Письма в ЖЭТФ, том 82, вып. 12, с. 868 – 873

О каскадном методе коррекции ошибок в первичных ключах в квантовой криптографии, сохраняющем конфиденциальность

А. В. Тимофеев[+], *С. Н. Молотков*[+*]

[+]*Факультет вычислительной математики и кибернетики, МГУ им. М. В. Ломоносова 119899 Москва, Россия*

[*]*Институт физики твердого тела РАН, 142432 Черноголовка, Московская обл., Россия*

Поступила в редакцию 27 октября 2005 г.

Предложен общий метод сохранения конфиденциальности при коррекции ошибок в первичных ключах в квантовой криптографии через открытый канал связи. На примере каскадной процедуры исправления ошибок описан метод устранения информации, выданной в открытый канал связи в результате "чистки" ключа. Найден критический процент ошибок для квантового криптографического протокола BB84, до которой каскадный метод коррекции ошибок с выбрасыванием гарантирует секретность финального ключа. Предложенный способ устранения информации, выданной в открытый канал связи, является достаточно общим и может быть использован для многих протоколов "чистки" первичных ключей в квантовой криптографии.

PACS: 03.65.Bz, 03.67.Db

Квантовая криптография – система распределения секретных ключей между пространственно удаленными пользователями – позволяет передавать ключи по открытому квантовому каналу связи. Любой квантовый криптографический протокол генерации секретных ключей включает в себя три стадии [1, 2]. Первая стадия – стадия генерации первичных ключей – состоит в передаче квантовому каналу связи квантовых состояний и проведению измерений на приемном конце по определенному протоколу. Результатом измерений является строка битов, которая, вообще говоря, отличается от битовой последовательности на передающем конце. После получения результатов измерений производится обмен информацией через открытый классический канал связи с целью, например, как это имеет место для наиболее распространенного протокола BB84 [1], согласования базисов и оценки вероятности ошибок на приемном конце. При этом часть битовой последовательности раскрывается, и затем раскрытые позиции отбрасываются. В результате первой стадии протокола у легитимных пользователей (именуемых обычно Алисой и Бобом) возникает первичный ключ.

Любая система квантовой криптографии гарантирует секретность ключей, если процент ошибок на приемном конце не превышает некоторой критической величины [3 – 5]. В противном случае протокол прерывается.

В конце первой стадии подслушиватель (обычно Ева) уже *может* иметь[1)] некоторую информацию о первичном ключе, которую он может получить из квантового канала связи во время передачи квантовых состояний. Последующие обмены информацией легитимных пользователей через открытый классической канал связи не могут увеличить информацию подслушивателя о первичном ключе, поскольку раскрытые позиции отбрасываются.

Вторая стадия протокола заключается в коррекции ошибок посредством обмена классической информацией через открытый классический канал связи. Исправление ошибок на приемном конце, по своей сути, является классической процедурой. Принципиальное отличие этой процедуры от обычных методов коррекции ошибок в классической теории информации состоит в том, что она проводится между пространственно удаленными пользователями, и вся вспомогательная информация, передаваемая по открытому каналу связи, считается известной подслушивателю. Главное требование к процедуре коррекции ошибок состоит в сохранении конфиденциальности. Иначе говоря, в результате исправления ошибок

[1)]В данном контексте *"может"* означает следующее. Вообще говоря, подслушиватель может получать информацию о финальном ключе на самой последней стадии, когда закончатся все вспомогательные обмены классической информацией между легитимными пользователями, сохраняя, например, свои квантовые состояния в квантовой памяти.

подслушиватель не должен увеличить свою информацию об "очищенном" ключе, которую он имел до этой стадии. При этом часть битов в первичной последовательности Алисе и Бобу приходится отбрасывать. Второе требование относится к эффективности процедуры, и состоит в том, чтобы "очищенный" ключ оставался как можно большей длины.

