Квантовые алгоритмы и их влияние на безопасность современных классических криптографических систем

Обзор

А.Ю. Богданов и И.С. Кижватов,
Российский государственный гуманитарный университет,
Факультет защиты инфомации

В статье дается обзор проблем безопасности симметричных и асимметричных алгоритмов классической криптографии, которые могут возникнуть с появлением квантового компьютера, обладающего достаточно длинным квантовым регистром. Более детально рассматривается квантовый алгоритм Шора для факторизации чисел и необходимые для его реализации квантовое преобразование Фурье и квантовое возведение в степень. Проводится сравнительный анализ квантовой сложности факторизации, дискретного логарифмирования в группе точек эллиптической кривой и переборной задачи для различных длин ключей.

Содержание

Список иллюстраций

Список таблиц

Введение

Под современными классическими криптографическими системами понимаются алгоритмы симметричной и асимметричной криптографии, безопасность которых основывается в сущности на сложности решения определенных классов задач на классических компьютерах (перебор, факторизация и дискретное логарифмирование лежат в классе сложности NPP по длине ключа в битах), в противоположность квантовой криптографии, стойкость которой зиждется на законах квантовой физики. Симметричные криптографические алгоритмы, делящиеся на поточные и блочные, имеют одну природу в том смысле, что базируются на отмеченной выше сложности задачи перебора. Асимметричные криптоалгоритмы основаны на сложности вычисления дискретного логарифма в конечных группах, определенных над различными алгебраическими конструкциями, или на сложности разложения натурального числа на простые сомножители. В данной работе будет рассматриваться степень уязвимости асимметричных криптосистем, базирующиеся на вычислительной сложности дискретного логарифмирования в мультипликативной группе простого поля F_p^* (ElGamal, DSA, схема аутентификации Шнорра, ГОСТ Р34.10–94 и др.), дискретного логарифмирования в группе точек на эллиптической кривой (ECDSA, ECGDSA, эллиптический аналог ElGamal, ГОСТ Р34.10–2001 и др.) и факторизации чисел (схемы аутентификации и ЭЦП Фиата-Шамира, RSA и др.), а также безопасность симметричных криптосистем, независимо от того, являются ли они блочными или поточными. Особого рассмотрения требует совершенно стойкий по Шеннону шифр.

Все перечисленные типы криптосистем оказываются уязвимыми к атакам с использованием квантового компьютера. Следует отметить, что эта уязвимость выражена для каждого типа в различной степени. Пусть n выражает длину ключа в битах. Тогда для симметричных криптосистем (не совершенный шифр) квантовый прогресс в криптоанализе тотальным перебором выражается как

$$\mid K \mid^{1/2} = 2^{n/2},$$

т.е. как корень квадратный от мощности ключевого пространства. Это происходит благодаря алгоритму поиска Гровера. На классическом компьютере переборная задача решается в среднем за

$$\frac{\mid K \mid}{2} = 2^{n-1}$$

шагов. Задачи дискретного логарифмирования и факторизации имеет асимптотическую сложность при решении на квантовом компьютере

$$O(n^2 \log n \log\log n)$$

квантовых шагов. На классическом компьютере с использованием лучшего из известных алгоритмов — алгоритма решета числового поля — асимптотическая сложность составляет

$$O(e^{cn^{1/3}\log^{2/3} n})$$

, где c — некоторая константа. Полиномиальное решение ECDLP на квантовом компьютере находится также за

$$O(n^2 \log n \log\log n)$$

квантовых шагов, где n — битовая длина представления элементов базового поля, над которым определена эллиптическая кривая. Но данная сложность имеет существенно больший коэффициент, чем сложность факторизации соответствующего числа в предположении, что рассматриваемые проблемы имеют одинаковую сложность решения на классическом компьютере. Для проблемы дискретного логарифмирования в группе точек общей эллиптической кривой на классическом компьютере не существует субэкспоненциального алгоритма, что означает сложность решения данной проблемы эквивалентной сложности ρ-метода Полларда, т.е. $\sqrt{\pi 2^n}$ классическим операциям.

Работа устроена следующим образом. В разделе (1) вводится понятие квантового дискретного преобразования Фурье и рассматриваются его основные свойства, необходимые для всех вариантов алгоритма Шора. В разделе (2) описывается базовый алгоритм Шора, факторизующий числа за полиномиальное время. В заключении (2.6) даются общие соображения и выводы по безопасности классических криптографических схем в новых условиях, связанных с разработкой квантового компьютера.

В работе над темой были использованы следующие источники. В качестве введения в квантовые вычисления и квантовую теорию информации использована книга [16]. Основы квантовых алгоритмов были рассмотрены по работе [11]. Алгоритм факторизации Шора и его модификация для дискретного логарифмирования в представлены в статьях [10], [12] и [9]. Информация по решению ECDLP на квантовом компьютере была взята из работы [8]. Описание и интерпретация алгоритма Гровера осуществлялись на основе работ [11] и [14]. Классические методы факторизации и дискретного логарифмирования можно найти в работах [5], [6], [13]. Подробный обзор классических арифметических алгоритмов представлен в [15]. Классическое дискретное преобразование Фурье и быстрое дискретное преобразование Фурье рассматривались по книге [1].

1 Классическое и квантовое дискретное преобразование Фурье

При проведении классических, а с появлением квантовой информатики - и квантовых вычислений дискретное преобразование Фурье (discrete Fourier transform, DFT) применяется для того, чтобы определить периодическую структуру последовательности в общем случае комплексных значений конечной длины [1]. Примером задачи, для решения которой нужно узнать, периоды какой длины присутствуют в последовательности, является рассматриваемый в данной работе квантовый алгоритм факторизации. В классических вычислениях DFT применяется, например, при умножении чисел методом Шонхаге-Штрассена [15].

