Об экспериментальной оценке стойкости метода случайного кодирования к атаке многократного наблюдения частичных кодовых векторов

Ю. О. Газарян, Ю. В. Косолапов*

Южный федеральный университет, Ростов-на-Дону, Россия

*Контактный e-mail: itaim@mail.ru

Рассматривается модель наблюдения частично стертых данных, которая возникает, например, в рамках задачи гарантированного удаления данных на носителях информации. Предполагается, что данные закодированы с помощью метода случайного кодирования смежными классами, для которого ставится задача оценки среднего количества частично стертых кодовых сообщений, по которым наблюдатель сможет восстановить закодированный информационный вектор. С этой целью построены и обоснованы алгоритмы для экспериментальной оценки количества кодовых сообщений в зависимости от модели стирания данных. Для модели случайного и равновероятного стирания фиксированного количества кодовых символов, а также для модели стирания каждого кодового символа с заданной вероятностью представлены результаты вычислений в случаях, когда метод случайного кодирования реализован на основе $[7, 4]$-кода Хэмминга и $[8, 4]$-кода Рида — Маллера.

Ключевые слова: случайное кодирование, случайное стирание, многократное наблюдение.

1. Постановка задачи

Рассмотрим схему информационно-аналитической модели на рис. 1, где два легитимных пользователя (отправитель и получатель) обмениваются сообщениями по каналу передачи данных без помех (главному каналу), а нелегитимный участник (наблюдатель) подслушивает передаваемые данные по каналу со стираниями (каналу наблюдения). Наблюдатель не имеет априорной информации о содержании передаваемых сообщений. Задачей отправителя является кодирование информационных векторов таким образом, чтобы наблюдатель не смог раскрыть содержание закодированных данных в рамках заданной модели стирания кодовых векторов. Часто в качестве метода кодирования используется случайное кодирование, когда каждому информационному вектору ставится в соответствие подмножество кодовых векторов, при этом разным информационным векторам соответствуют непересекающиеся подмножества кодовых векторов [1].

В настоящей работе ставится задача оценки среднего количества частичных кодовых слов, необходимого для снятия неопределенности об информационном слове в рамках

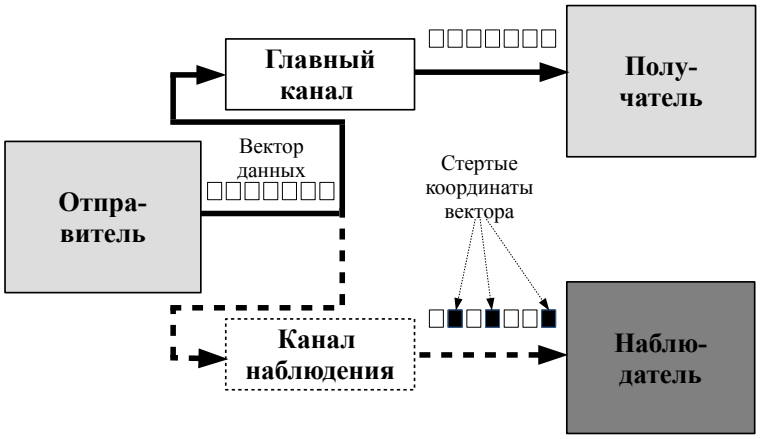

Рис. 1. Схема информационно-аналитической модели: получатель принимает данные от отправителя без помех, а наблюдатель принимает данные со стираниями

случайного стирания кодовых символов. Отметим, что такая постановка задачи характерна, например, для современных систем хранения данных в рамках задачи гарантированного удаления; мотивировка задачи подробно раскрывается в разд. 2. В третьем разделе приводится описание математической модели атаки многократного наблюдения, а в четвертом разделе строится алгоритм для экспериментальной оценки числа $p(\mathcal{M}, \alpha)$ — среднего количества частичных кодовых слов, необходимого для снятия неопределенности о неизвестном информационном векторе с вероятностью α в рамках модели случайного стирания $\mathcal{M}$. Результаты вычислений для бинарных кодов Хэмминга и Рида — Маллера приводятся в пятом разделе. Отметим, что для исследования выбраны две модели стирания символов в кодовых словах: $\mathcal{M}^{U(\rho)}$ — модель равновероятного стирания ρ координат в кодовом векторе, $\mathcal{M}^{\mathrm{EC}(p)}$ — модель стирания с вероятностью стирания p каждого символа кодового слова. Выбор этих моделей стирания мотивирован тем, что они могут быть просто реализованы, например, для стирания данных на носителях информации.