Строго говоря, величина критической ошибки, до которой протокол обеспечивает секретность ключей, эффективность процедуры коррекции ошибок в первичных ключах, длина финального ключа тесно взаимосвязаны между собой. Например, максимально допустимая величина критической ошибки 11% для протокола BB84 достигается при коррекции ошибок при помощи случайных кодов (шенноновский предел [6]), которые не являются конструктивно реализуемыми, поскольку требуют экспоненциально большого набора кодовых слов по длине битовой последовательности. На случайных кодовых словах достигается максимум минимального кодового расстояния. В этом случае, при последовательности достаточно большой длины (формально при $n \to \infty$) и вероятности ошибки, меньше критической, вся длина битовой последовательности после исправления ошибок может быть принята как финальный секретный ключ. Если для коррекции ошибок используются коды, на которых достигается граница скорости Варшамова–Гильберта [7], то допустимый процент ошибок не превышает 7.5% [3–5]. В этом случае после исправления ошибок вся последовательность также может быть принята как финальный секретный ключ.

Использование конструктивных кодов при коррекции ошибок является исправлением ошибок "вперед", в том смысле, что в зависимости от наблюдаемого процента ошибок выбирается набор кодовых слов, которые открыто анонсируются. Исправление ошибок на приемном конце происходит по вычисленному синдрому. При этом критический процент ошибок, до которого протокол гарантирует секретность *целой* исправленной битовой последовательности, зависит от избыточности кода. При таком подходе критическую величину ошибки приходится для каждого кода вычислять заново.

Практически более удобными оказываются итерационные процедуры исправления ошибок, которые сводятся, в том или ином виде, к вычислению четностей различных подмножеств первичного ключа и двустороннему обмену информацией о них по открытому каналу связи. При этом четности этих подмножеств становятся известны подслушивателю. Если определенным образом выбрасывать некоторые биты, либо в процессе "чистки", либо после ее завершения, подслушиватель не сможет получить дополнительной информации по сравнению с той, которую он имел до процедуры коррекции ошибок.

Третья стадия протокола – усиление секретности или сжатие (privacy amplification) "очищенного" ключа [8]. После коррекции ошибок и отбрасывания части битов у легитимных пользователей остается битовая строка меньшей длины. Информация подслушивателя об этой строке ограничена исходной информацией, которую он мог получить из квантового канала связи. Таким образом, если возможно найти надежную верхнюю границу информации, которую подслушиватель может получить из квантового канала связи, то это бы конструктивно решало проблему извлечения секретного ключа из первичной битовой последовательности. Эта информация может быть уменьшена до экспоненциально малой величины по выбранному параметру секретности путем сжатия (хэширования при помощи однородных универсальных функций второго рода [8, 9] "очищенного" ключа).

Каскадный метод коррекции ошибок (CASCADE). Наиболее простая итерационная процедура коррекции ошибок – это бисективный поиск ошибок, который сводится к разбиению первичного ключа на случайные непересекающиеся блоки и вычислению четностей этих блоков. Четности множеств сравниваются как на приемном, так и на передающем конце через открытый канал. После раскрытия четности какого-либо множества один из случайно выбранных битов отбрасывается. При несовпадении четностей размер блока уменьшается вдвое и процесс повторяется. Поскольку блоки не пересекаются, то выбрасывание битов не представляет труда. Такая процедура сохраняет конфиденциальность – подслушиватель не получает дополнительной информации при "чистке" первичного ключа. Однако такая процедура крайне неэффективна, поскольку остается достаточно мало битов в "очищенном" ключе (например, при вероятности ошибки в 10% в первичном ключе в "очищенном" ключе остается не более 10% от исходной длины).

Наиболее эффективным, в смысле длины "очищенного" ключа, на сегодняшний день, по-видимому, является каскадный метод коррекции ошибок, предложенный в [10]. В исходном варианте каскадного метода биты четности отдельных и, в общем случае, *пересекающихся* подмножеств запоминаются и используются на следующих проходах. В процессе работы метода никакие биты не выбрасываются, поэтому метод не сохраняет конфиденциальность. Оценить информацию, которую получает подслушиватель, когда раскрываются биты четности набора пересекающихся подмножеств, которые возникают на разных проходах, достаточно сложно. До сих пор, насколько нам известно, полный анализ не выполнен.

Сохранить конфиденциальность и эффективность метода можно, если в конце "чистки" первичного ключа отбрасывать некоторое количество битов. Поскольку возникающие на каждом проходе множества битов пересекаются, то процедура выбрасывания не является тривиальной.

В данной работе предлагается простой регулярный способ сохранения конфиденциальности.