1.1 Дискретное преобразование Фурье

Классическое дискретное преобразование Фурье (discrete Fourier transform, DFT) последовательности значений $h_0, h_1, ..., h_{N-1}, h_i \in C$, определяется как

$$H_k = \sum_{n=0}^{N-1} e^{2\pi ikn/N} h_n.$$

Получаемый в результате набор комплексных значений $H_0, H_1, ..., H_{N-1}$ называется дискретным спектром исходной последовательности, а сами значения - спектральными коэффициентами. Чем больше спектральный коэффициент H_i по своей абсолютной величине $|H_i|^2$, тем большую амплитуду имеют в последовательности периоды длины $N/(i+1)$. Прямому DFT соответствует обратное преобразование

$$h_k = 1/N \sum_{n=0}^{N-1} e^{-2\pi ikn/N} H_n.$$

Если переписать формулу DFT в виде

$$H_k = \sum_{n=0}^{N-1} W^{nk} h_k,$$

где $W = e^{2\pi i/N}$, становится видно, что строка спектральных коэффициентов H_k получается умножением строки значений h_k на матрицу, элементами которой являются W^{nk}. Матрица имеет размерность $N \times N$, поэтому сложность вычисения DFT таким способом составляет $O(N^2)$.

1.2 Быстрое преобразование Фурье

Однако вычисление DFT может быть произведено за $O(N \log N)$ умножений. Такое преобразование получило название быстрого преобразования Фурье (fast Fourier transform, FFT). FFT основывается на лемме Даниэльсона-Ланцоша, согласно которой преобразование Фурье чётного размера N может быть представлено в виде суммы двух преобразований Фурье размера $N/2$. Действительно,

$$H_k = \sum_{n=0}^{N-1} e^{2\pi ikn/N} h_n = \sum_{n=0}^{N/2-1} e^{2\pi ik(2n)/N} h_{2n} + \sum_{n=0}^{N/2-1} e^{2\pi ik(2n+1)/N} h_{2n+1} =$$

$$= \sum_{n=0}^{N/2-1} e^{2\pi ikn/(N/2)} h_{2n} + \sum_{n=0}^{N/2-1} e^{2\pi ik/N} e^{2\pi ikn/(N/2)} h_{2n+1} = H^0 + W^k H^1.$$

H^0 является преобразованием Фурье чётных, а H^1 — нечётных элементов исходной последовательности. Каждое из этих преобразований вычисляется за $\left(\frac{N}{2}\right)^2$ умножений, и ещё N умножений дают дополнительные множители W^k. Получается, что исходное преобразование, требующее N^2 умножений, в результате разложения оказывается проще:

$$2\left(\frac{N}{2}\right)^2 + N < N^2, N > 2.$$

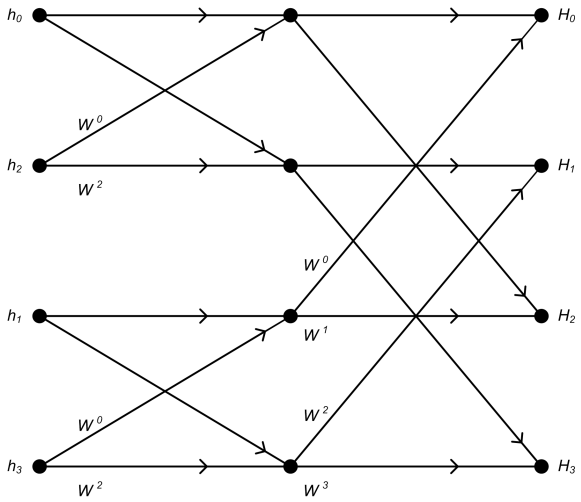

Рис. 1: Схема вычисления FFT размера 4

Если размер исходной последовательности является степенью двойки, $N = 2^m$, то последовательное применение леммы Даниэльсона-Ланцоша приведёт к преобразованиям Фурье единичного размера, которые представляют собой просто значения исходной последовательности:

$$H_k^{010...10} = h_n.$$

Поскольку каждое применение леммы является проверкой младших бит индексов значений, то, прочитывая строку 010...10 справа налево, получим двоичный индекс n значения h_n исходной последовательности, соответствующего данному единичному преобразованию Фурье.

Такой способ нахождения значений единичных преобразований Фурье позволяет выполнить полное преобразование следующим образом. Сначала производится перестановка значений исходной последовательности, в ходе которой двоичный индекс значения заменяется своим обратным прочтением. Затем в соответствии с леммой Даниэльсона-Ланцоша последовательно вычисляются преобразования Фурье размера 2, 4 и так далее до получения полного преобразования Фурье размера N. Процесс вычисления FFT размера 4 представлен в виде графа на рисунке 1.

Таким образом, FFT размера N вычисляется за $\log N$ шагов, на каждом из которых производится N умножений. Перестановка значений исходной последовательности требует порядка N операций. В итоге полная сложность преобразования, проведённого таким способом, составляет $O(N \log N)$. Рассмотренный алгоритм называется прореживанием по времени. Существует также алгоритм прореживания по частоте, который имеет такую же сложность.

1.3 Квантовое преобразование Фурье

Квантовое преобразование Фурье (quantum Fourier transform, QFT) [10] переводит состояние квантового регистра $|a\rangle$, где $a \in \mathbb{Z}$, $0 \leq a < q$ для некоторого q, в

состояние

$$\frac{1}{q^{1/2}}\sum_{c=0}^{q-1}|c\rangle e^{2\pi iac/q}.$$

Таким образом, действие QFT описывается применением унитарной матрицы, значение которой с индексом (a,c) равно $\frac{1}{q^{1/2}}e^{2\pi iac/q}$. Закрепим за этой матрицей обозначение A_q. В алгоритме факторизации используется матрица A_q для q экспоненциального размера. В этом случае QFT может быть выполнено за полиномиальное время, если q является степенью 2 (фактически по принципу FFT), а также если q принадлежит к специальному классу гладких чисел с малыми простыми множителями.

