
78
79 ----------------------2 раунд----------------------
80
81 round2 :: ([Qubit], [Qubit]) -> Circ [Qubit]
82 round2 ([k0, k1, k2, k3, k4, k5, k6, k7, k8, k9], [x0, x1, x2, x3, x4, x5, x6, x7]) = do
83  comment "ROUND_2"
84
85  [x0, x1, x5, x2] <- sum_qubit ([k7, k2, k5, k4], [x0, x1, x5, x2])
86  [x6, x3] <- s0_box ([x0, x1, x5, x2], [x6, x3])
87  [x0, x1, x5, x2] <- sum_qubit ([k7, k2, k5, k4], [x0, x1, x5, x2])
88
89  [x5, x2, x0, x1] <- sum_qubit ([k9, k1, k8, k0], [x5, x2, x0, x1])
90  [x4, x7] <- s1_box ([x5, x2, x0, x1], [x4, x7])
91  [x5, x2, x0, x1] <- sum_qubit ([k9, k1, k8, k0], [x5, x2, x0, x1])
92
93  return [x0, x1, x2, x3, x4, x5, x6, x7]
94
95 -- Схема QSDES ("прямая").
96 sdes :: ([Qubit], [Qubit]) -> Circ ([Qubit], [Qubit])
97 sdes (key, plaintext) = do
98  let [k0, k1, k2, k3, k4, k5, k6, k7, k8, k9] = key
99  let [x0, x1, x2, x3, x4, x5, x6, x7] = plaintext
100
101  comment_with_label "Key"
102  (k0, k1, k2, k3, k4, k5, k6, k7, k8, k9)
103  ("k0", "k1", "k2", "k3", "k4", "k5", "k6", "k7", "k8", "k9")
104
105  comment_with_label "Plaintext"
106  (x0, x1, x2, x3, x4, x5, x6, x7)
107  ("x0", "x1", "x2", "x3", "x4", "x5", "x6", "x7")
```

```
108
109  [x0, x1, x2, x3, x4, x5, x6, x7] <- round1 ([k0, k1, k2, k3, k4, k5, k6, k7, k8, k9],
     [x0, x1, x2, x3, x4, x5, x6, x7])
110  [x0, x1, x2, x3, x4, x5, x6, x7] <- round2 ([k0, k1, k2, k3, k4, k5, k6, k7, k8, k9],
     [x0, x1, x2, x3, x4, x5, x6, x7])
111
112  swap x6 x0
113  swap x3 x1
114  swap x4 x2
115  swap x5 x7
116
117  return ([k0, k1, k2, k3, k4, k5, k6, k7, k8, k9], [x0, x1, x2, x3, x4, x5, x6, x7])
118
119 -- Схема QSDES ("обращенная").
120 sdes_reverse :: ([Qubit], [Qubit]) -> Circ ([Qubit], [Qubit])
121 sdes_reverse = reverse_generic_endo sdes
122
123 -- Тестовая схема QSDES с ключом, приводимым в суперпозицию.
124 -- Используется для теста № 2.
125 sdes_key_superposition :: [Bool] -> Circ ([Qubit], [Qubit])
126 sdes_key_superposition plaintext = do
127  key_in_superposition <- qinit (replicate 10 False)
128  mapUnary hadamard key_in_superposition
129  plaintext <- qinit plaintext
130  (key, cyphertext) <- sdes (key_in_superposition, plaintext)
131  return (key, cyphertext)
```

GROVER.HS

```
1 module Grover where
2 -- Алгоритм Гровера для QSDES
3
4 import Quipper
5 import Quipper.QData
6 import QSDES
7 -- Оракул для QSDES.
8 sdes_oracle :: ([Qubit], [Qubit], [Bool], Qubit) -> Circ Qubit
9 sdes_oracle (key, plaintext, cyphertext_bool, oracle) = do
10  (key, cyphertext) <- sdes (key, plaintext)
11  qnot_at oracle `controlled` cyphertext .==. cyphertext_bool
12  (key, plaintext) <- sdes_reverse (key, cyphertext)
13  return oracle
14 -- Схема inversion about the mean или Conditional Phase Flip (CPF)
15 inversion_about_mean ::  [Qubit] -> Circ [Qubit]
16 inversion_about_mean top_qubits = do
17  comment "start inversion about mean"
18  mapUnary hadamard top_qubits
19  mapUnary gate_X top_qubits
20  let pos = (length top_qubits) - 1
21  let target_qubit = top_qubits !!  pos
22  let controlled_qubit = take pos top_qubits
```