Вначале для удобства приведем описание каскадного метода коррекции ошибок без выбрасывания.

Алгоритм CASCADE заключается в нескольких проходах по первичному ключу. На каждом проходе ключ случайным образом разбивается на блоки определенного размера с помощью вспомогательных хэш-функций, и в этих блоках ищутся ошибки по описанному ниже алгоритму. Количество проходов и размер блоков определяются заранее по вероятности ошибки в первичном ключе способом, описанным в работе [10].

Каждый проход состоит из следующих шагов.

1. Вначале ключ разбивается на блоки с помощью хэш-функции: в блок m входят те биты, для номеров которых выполняется равенство $f_k(i) = m$. Функция выбирается из класса универсальных хэш-функций [9].

2. Далее Алиса и Боб вычисляют четности своих блоков и сравнивают их. Если четности какого-либо блока у Алисы и Боба не совпадают, значит в этом блоке есть нечетное число ошибочных битов. В таком блоке Алиса с Бобом могут найти ошибку бисективным поиском: Алиса с Бобом делят этот блок пополам и сравнивают четности первой половины блока. Если они совпадают, то во второй половине осталось нечетное число ошибок, в противном случае нечетное число ошибок имеется в первой части. Половина блока с ошибками снова делится пополам, и так продолжается до тех пор, пока ошибка не будет локализована, тогда Боб исправляет значение ошибочного бита.

3. На проходах, начиная со второго, найденную ошибку можно использовать для нахождения других ошибок. Обозначим номер бита с ошибкой i, а номер прохода k. Бит с номером i принадлежал каким-то блокам на предыдущих проходах. У каждого из этих блоков теперь изменилась четность, так что в них можно найти ошибки тем же методом. Составим из этих блоков множество $\mathcal{K}$, выберем из этого множества один блок и найдем в нем еще одну ошибку. Пусть номер бита с ошибкой j, а множество блоков, содержащих этот бит, – $\mathcal{M}$. Теперь нечетное число ошибок содержат все блоки, которые содержат бит j или i, но не оба вместе. Составим новое множество $\mathcal{K}^* = \mathcal{M}\nabla\mathcal{K}$, равное симметрической разности множеств $\mathcal{K}$ и $\mathcal{M}$. Найдем ошибку в блоке из этого множества. Этот процесс продолжается до тех пор, пока новое множество $\mathcal{K}$ не станет пустым.

Практика показывает, что достаточно четырех проходов по ключу, чтобы вычистить все ошибки в подавляющем большинстве случаев. Чтобы убедиться, что все ошибки исправлены, составим N случайных подмножеств битов ключа и сравним их четности. Вероятность того, что ключ с ошибками пройдет такую проверку, равна 2^{-N}.

Информация, известная подслушивателю. Рассмотрим, какую информацию о ключе получит подслушиватель в результате работы каскада. Как обычно в квантовой криптографии, будем считать, что подслушивателю известны хэш-функции и вообще все параметры работы протокола. Тогда на каждом проходе он узнает биты четности для блоков плюс на каждую найденную ошибку он получает log(размер блока) битов четности подмножеств блока. Блоки на одном проходе не пересекаются, каждое подмножество, полученное в результате поиска ошибки, целиком вложено в какое-то из других подмножеств либо в блок целиком (рис.1)

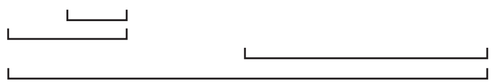

Рис.1

В результате проверки ключа раскрывается еще N битов четности. Итого, вся информация, которой обладает подслушиватель, – это некоторое количество битов четности подмножеств ключа.

Сохранение конфиденциальности. Если есть единственное множество, для которого известен бит четности, от этой информации можно избавиться, удалив из этого множества любой бит (при условии, что значения бита 0 и 1 равновероятны), поскольку разным значениям бита соответствуют разные четности. Предположим, у нас есть два множества, одно из которых вложено в другое (см. рис.2).

Рис.2

Очевидно, знание битов четности этих множеств эквивалентно знанию четностей таких множеств (рис.3).

Рис.3

Теперь допустим, что у нас есть набор пересекающихся множеств M (рис.4).