Ниже рассматривается построение A_q для $q=2^l$, впервые независимо предложенное Копперсмитом [3] и Дойчем. Для этого потребуется два типа квантовых вентилей. Первый из них — это вентиль Адамара, действующий на j-том кубите квантового регистра,

$$R_j=\frac{1}{\sqrt{2}}\begin{vmatrix}1 & 1\\ 1 & -1\end{vmatrix}.$$

Второй вентиль действует на кубитах с номерами j и k, $j<k$, и определяется как

$$S_j=\begin{vmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & e^{i\frac{\pi}{2^{k-j}}}\end{vmatrix}.$$

Вычисление QFT производится применением вентилей в порядке (слева направо)

$$R_{l-1}S_{l-2,l-1}R_{l-2}S_{l-3,l-1}S_{l-3,l-2}R_{l-3}\dots R_1S_{0,l-1}S_{0,l-2}\dots S_{0,2}S_{0,1}R_0,$$

то есть вентили R_j применяются в обратном порядке от R_{l-1} до R_0, и между каждыми R_j, R_{j+1} применяются вентили $S_{j,k}$ для всех $k>j$. В результате этих операций мы получим состояние

$$\frac{1}{q^{1/2}}\sum_b e^{2\pi iac/q}|b\rangle,$$

где b является двоичным представлением c, записанным в обратном порядке. Таким образом, для QFT размером $q=2^l$ необходимо $\frac{l(l-1)}{2}$, то есть $O(l^2)$, квантовых вентилей.

Для того, чтобы показать, что приведенная операция действительно является квантовым преобразованием Фурье, рассмотрим амплитуду перехода от $|a\rangle = |a_{l-1}\dots a_0\rangle$ к $|b\rangle = |b_{l-1}\dots b_0\rangle$. В результате перемножения матриц R коэффициенты $\frac{1}{\sqrt{2}}$ дадут множитель $\frac{1}{\sqrt{q}}$, и поэтому осталось определить, как получается фаза $2\pi iac/q$ в множителе $e^{2\pi iac/q}$. Каждое умножение на матрицу Адамара R_j добавляет к фазе π, если биты a_j и b_j оба равны 1, и оставляет фазу неизменной в остальных случаях. Действительно, в случае $a_j=b_j=1$ происходит умножение на $-1=\cos\pi+i\sin\pi=e^{i\pi}$. Далее, каждое умножение на матрицу $S_{j,k}$ увеличивает фазу на $\frac{\pi}{2^{k-j}}$ только при при $a_j=b_k=1$. Следовательно, при переходе от $|a\rangle$ к $|b\rangle$ фаза имеет вид

$$\sum_{0\le j<l}\pi a_jb_j+\sum_{0\le j<k<l}\frac{\pi}{2^{k-j}}a_jb_k.$$

Поскольку в этом выражении первая сумма является частным случаем второй при $j = k$, его можно переписать в виде

$$\sum_{0 \le j \le k < l} \frac{\pi}{2^{k-j}} a_j b_k,$$

а так как c является обращенным двоичным представлением b, то получим

$$\sum_{0 \le j \le k < l} \frac{\pi}{2^{k-j}} a_{jl-1-k}.$$

Заменяя в сумме k на $l - k - 1$, записываем ее как

$$\sum_{0 \le j+k < l} 2\pi \frac{2^j 2^k}{2^l} a_{jk}.$$

Наконец, если расширить диапазон суммирования на все j и k, меньшие l, то при $j + k \ge l$ выражение $\frac{2^j 2^k}{2^l}$ примет целое значение, а прибавление к фазе слагаемых, кратных 2π, не изменяет значение множителя. Следовательно, получаем

$$\sum_{j,k=0}^{l-1} 2\pi \frac{2^j 2^k}{2^l} a_j c_k = \frac{2\pi}{2^l} \sum_{j=0}^{l-1} 2^j a_j \sum_{k=0}^{l-1} 2^k c_k = \frac{2\pi ac}{q}.$$

Это выражение является фазой для амплитуды перехода от $|a\rangle$ к $|b\rangle$.

Итак, QFT сводится к последовательности $O(l^2)$ квантовых вентилей, если $q = 2^l$. Но для этого требуются вентили с малыми фазами, которые на практике могут оказаться трудно реализуемыми с любой точностью. Копперсмит показал [3], что можно исключить такие вентили и получить приближенное преобразование Фурье, точности которого будет достаточно для того, чтобы вероятность успеха квантового алгоритма Шора осталась приемлемой. При этом число необходимых для QFT вентилей уменьшается до $O(l \log l)$.

2 Алгоритм факторизации Шора

В данном разделе рассматривается квантовый алгоритм Шора для факторизации натуральных чисел. Всюду в этом разделе n — факторизуемое число. Для факторизации чисел на классическом компьютере на настоящий момент существует два основных субэкспоненциальных метода: метод квадратичного решета (предложен К. Померансом в 1981 году, см. [13]) и метод решета числового поля (первая версия, т.н. SNFS — special number field sieve, появилась в 1988 году в работе Полларда [5] в приложении к факторизации седьмого числа Ферма F_7). Обозначим функцию, выражающую субэкспоненциальную сложность работы алгоритма, следующим образом:

$$L_n[\gamma; c] = e^{(c+o(1))(\log n)^\gamma (\log\log n)^{1-\gamma}},$$

где $0 < \gamma < 1$ и $c = const$, $c > 0$. В этих обозначениях эвристическая оценка сложности усовершенствованного алгоритма квадратичного решета (QNS — quadratic

number sieve) запишется как $L_n[\frac{1}{2};1]$. Эвристическая же оценка сложности алгоритма решета числового поля (GNFS — general number field sieve) составляет $L_n[\frac{1}{3};c]$ при некоторой постоянной c. Т.о. асимптотически GNFS быстрее QNS ($\frac{1}{3}$ вместо $\frac{1}{2}$). Однако вследствие того, что в GNFS $c > 1$, на практике QNS начинает уступать GNFS только на числах $n > 10^{110}$. GNFS является на сегодняшний день асимптотически самым быстрым известным алгоритмом факторизации чисел для классического компьютера.