```
 hadamard_at target_qubit
 qnot_at target_qubit `controlled` controlled_qubit
 hadamard_at target_qubit
 mapUnary gate_X top_qubits
 mapUnary hadamard top_qubits
 comment "end inversion about mean"
 return top_qubits
-- Алгоритм Гровера для QSDES
grover_search_circuit_sdes ::  (Int, [Bool], [Bool]) -> Circ ([Bit], Bit)
grover_search_circuit_sdes (iterations_num, plaintext, cyphertext) = do
 key <- qinit (replicate 10 False)
 plaintext <- qinit plaintext
 oracle <- qinit True
 mapUnary hadamard key
 hadamard_at oracle
 -- Начало итераций Гровера
 let index = iterations_num
 for 1 (index) 1 $ \i -> do
   comment "start grover iteration"
   oracle <- sdes_oracle (key, plaintext, cyphertext, oracle)
   key <- inversion_about_mean key
   comment "after grover iteration"
 endfor
 -- Измерение кубитов, возвращение результата.
 hadamard_at oracle
 (key, oracle) <- measure (key, oracle)
 --cdiscard oracle
 return (key, oracle)
```

MAIN.HS

```
module Main where
-- Основной модуль. Запуск примеров.

import System.Random
import Data.Time
import Quipper
import Quipper.Printing
import QuipperLib.Simulation
import Quipper.QData
import QSDES
import Grover

-- ЗАПУСК ПРОГРАММЫ
main :: IO ()
--main = test1_circuit
main = test3_exec --25 итераций Гровера,выведет распределение ключей.

-- Тест функциональности. Проверка правильности составления схемы.
test1_circuit :: IO ()
test1_circuit = do
```

```
21  putStrLn "QSDES functionality test:"
22  print_generic GateCount sdes  ((replicate 10 qubit),(replicate 8 qubit))
23  print_generic PDF sdes ((replicate 10 qubit),(replicate 8 qubit))
24
25 test1_exec :: IO ()
26 test1_exec = do
27  putStrLn "QSDES functionality test:"
28  print_generic GateCount sdes  ((replicate 10 qubit),(replicate 8 qubit))
29  g <- newStdGen
30  print $ run_generic g (0.0 :: Double) sdes ([True,True,False,False,False,True,
    True,True,True,False], [False,False,True,False,True,False,False,False])
31  print $ run_generic g (0.0 :: Double) sdes ([True,True,False,False,False,True,True,
    True,True,False],   [True,False,False,False,True,True,False,True])
32  print $ run_generic g (0.0 :: Double) sdes ([True,True,False,False,False,True,True,
    True,True,False],   [True,True,True,True,False,False,True,False])
33  print $ run_generic g (0.0 :: Double) sdes ([True,True,False,False,False,True,True,
    True,True,False],   [False,True,False,True,False,True,True,True])
34
35 -- Тест схемы при приведении ключевых кубитов в суперпозицию.
36 test2_circuit :: IO ()
37 test2_circuit = do
38  putStrLn "QSDES results when key is in superposition:"
39  print_generic PDF (sdes_key_superposition [True,False,False,True,True,False,True,False])
40
41 test2_exec :: IO ()
42 test2_exec = do
43  putStrLn "QSDES results when key is in superposition:"
44 --  t1 <- getZonedTime
45 --  putStrLn $ formatTime defaultTimeLocale "%FT%T%z" t1
46  g <- newStdGen
47  print $ sim_generic undefined (sdes_key_superposition
    ([True,False,False,True,True,False,True,False]))
48 --  t2 <- getZonedTime
49 --  putStrLn $ formatTime defaultTimeLocale "%FT%T%z" t2
50
51
52 -- Поиск ключа по паре открытого и закрытого текстов.
53 -- Пример с одним подходящим ключом.
54 test3_circuit :: IO()
55 test3_circuit = do
56  putStrLn "Quantum exhaustive key search (1 key):"
57  let parameters = (25,[False,False,False,True,False,False,False,False],
58 [False,False,True,True,False,False,True,True])
59  print_generic GateCount (grover_search_circuit_sdes parameters)
60 --  print_generic PDF (grover_search_circuit_sdes parameters)
61
62 test3_exec :: IO()
63 test3_exec = do
64  putStrLn "Quantum exhaustive key search (1 key):"
65  startTime <- getCurrentTime
66  let parameters = (25,[False,False,False,True,False,False,False,False],
    [False,False,True,True,False,False,True,True]) -- O.T.,C.T.
```