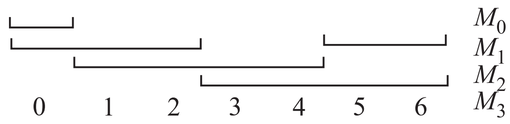

Рис.4

Требуется найти такой набор битов B, чтобы после его удаления исчезла вся информация о четности этих множеств. Это означает, что, перебирая значения битов B, можно получить все возможные значения четностей M. Такой набор существует (например, все биты), найдем минимальный из таких наборов. Изменяя значение одного бита, мы изменяем четность всех множеств, содержащих этот бит. Составим для каждого бита B_i битовый вектор V_i, в котором бит V_i^j равен единице, если бит $B_i \in M_j$:

$$
\begin{aligned}
V_1 &= (1,1,0,0),\\
V_2 &= (0,1,1,0),\\
V_3 &= (0,1,1,0),\\
V_4 &= (0,0,1,1),\\
V_5 &= (0,0,1,1),\\
V_6 &= (0,1,0,1),\\
V_7 &= (0,1,0,1).
\end{aligned}
$$

Изменяя значения нескольких битов $i_1, \ldots, i_k$, мы изменяем четность всех множеств, у которых стоит единица в линейной комбинации $V_{i_1} + \ldots + V_{i_k}$. Следовательно, если один из векторов V_i является линейной комбинацией других, то с помощью бита i невозможно получить новые значения четностей. То есть в качестве набора B можно взять биты, векторы которых образуют базу векторов V.

Найти базу можно, если составить из этих векторов матрицу, и привести ее к ступенчатому виду методом Гаусса, но вначале изменим немного набор множеств M с целью уменьшения количества единиц в матрице. Во-первых, выкинем все повторяющиеся векторы:

$$
\begin{aligned}
V_1 &= (1,1,0,0),\\
V_2 &= (0,1,1,0),\\
V_4 &= (0,0,1,1),\\
V_6 &= (0,1,0,1).
\end{aligned}
$$

Каждый набор множеств, полученный в результате поиска одной ошибки, можно превратить в набор непересекающихся множеств (рис.5).

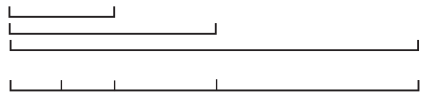

Рис.5

После этой операции все множества, полученные на одном проходе каскада не будут пересекаться и, следовательно, в каждой строке матрицы будет не более четырех битов (не считая N проверочных). Для нашего примера получается (рис.6).

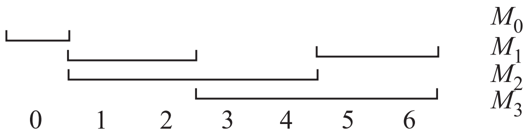

Рис.6

И, соответственно, векторы

$$
\begin{aligned}
V_1 &= (1,0,0,0),\\
V_2 &= (0,1,1,0),\\
V_4 &= (0,0,1,1),\\
V_6 &= (0,1,0,1).
\end{aligned}
$$

Упорядочим множества по возрастанию их мощностей:

$$|M_1| < |M_2| < \ldots < |M_m|,$$

$$
\begin{aligned}
V_1 &= (1,1,0,0),\\
V_2 &= (0,1,1,0),\\
V_4 &= (0,0,1,1),\\
V_6 &= (0,1,0,1).
\end{aligned}
$$

Теперь матрицу можно хранить в разреженном виде, и метод Гаусса потребует намного меньше операций. После метода Гаусса получаем матрицу

$$\begin{aligned} V_1 &= (1,1,0,0), \\ V_2 &= (0,1,1,0), \\ V_4 &= (0,0,1,1), \\ V_6 &= (0,0,0,0). \end{aligned}$$

И значит, достаточно выкинуть три бита с номерами 1, 2 и 4.

Степень сжатия "очищенного" ключа. Степень сжатия "очищенного" ключа зависит от эффективности процедуры исправления ошибок, точнее, от избыточности. Использование случайных кодовых слов (шенноновский предел) позволяет получить максимальное количество бит информации – $nH(Q)$ (n – длина последовательности, $H(Q) = 1+Q\log(Q)+(1-Q)\log(1-Q)$, Q – вероятность ошибки). Соответственно, избыточность случайного кода $n(1-H(Q))$. Шенноновский случайный код обладает максимальным минимальным расстоянием (минимальной избыточностью) по сравнению с другими. Исправление ошибок любыми другими процедурами приводит к большей избыточности. Другими словами, *раскрытая информация в битах* при коррекции ошибок не может быть меньше, чем $n(1-H(Q))$. Чем больше избыточность кода, исправляющего ошибки, тем меньше длина финального ключа, а также тем меньше критическая величина ошибки, до которой гарантируется секретность ключа.