В противоположность этим субэкспоненциальным (но все же не полиномиальным) классическим алгоритмам предложенный Питером Шором в 1994 году квантовый алгоритм факторизации чисел [9] требует только

$$O((\log n)^2 \log\log n \log\log\log n)$$

шагов на квантовом компьютере, т.е. задача факторизации решается на квантовом компьютере за полиномиальное время. Данный алгоритм требует также полиномиального по $\log n$ времени последующей обработки на классическом компьютере (поиск приближения дроби через цепные дроби, алгоритм Евклида нахождения НОД двух чисел), что необходимо для получения конкретного простого сомножителя факторизуемого числа n из выхода квантового алгоритма. В принципе эти вычисления могут быть произведены и на квантовом компьютере, и их предлагается делать на классическом компьютере только затем, чтобы не строить квантовую вентильную схему для указанных задач, возникающих на стадии последующей обработки.

Далее мы переформулируем задачу в более удобном для решения на квантовом компьютере виде, опишем квантовый алгоритм Шора для факторизации чисел и покажем некоторые вероятностные характеристики выхода этого алгоритма.

2.1 Переформулировка задачи факторизации

Для поиска нетривиального делителя числа n предлагается найти мультипликативный порядок некоторого случайного элемента x кольца Z/nZ, т.е. наименьшее натуральное r такое, что $x^r \equiv 1\ (mod\ n)$. Тогда в предположении, что r — четное, получим:

$$x^r - 1 \equiv 0\ (mod\ n),$$

$$(x^{r/2} - 1)(x^{r/2} + 1) \equiv 0\ (mod\ n),$$

$(x^{r/2} - 1) \neq 0\ (mod\ n)$, т.к. r — порядок x. Т.о., если x — ненулевой и неединичный элемент Z/nZ, $x^{r/2} \neq -1\ (mod\ n)$ и r — четное, то $gcd(x^{r/2} - 1, n)$ есть нетривиальный делитель n.

Можно доказать следующее утверждение: при применении данной процедуры к случайному и равновероятному $x\ (mod\ n)$ нетривиальный делитель n получается с вероятностью, не меньшей $1 - \frac{1}{2^{k-1}}$, где k — количество различных нечетных простых делителей n, если n — нечетное, $k > 1$, т.е. n не есть степень простого числа. Доказательство основывается на китайской теореме об остатках и основной теореме арифметики. В том случае, если n есть степень простого числа, можно воспользоваться известными эффективными алгоритмами для поиска этого простого числа (см. [15]).

2.2 Описание алгоритма

Опишем теперь алгоритм для поиска мультипликативного порядка r элемента $x\ (mod\ n)$. Этот алгоритм использует два квантовых регистра, в которых содержатся целые числа, представленные в двоичном виде. Для работы алгоритма будет необходимо также некоторое количество дополнительного рабочего пространства, содержимое которого устанавливается в 0 после каждой подпроцедуры алгоритма. При описании алгоритма это дополнительное рабочее пространство фигурировать не будет.

Пусть n — число для факторизации, а q — некоторое число из промежутка $[n^2; 2n^2)$, являющееся степенью двойки, т.е. $q = 2^s$ для некоторого натурального s. Пусть также x — случайный элемент Z/nZ, порядок которого и будет определяться описываемым ниже алгоритмом. Параметры n, q и x не входят в описание состояний квантового регистра, т.к. в процессе работы алгоритма они не меняются и могут быть встроенными в структуру массива квантовых вентилей. Более строго, для работы алгоритма необходимо два регистра: один, соответствующий длине в битах двоичного представления числа n^2 (первый регистр), другой — числа n (второй регистр).

Шаг 0(подготовка). Привести все кубиты квантовых регистров в нулевое состояние:

$$\psi_0 = |0\rangle|0\rangle.$$

Шаг 1(вентиль Адамара). Применить вентиль Адамара к каждому кубиту первого регистра:

$$\psi_1 = \frac{1}{q^{1/2}} \sum_{a=0}^{q-1} |a\rangle|0\rangle,$$

что переводит регистр в равновероятную суперпозицию всех возможных состояний, т.е. каждый кубит первого регистра находится в состоянии, описываемом выражением:

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

Шаг 2(возведение в степень). Вычислить значение $x^a\ (mod\ n)$ во втором регистре, беря показатель степени из первого регистра. Это преобразование оставит квантовый компьютер в следующем состоянии:

$$\psi_2 = \frac{1}{q^{1/2}} \sum_{a=0}^{q-1} |a\rangle|x^a\ (mod\ n)\rangle.$$

Шаг 3(квантовое преобразование Фурье). Выполнить квантовое преобразование Фурье на первом регистре. $|a\rangle$ в первом регистре отобразится в:

$$\frac{1}{q^{1/2}} \sum_{c=0}^{q-1} exp(2\pi iac/q)|c\rangle.$$

Тогда общее состояние квантового компьютера запишется как:

$$\psi_3 = \frac{1}{q} \sum_{a=0}^{q-1} \sum_{c=0}^{q-1} exp(2\pi iac/q)|c\rangle|x^a\ (mod\ n)\rangle.$$

Шаг 4(наблюдение состояния). Измерить состояние регистров квантового компьютера:

$$|c\rangle|x^k \ (mod\ n)\rangle,$$

где можно принять $0 \leq k < r$, и завершить алгоритм.