```
67  g <- newStdGen
68 --  print $ run_generic g (0.0 :: Double) (grover_search_circuit_sdes parameters)
69  print_generic GateCount       (grover_search_circuit_sdes parameters)
70  print $ sim_generic undefined (grover_search_circuit_sdes parameters)
71  stopTime <- getCurrentTime
72  let deltaTime = show $ diffUTCTime stopTime startTime
73  putStrLn deltaTime
74
75 -- Поиск ключа по паре открытого и закрытого текстов.
76 -- Пример с двумя подходящими ключами.
77 test4_circuit :: IO()
78 test4_circuit = do
79  putStrLn "Quantum exhaustive key search (2 keys):"
80  let parameters = (18,[True,False,True,False,False,True,False,True],
    [False,False,True,True,False,True,True,False])
81  print_generic GateCount (grover_search_circuit_sdes parameters)
82 --  print_generic PDF (grover_search_circuit_sdes parameters)
83
84 test4_exec :: IO()
85 test4_exec = do
86  putStrLn "Quantum exhaustive key search (2 keys):"
87  startTime <- getCurrentTime
88  let parameters = (18,[True,False,True,False,False,True,False,True],
    [False,False,True,True,False,True,True,False])
89  print_generic GateCount (grover_search_circuit_sdes parameters)
90  g <- newStdGen
91  --print $ run_generic g (0.0 :: Double) (grover_search_circuit_sdes parameters)
92  print $ sim_generic undefined (grover_search_circuit_sdes parameters)
93  stopTime <- getCurrentTime
94  let deltaTime = show $ diffUTCTime stopTime startTime
95  putStrLn deltaTime
```

ЛИТЕРАТУРА

1. H. Bernien, S. Schwartz, A. Keesling, H. Levine, and A. Omran, Nature **551**, 579 (2017); DOI:10.1038/nature24622; arXiv:1707.04344.

2. J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, and P. Becker, Nature **551**, 601 (2017); DOI:10.1038/nature24654; arXiv:1708.01044.

3. https://www.ibm.com/blogs/research/2017/11/the-future-is-quantum/.

4. M. Veldhorst, H. G. J. Eenink, C. H. Yang, and A. S. Dzurak, Nature Comm. **8**, 1766 (2017); DOI:10.1038/s41467-017-01905-6; https://doi.org/10.1038/s41467-017-01905-6.

5. C. S. Calude and E. Calude, arXiv:1712.01356v1.

6. J. Kelly, *A Preview of Bristlecone, Google's New Quantum Processor. Quantum AI Lab*, 05.03.2018, https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html.

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

8. D. R. Simon, SIAM J. Comput. **26**, 1474 (1997).

9. M. Kaplan, G. Leurent, A. Leverrier et al., Lect. Notes Comp. Sci., Vol. **9815**, Berlin, Springer-Verlag (2016).

10. L. K. Grover, Proc. STOC 1996, in ed. by G. L. Miller, ACM (1996), p. 212.

11. G. Brassard, P. Hoyer, M. Mosca et al., arXiv: quant-ph/0005055.

12. М. Нильсен, И. Чанг, *Квантовые вычисления и квантовая информация*, Мир, Москва (2006).

13. M. Almazrooie, A. Samsudin, R. Abdullah, and K. N. Mutter, SpringerPlus **5**, 1494 (2016); DOI: 10.1186/s40064-016-3159-4.

14. *Квантовый симулятор libquantum*, http://www.libquantum.de.

15. A. S. Green, P. L. Lumsdaine, N. J. Ross, P. Selinger, and B. Valiron, arXiv:1304.5485v1.

16. S. Siddiqui, M. J. Islam, and O. Shehab, arXiv:1406.4481v2 [quant-ph].

17. *Квантовый симулятор Quipper*, http://www.mathstat.dal.ca/∼selinger/quipper/.

18. К. Шеннон, *Теория связи в секретных системах*, в кн. *Работы по теории информации и кибернетике*, Изд-во иностр. лит., Москва (1963), с. 333.