На рис.7 для сравнения приведена избыточность для шенноновских случайных кодов и процедуры CASCADE с выбрасыванием.

Воспользуемся следствием из фундаментальной теоремы об усилении секретности [8]. Если $E:\{0,1\}^{n_{AB}} \rightarrow \{0,1\}^t$ – произвольная функция, описывающая стратегию Евы в том смысле, что для произвольной строки бит длиной n_{AB} Еве известно не более t бит. Здесь n_{AB} – битовая строка у Алисы и Боба после исправления ошибок. Пусть s – параметр секретности ($s < n_{AB} - t$). Далее пусть $G : \{0,1\}^n \rightarrow \{0,1\}^r$ – универсальная однородная функция хэширования второго рода [9, 8], которая сама является случайной величиной. Тогда взаимная информация Евы о секретном ключе $K = G(X)$ не превосходит

$$I(K;GZ) \leq 2^{-s}/\ln 2. \tag{1}$$

Пусть $Z : \{0,1\}^t$ – строка Евы, согласованная со строкой легитимных пользователей X, в том смысле, что эта строка могла произойти из строк X как

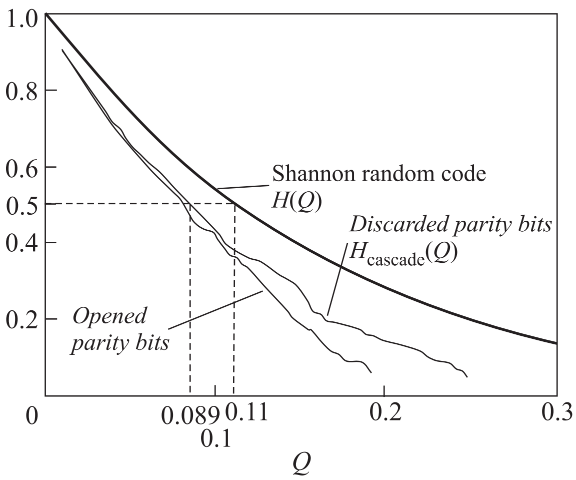

Рис.7. Коррекция ошибок методом CASCADE с выбрасыванием проводилась на последовательности длиной в 10^4 бит

$Z = E(X)$. Наша цель будет состоять в вычислении длины финального ключа r, о котором Ева имеет экспоненциально малую информацию по s. Для этого потребуется знание энтропии Реньи второго рода, которая в свою очередь выражается через вероятность коллизий.

Для квантовой криптографии (протокол BB84) Ева может различить не более $2^{n\frac{\overline{C}(\rho)}{2}}$ строк ($\overline{C}(\rho)$ – классическая пропускная способность квантового канала связи [11]) из общего множества $2^{nH_{\text{CASCADE}}(Q)}$ – числа возможных строк у Алисы и Боба после исправления ошибок процедурой CASCADE. Другими словами, $n_{AB} = nH_{\text{CASCADE}}(Q)$ – длина битовой строки, которая остается после исправления ошибок процедурой CASCADE с выбрасыванием (n – длина строки до исправления ошибок).

Условная вероятность определяется отношением общего числа строк к числу областей декодирования у Евы:

$$P_{X|Z=z} = \frac{2^{n\frac{\overline{C}(\rho)}{2}}}{2^{nH_{\text{CASCADE}}(Q)}} = 2^{-n(H_{\text{CASCADE}}(Q) - \frac{\overline{C}(\rho)}{2})} = a_z. \tag{2}$$

Фактически $1/a_z$ – доля строк таких, что $z = E(X)$. То есть с каждой частичной строкой Евы согласовано множество строк, диктуемое формулой (1). Для вероятности коллизий получаем

$$P_c(X|Z=z) = \sum_{X:\{z=E(X)\}} P^2_{X|Z=z} = \\ = 2^{-n(H_{\text{CASCADE}}(Q) - \frac{\overline{C}(\rho)}{2})} = \frac{1}{a_z} a_z^2. \tag{3}$$