Далее происходит интерпретация результатов измерения и выделение из результатов мультипликативного порядка r элемента x, что описывается в следующем подразделе.

2.3 Вероятностные свойства и интерпретация выхода

Покажем, как оценивается вероятность наблюдать конкретное состояние

$$|c\rangle|x^k \ (mod\ n)\rangle$$

при измерении на последнем шаге работы алгоритма. Суммируя по всем возможным способам достижения указанного состояния при измерении, получим следующую вероятность:

$$\Big|\frac{1}{q}\sum_{a:x^a\equiv x^k} exp(2\pi iac/q)\Big|^2,$$

где сумма берется по всем таким a, $0 \leq a < q$, что $x^a \equiv x^k \ (mod\ n)$. Поскольку мультипликативный порядок x есть r, то эта сумма берется по всем a, удовлетворяющим $a \equiv k \ (mod\ r)$. Полагая $a = k + br$, запишем вероятность:

$$\Big|\frac{1}{q}\sum_{b=0}^{\lfloor (q-k-1)/r\rfloor} exp(2\pi i(br+k)c/q)\Big|^2.$$

Т.к. b пробегает значения от 0 до $\lfloor (q-k-1)/r\rfloor$, то

$$a_{max} = br + k = \lfloor (q-k-1)/r\rfloor r + k = (q-k-1) + k = q-1,$$

т.е. a пробегает все возможные значения из промежутка от 0 до $q-1$. Вынесем константу из-под знака суммы и возведем ее в квадрат:

$$\Big|exp(2\pi ikc/q)\frac{1}{q}\sum_{b=0}^{\lfloor (q-k-1)/r\rfloor} exp(2\pi ibrc/q)\Big|^2 =$$

$$= \Big|exp(2\pi ikc/q)\Big|^2\Big|\frac{1}{q}\sum_{b=0}^{\lfloor (q-k-1)/r\rfloor} exp(2\pi ibrc/q)\Big|^2 =$$

$$= \Big|\frac{1}{q}\sum_{b=0}^{\lfloor (q-k-1)/r\rfloor} exp(2\pi ibrc/q)\Big|^2.$$

В данной формуле можно интерпретировать число rc как вычет $\{rc\}_q$, конгруэнтный $rc \ (mod\ q)$, в промежутке $(-q/2; q/2]$. Переходя от суммы к интегралу (см. [10]), можно найти вероятность получения при измерениях значения

$$|c\rangle|x^k \ (mod\ n)\rangle$$

в квантовых регистрах. Для достаточно больших n эта вероятность равна:

$$Pr\{|c\rangle|x^k \ (mod\ n)\rangle : \frac{-r}{2} \leq \{rc\}_q \leq \frac{r}{2}\} \geq \frac{1}{3r^2}.$$

Т.е. предполагается, что существует такое d, что

$$\frac{-r}{2} \leq rc - dq \leq \frac{r}{2}.$$

Переписывая это неравенство в модульной форме и деля на rq, получим:

$$|rc - dq| \leq \frac{r}{2}, \quad \left|\frac{c}{q} - \frac{d}{r}\right| \leq \frac{1}{2q}.$$

Т.к. $q > n^2$, то существует не более одной дроби $\frac{d}{r}$, $r < n$, удовлетворяющей последнему неравенству. В этом выражении известными являются c и q. Требуется найти r. d — параметр. При этом $\frac{c}{q}$ и $\frac{d}{r}$ достаточно близки друг к другу. Дробь $\frac{d}{r}$ может быть найдена по $\frac{c}{q}$ за полиномиальное время (т.к. используется лишь деление) приближением $\frac{c}{q}$ через цепные дроби. В ряде приближений $\frac{c}{q}$ будем рассматривать только те дроби, у которых знаменатель меньше n, т.к. нам необходимо найти приближение со знаменателем $r < n$, которое заранее неизвестно. Среди этих дробей ближайшая и даст нам r в знаменателе, если $(d, r) = 1$.

Подсчитаем количество состояний $|c\rangle|x^k \ (mod\ n)\rangle$, которые позволяют найти r описанным способом. При подсчете учитываем следующие факторы:

Количество различных x^k. Т.к. r — порядок x, то существует r различных x^k.

$Pr\{|c\rangle|x^k \ (mod\ n)\rangle : \frac{-r}{2} \leq \{rc\}_q \leq \frac{r}{2}\}$. Эта вероятность оценивается как $\geq \frac{1}{3r^2}$.

Количество $d \in Z/nZ : (d, r) = 1$, $d < r$. Таких вычетов по модулю r есть $\varphi(r)$, где φ — функция Эйлера.

Тогда существует $r\varphi(r)$ различных состояний квантового компьютера вида

$$|c\rangle|x^k \ (mod\ n)\rangle,$$

которые обеспечили бы поиск r по описанной схеме. Но каждое из таких состояний встречается с вероятностью $\geq 1/3r^2$. Поэтому вероятность получить при измерении состояния регистров значения со свойствами, которые позволяют найти r, есть

$$r\varphi(r)\frac{1}{3r^2} = \frac{\varphi(r)}{3r}.$$

Существует теорема, позволяющая оценить значение этой вероятности снизу:

$$\frac{\varphi(r)}{3r} > \frac{\delta}{\log\log r},$$

где δ — некоторая константа. Повторяя алгоритм $O(\log\log r) = O(\log\log n)$ раз, получим достаточно высокую вероятность успеха (см. подраздел 2.5 для уточнения этого утверждения).