Энтропия Реньи второго рода равна

$$R(X|Z=z) = -\log P_c(X|Z=z) = = n\left(H_{\text{CASCADE}}(Q) - \frac{\overline{C}(\rho)}{2}\right). \quad (4)$$

Согласно теореме об усилении секретности [8], находим

$$H(K|G,Z=z) \geq r - 2^{r-R(X|Z=z)}/\ln 2 > r - 2^r/a_z\ln 2. \quad (5)$$

Для взаимной информации между строками Евы и секретным ключом у Алисы и Боба, с учетом того, что $P_Z(z) = 1/2^{n\overline{C}(\rho)/2} = 2^{-nH_{CASCADE}(Q)}/a_z$, имеем

$$I(K;GZ) = H(K) - H(K|GZ) \leq$$
$$\leq r - \sum_{z\in\{0,1\}^t} P_Z(z)H(K|G,Z=z) \leq$$
$$\leq \sum_{z\in\{0,1\}^t} a_z 2^{-nH_{CASCADE}(Q)} \frac{2^r}{a_z\ln 2} =$$
$$= 2^{-nH_{\text{CASCADE}}(Q)+t+r}/\ln 2 = 2^{-s}/\ln 2. \quad (6)$$

Секретный ключ $K = G(X) \in \{0,1\}^r$, длина которого

$$r = n\left(H_{\text{CASCADE}}(Q) - \frac{\overline{C}(\rho)}{2}\right) - s. \quad (7)$$

Для протокола BB84 $\overline{C}(\rho) = 1$, поэтому "чистка" первичного ключа процедурой CASCADE с выбрасыванием обеспечивает секретность финального ключа, если процент ошибок не превышает $Q_{\text{CASCADE}} \approx \approx 8.9\%$ (Q_{CASCADE} определяется как корень уравнения $H_{\text{CASCADE}}(Q_{\text{CASCADE}}) = \overline{C}(\rho)/2$).

При коррекции ошибок случайными кодовыми словами (шенноновский предел) протокол BB84 секретен до $Q_c \approx 11\%$ ($H(Q_c) = \overline{C}(\rho)/2$).

Описанный метод сохранения конфиденциальности является достаточно общим и может быть использован для других процедур коррекции ошибок.

Работа поддержана Российским фондом фундаментальных исследований (гранты #05-02-17387-а, 02-208306-офи), гос. контрактом ФАНИ 02.435.11.1004. Один из авторов (С.Н.М) благодарит Академию криптографии РФ за поддержку.

1. C. H. Bennett and G. Brassard, Proc. of IEEE Int. Conf. on Comput. Sys. and Sign. Proces., Bangalore, India, December 1984, p. 175.
2. C. H. Bennett, F. Bessette, G. Brassard et al., Journal of Cryptology **5**, 3 (1992).
3. D. Mayers and A. Yao, quant-ph/9802025.
4. E. Biham, M. Boyer, P. O. Boykin et al., quant-ph/9912053.
5. P. W. Shor and J. Preskill, quant-ph/0003004.
6. C. E. Shannon, Bell Syst. Tech. Jour. **27**, 397; 623 (1948).
7. E. J. Mac Williams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Publ. Company, Amsterdam, New York, Oxford, 1977.
8. C. H. Bennett, G. Brassard, C. Crépeau, and U. Maurer, IEEE Transaction on Information Theory **41**, 1915 (1995).
9. J. L. Carter and M. N. Wegman, J. of Computer and System Sciences **18**, 143 (1979).
10. Gilles Brassard and Louis Salvail, Secret-Key Reconciliation by Public Discussion, EUROCRYPT, 1993, p. 410; G. Brassard and L. Salvail, Lecture Notes in Computer Sci. **765**, 410 (1994).
11. А. С. Холево, Проблемы передачи информации **8**, 63 (1972); **15**, 3 (1979); Успехи математических наук **53**, 193 (1998); А. С. Холево, Введение в квантовую теорию информации, серия Современная математическая физика, вып. 5, МЦНМО, Москва, 2002.