2.4 Квантовое возведение в степень

Здесь мы рассмотрим подробнее, как выполняется второй шаг квантового алгоритма факторизации, являющийся его наиболее трудоёмкой частью. На входе шага квантовые регистры находятся в состоянии $|a\rangle|0\rangle$, которое необходимо преобразовать в $|a\rangle|x^a \pmod n\rangle$ на выходе, где a, x и n — l-битные числа, причём x и n являются параметрами массива вентилей.

Классическим алгоритмом возведения в степень по модулю n является бинарный алгоритм, который заключается в вычислении степеней x^{2^i} для $i \leq \log_2 a$ путём последовательного возведения в квадрат по модулю n и последующем перемножении подмножества полученных степеней по модулю n для получения результата. При работе с l-битными числами бинарный алгоритм требует $O(l)$ возведений в квадрат и умножений по модулю. Существуют более эффективные общие (работающие по произвольному основанию и с произвольным показателем степени) алгоритмы возведения в степень (например, возведение в степень с использованием окна фиксированной длины или скользящего окна), улучшающие константу, но требующие также $O(l)$ возведений в квадрат и умножений. Если умножения в бинарном алгоритме выполняются по алгоритму Шонхаге-Штрассена [15], асимптотическим самому быстрому среди известных и имеющему сложность $O(l \log l \log\log l)$, общая сложность возведения в степень составляет $O(l^2 \log l \log\log l)$. Однако алгоритм Шонхаге-Штрассена хорош при построении массивов вентилей для длинных чисел. Для небольших значений l самые быстрые класические массивы вентилей построены по обычному "школьному" алгоритму умножения в столбик, имеющему сложность $O(l^2)$, что приводит к общей сложности $O(l^3)$.

Для реализации на квантовых вентилях необходимо сделать вычисление по алгоритму бинарного возведения в степень обратимым, поскольку все физические преобразования квантовой системы обратимы и элементарные квантовые вентили описываются унитарными матрицами. Любое необратимое вычисление можно выполнить с использование набора одно- и двухбитных вентилей; для выполнения обратимых преобразований, отличных от линейных булевых операций, требуется хотя бы один обратимый трёхбитный вентиль. Как следствие, превращение вычисления в обратимое требует затрат некоторого количества дополнительных бит [11]. В [10] приводится общий метод превращения произвольного вычисления, имеющего полиномиальную сложность, в обратимое, и оценка возникающих в связи с этим дополнительных затрат времени и памяти. Здесь мы приведём описанный в [10] непосредственный способ построения обратимого массива вентилей для возведения в степень по модулю с использованием школьного алгоритма умножения.

Основной цикл алгоритма описывается следующим псевдокодом, преобразующим пару значений $(a, 1)$ в $(a, x^a \pmod n)$.

```
power := 1
for i = 0 to l-1
    if (a_i == 1) then
        power := power*x^(2^i) (mod n)
    endif
endfor
```

Здесь a_i обозначает i-тый бит a. Степени x^{2^i} могут быть предвычислены классическим способом и встроены в структуру массива квантовых вентилей. Для вы-

полнения четвёртой строки псевдокода основного цикла необходима процедура, преобразующая b на входе в $bc \pmod n$ на выходе. Это вычисление может быть обратимым, если $(c, n) = 1$, и такого условия достаточно для алгоритма факторизации. Вычисление состоит из двух шагов. Первый шаг принимает b на входе и выдаёт пару значений $(b, bc \pmod n)$ на выходе, выполняя умножение путём последовательного сложения по модулю n, чему соответствует следующий псевдокод.

```
result := 0
for i = 0 to l-1
    if (b_i == 1) then
        result := result + (2^i)c (mod n)
    endif
endfor
```

Поскольку значения c являются параметрами массива вентилей, $2^i c$ могут быть классически предвычислены и встроены в массив. Второй шаг преобразует выход первого шага в $bc \pmod n$ так, чтобы вычисление было обратимым. Поскольку $(c, n) = 1$, то существует c^{-1}, $cc^{-1} \equiv 1 \pmod n$. Следовательно, с помощью умножения на c^{-1} можно выполнить обратимое преобразование из $bc \pmod n$ в $(bc \pmod n, bcc^{-1} \pmod n) = (bc \pmod n, b)$, и применяя его в обратном порядке к выходу первого шага, обнулить b. Псевдокод для второго шага выглядит следующим образом.

```
for i = 0 to l-1
    if (result_i == 1) then
        b := b - (2^i)c^(-1) (mod n)
    endif
endfor
```

Приведённый алгоритм имеет сложность $O(l^3)$, использует $O(l^2)$ квантовых вентилей и сделан обратимым ценой дополнительных $O(l)$ кубит. Возведение в степень по модулю с выполнением умножения по алгоритму Шонхаге-Штрассена вычисляется за время $O(l^2 \log l \log\log l)$ и использует $O(l \log l \log\log l)$ квантовых вентилей, но делается обратимым с использованием $O(l \log l \log\log l)$ кубит.

2.5 Сложность алгоритма

Для реализации описанного алгоритма Шора необходимо произвести 2 основные квантовые операции:

- возведение в степень,
- квантовое дискретное преобразование Фурье.

Пусть l — количество бит в двоичном представлении факторизуемого числа n. Асимптотически лучший алгоритм умножения для массивов вентилей — это алгоритм Шонхаге-Штрассена (см., например, [15]). Его квантовая версия позволяет построить массив квантовых вентилей для возведения в степень по модулю объема $O(l \log l \log\log l)$, решающий задачу за время $O(l^2 \log l \log\log l)$. Элементарный же школьный метод умножения "столбиком" сложности $O(l^2)$ ведет к массиву квантовых вентилей объема $O(l)$, решающему задачу за $O(l^3)$ шагов.

Квантовое дискретное преобразование Фурье в соответствии с квантовой версией классической идеи Кули-Тьюки быстрого преобразования Фурье требует $O(l^2)$ квантовых вентилей (или $O(l \log l)$, если вычисляется приближённое QFT). Шор показал, что его алгоритм решает задачу факторизации с вероятностью $1 - \epsilon$ за N прогонов базового алгоритма:

$$N \geq \frac{2 \log 1/\epsilon}{\alpha\beta} l^2,$$

где α и β — независимые константы относительно n. В работе [4] оценка N уточняется для случая двух простых сомножителей. Если $n = pq$, а p и q просты, то

$$N \geq \frac{2 \log 1/\epsilon}{\alpha(1 - \frac{1}{3}\frac{2+2^{2\tau'}}{2^{\tau_p+\tau_q}})} l,$$

где $p - 1 = 2^{\tau_p}\sigma_p$, $q - 1 = 2^{\tau_q}\sigma_q$ и $\tau' = min(\tau_p, \tau_q)$, а σ_p и σ_q суть такие нечетные числа, что $\tau_p, \tau_q \geq 1$. Исходя из указанных сложностей возведения в степень и преобразования Фурье, квантовое время работы базового алгоритма факторизации Шора (одной итерации общего алгоритма) составляет $O(l^2 \log l \log\log l)$.

2.6 Квантовая сложность факторизации в сравнении с ECDLP и переборной задачей

Оценим объем квантовых ресурсов, необходимых для решения некоторых асимметричных криптографических задач при помощи производных алгоритма Шора, при различных параметрах этих задач, и сравним их со сложностью решения переборной задачи при поиске ключа симметричной криптосистемы. Например, для успешного (за полиномиальное квантовое время) криптоанализа криптосистемы RSA, длина используемого модуля которой составляет 2048 бит, с использованием описанного квантового алгоритма факторизации Шора, необходимо построить квантовый компьютер, состоящий из 2-х квантовых регистров, общей длиной около 4096 кубит. При этом требуется около $34 \cdot 10^9$ квантовых операций. Для решения же эквивалентной ей в классическом смысле задачи ECDLP требуется 1300 кубит и $4 \cdot 10^9$ квантовых операций. В таблице 1 приведены дальнейшие сравнительные данные, отражающие объем квантовых ресурсов, необходимых для решения задачи факторизации и дискретного логарифма в группе точек на эллиптической кривой. Данная таблица, основанная на данных из [7], взята из работы [8] (в части квантовых вычислений) и дополнена собственными данными по классической сложности, рассчитанными по формуле $\sqrt{\pi 2^n}$, где n - длина в битах двоичного представления порядка группы точек соответствующей эллиптической кривой.

В этой таблице принято следующее значение $f(n)$:

$$f(n) = 5n + 8\sqrt{n} + 2\log_2 n + \epsilon,$$

где $\epsilon = 10$. Количество кубитов и квантовых шагов для факторизации соответствует модификации схемы Борегарда (Beauregard) [2] для алгоритма Шора. В каждой строке таблицы приведены длины асимметричного ключа, определяющие сложность задачи факторизации и дискретного логарифмирования в группе точек на эллиптической кривой, которые требуют приблизительно равного количества операций на классическом компьютере. Эта сложность приведена в последней

Таблица 1: Ресурсы для квантового решения задач факторизации и ECDLP

Факторизация			ECDLP			
n	число кубитов	кв. время	n	число кубитов	кв. время	кл. время
	$2n$	$4n^3$		$f(n)$	$360n^3$	
512	1024	$0{,}54{\cdot}10^9$	110	700	$0{,}5{\cdot}10^9$	$6{,}39{\cdot}10^{16}$
1024	2048	$4{,}3{\cdot}10^9$	163	1000	$1{,}6{\cdot}10^9$	$3{,}03{\cdot}10^{24}$
2048	4096	$34{\cdot}10^9$	224	1300	$4{,}0{\cdot}10^9$	$9{,}20{\cdot}10^{33}$
3072	6114	$120{\cdot}10^9$	256	1500	$6{,}0{\cdot}10^9$	$6{,}03{\cdot}10^{38}$
15360	30720	$1{,}5{\cdot}10^{13}$	512	2800	$50{\cdot}10^9$	$2{,}05{\cdot}10^{77}$

строке таблицы. Из этого видно, что для эквивалентных в классическом смысле задач факторизации и ECDLP имеет место утверждение: квантовое решение задачи ECDLP требует меньше ресурсов (как кубит, так и квантового времени), чем решение задачи факторизации; разрыв объемов требуемых ресурсов растет с увеличением классической сложности.

Однако основным препятствием при построении квантового компьютера, способного решать реальные асимметричные криптоаналитические задачи, является сложность создания квантового регистра достаточно большой размерности и приемлемого качества. Отсюда следует естественный метод защиты классических методов от угрозы квантовых алгоритмов: в предположении, что на данный момент технически возможно построить квантовый регистр длины k кубит, предлагается использовать асимметричные алгоритмы, базирующиеся на DLP или факторизации, с ключом длины $> k/2$ бит, а базирующиеся на ECDLP — с ключом длины $> k/5$ бит. Этот параметр (длина квантового регистра) является ключевым, т.к. определяет принципиальную возможность решения задачи за квантовое полиномиальное время. Для получения более полной картины возможностей квантового криптоанализа указанных асимметричных алгоритмов следовало бы также рассмотреть возможность разделения общей задачи на несколько подзадач (возможно, частично решаемых на классическом компьютере за достаточно короткое время), требующих меньшего количества кубитов и решаемых за полиномиальное квантовое время.

В таблице 2 можно найти соответствующие классическим сложностям из таблицы 1 длины k ключа симметричной криптосистемы, необходимое для решения задачи алгоритмом Гровера число кубитов, а также число квантовых шагов. При этом принимается, что вектор булевых функций, соответствующий симметричному криптопреобразованию, вычисляется за один квантовый шаг, что в общем случае не соответствует действительности. Это допущение соответствует упрощению задачи поиска ключа и ведет к квантовой сложности, которая может быть рассмотрена как нижняя (не всегда достижимая) граница стойкости. Классическая сложность, приведенная в таблице, именуется средней, поскольку соответствует среднему времени поиска ключа на классическом компьютере.

Если квантовый шаг будет когда-либо сопоставим по времени с классическим и если предположить наличие квантового регистра длины k кубит, где k — длина симметричного ключа, то достаточно будет увеличить длину ключа симметричного криптоалгоритма в 2 раза для получения эквивалентной сложности решения

Таблица 2: Ресурсы для квантового решения задачи поиска ключа симметричной криптосистемы

k	число кубитов	кв. время	ср. кл. время
	k	$(\pi/4)\sqrt{2^k}$	
57	57	$2{,}8{\cdot}10^{8}$	$6{,}39{\cdot}10^{16}$
82	82	$1{,}93{\cdot}10^{12}$	$3{,}03{\cdot}10^{24}$
113	113	$1{,}06{\cdot}10^{17}$	$9{,}20{\cdot}10^{33}$
129	129	$2{,}76{\cdot}10^{19}$	$6{,}03{\cdot}10^{38}$
258	258	$5{,}03{\cdot}10^{38}$	$2{,}05{\cdot}10^{77}$

переборной задачи. При этом абсолютно стойкий по Шеннону шифр остается абсолютно стойким, т.к. не решается перебором и на классическом компьютере. В случае симметричных шифров также представляется возможным использовать более короткие квантовые регистры для решения переборных задач, возникающих в процессе криптоанализа вследствие, например, декомпозиции исходной задачи, если такое позволяет атакуемый криптоалгоритм. Однако принципиально это ничего не меняет.

Заключение

Таким образом, следует отметить, что квантовое ускорение существенно для наиболее распространенных классических криптографических асимметричных алгоритмов, основанных на факторизации натуральных чисел и дискретном логарифмировании в конечных группах различной математической природы. Квантовый компьютер переводит указанные задачи из неполиномиального класса сложности в полиномиальный класс. Возможность создания квантового компьютера является серьезной угрозой современной классической асимметричной криптографии.

При рассмотрении классических симметричных криптографических систем и алгоритма Гровера можно констатировать, что переборная задача не получает экспоненциального квантового ускорения и ее квантовое решение не представляет принципиальной угрозы для симметричных шифров.

Коренной вопрос стойкости современных асимметричных криптосистем состоит в том, как скоро и насколько качественно удастся построить достаточно длинные (на сегодняшний день — от нескольких сотен кубит) квантовые регистры. Это является крайне сложной технической задачей, но исключать возможность технического прорыва в этом направлении все же представляется не совсем разумным. С появлением идеи квантового компьютера и алгоритма Шора (в различных модификациях) произошло качественное изменение в вопросе о решаемости/нерешаемости за разумное время задач, лежащих в основе классических асимметричных криптосистем: эти задачи уже заключаются не в поиске полиномиального алгоритма, а в отыскании возможности инженерной реализации квантового регистра и соответствующих квантовых вентилей.

Список литературы

[1] *Numerical Recipes in C: The Art of Scientific Computing.* www.nr.com, 2002.

[2] Stephane Beauregard. Circuit for Shor's algorithm using 2n+3 qubits. *arXiv.quant-ph/0205095 v3*, 2003.

[3] D. Coppersmith. An approximate fourier transform useful in quantum factoring. *IBM Research Report RC 19642*, 1994.

[4] K. Kuriyama, S. Sano, and S. Furuichi. A precise estimation of the computational complexity in Shor's factoring algorithm. *arXiv.quant-ph/0406145 v1*, 2003.

[5] A.K. Lenstra and Jr. (Eds.) H.W. Lenstra. *The developement of the number field sieve.* Springer-Verlag, 1993.

[6] A.K. Lenstra and E.R. Verheul. Selecting cryptographic key sizes. 1999.

[7] NIST. Revised draft of key management guideline. *Second Key Management Workshop, http://csrc.nist.gov/encryption/kms/*, 2001.

[8] John Proos and Christof Zalka. Shor's discrete logarithm quantum algorithm for elliptic curves. *arXiv.quant-ph/0301141 v2*, 2004.

[9] Peter W. Shor. Algorithms for quantum computer: Discrete logarithms and factoring. 1994.

[10] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *arXiv.quant-ph/9508027 v2*, 1996.

[11] Peter W. Shor. Introduction to quantum algorithms. *arXiv.quant-ph/0005003 v2*, 2001.

[12] I.V. Volovich. Quantum computing and Shor's factoring algorithm. *arXiv.quant-ph/0109004 v1*, 2001.

[13] О.Н. Василенко. *Теоретико-числовые алгоритмы в криптографии.* М.: МЦНМО, 2003.

[14] Лов К. Гровер. Квантовая механика помогает найти иголку в стоге сена. *Phys. Rev. Lett., 79(2), p. 325-328*, 1997.

[15] Дональд Эрвин Кнут. *Искусство программирования, том 2. Получисленные алгоритмы,3-е изд.* Москва, Издательский дом "Вильямс 2000.

[16] А.С. Холево. *Введение в квантовую теорию информации.* МЦНМО, 2002.