2. Мотивировка поставленной задачи

Поясним, когда возникает рассматриваемая информационно-аналитическая модель при использовании систем хранения, построенных по технологии электрически программируемой памяти — EEPROM (Electrically Erasable Programmable Read-Only Memory), в системах хранения в качестве отправителя и получателя данных выступает владелец носителя, а в качестве наблюдателя — субъект, получивший случайно или преднамеренно несанкционированный доступ к носителю. Положительными особенностями EEPROM-носителей являются их компактность и невысокое энергопотребление при чтении данных. С другой стороны, на операцию стирания ячеек памяти затрачивается существенно больше энергии [2]. Поэтому с учетом того, что перезапись ячеек памяти выполняется путем стирания предыдущих значений ячеек и записи (программирования) новых значений, в целом возрастает время выполнения операций по модификации данных и соответственно повышается количество потребляемой энергии. В связи с этим в теории кодирования возникло направление по разработке методов кодирования

данных таким образом, чтобы имелась возможность как можно дольше использовать ячейки памяти для записи новых данных, не прибегая к стиранию (перепрограммированию) предыдущих значений ячеек [3]. Примером такого кода является простой код, построенный по методу Р. Ривеста и А. Шамира [4].

Информационный бит	Кодовое слово № 1	Кодовое слово № 2
0	00	11
1	10	01

Код Ривеста — Шамира устроен таким образом, что каждому информационному биту ставится в соответствие пара кодовых слов длины два. Если $a(\in \{0;1\})$ — информационный бит, а $\mathbf{c}_a$ — соответствующее кодовое слово № 1, то изменение информационного бита a на значение $b(\in \{0;1\})$ осуществляется изменением некоторых нулевых битов кодового вектора $\mathbf{c}_a$ на единичные биты так, чтобы получился кодовый вектор № 2, соответствующий информационному биту b. Если каждая ячейка памяти может иметь два состояния: "0" — стертая ячейка и "1" — программированная ячейка, то, используя код Ривеста — Шамира, однократное изменение информационных бит можно осуществить только программированием некоторых ячеек без использования дорогой операции стирания. Например, если $a = 1$ и $\mathbf{c}_a = (10)$, то замена бита a значением $b = 0$ выполняется программированием второй ячейки в кодовом слове $\mathbf{c}_a$ так, чтобы получилось кодовое слово $\mathbf{c}_b = (11)$, соответствующее биту b. Отметим, что актуальность таких методов кодирования связана также с тем, что ячейки памяти обладают ограниченным количеством циклов перезаписи: от 10 000 до 100 000 для EEPROM-носителей, построенных по технологии NOR, и от 100 000 до 1 000 000 для носителей с технологией NAND [5].

К наиболее известным способам кодирования можно отнести WOM-коды [4, 6, 7], WAM-коды [8] и WEM-коды [9, 10]. С другой стороны, с точки зрения защиты конфиденциальности данных важным является гарантированное удаление ненужных данных из областей памяти [11]. В частности, как показало исследование [12], эта задача актуальна для EEPROM-носителей. Актуальность этой задачи связана с тем, что в таких носителях на уровне трансляции (FTL — Flash Translation Layer), отвечающем за представление EEPROM-носителя в виде диска, для ускорения операций модификации данных блок с измененными данными записывается в новую область памяти на носителе, а область памяти, содержащая устаревшие данные, помечается как незадействованная. При этом данные из незадействованной области не удаляются. В результате на EEPROM-носителе возникают области памяти, содержащие нестертые, возможно, ценные данные, которые недоступны для чтения стандартными средствами операционной системы, однако которые могут быть прочитаны с помощью недорогих специальных средств.

Требование сокращения количества стираний ячеек, с одной стороны, и гарантированного удаления данных, с другой — привело к появлению кодов, позволяющих кодировать данные в кодовые слова так (не используя секретных ключей), что стирание части кодового слова не позволяет в теоретико-информационном смысле восстановить закодированную информацию [13, 14]. Фактически предложенные в [13, 14] коды позволяют обеспечить защиту конфиденциальности данных в рамках модели, изображенной на рис. 1, и являются реализациями метода случайного кодирования, предложенного А. Вайнером в 1975 г. в работе [15]. В соответствии с [13, 14] данные на носителях сначала кодируются с помощью метода случайного кодирования, а затем символы кодовых

слов кодируются подходящим WOM/WAM/WEM-кодом. Заметим, что метод случайного кодирования активно исследуется в рамках защиты конфиденциальности данных в подслушиваемых каналах связи [15 – 19], и работы [13, 14] являются одними из тех немногих работ (см. также [20]), в которых метод случайного кодирования применяется для защиты данных в системах хранения. Подробный обзор существующих способов случайного кодирования можно найти в [1].

Одним из наиболее известных способов случайного кодирования является кодирование смежными классами линейного $[n, n-k]$-кода C. [17]. Этот способ характеризуется числом $\mu_0 (< n)$ — максимальным количеством кодовых символов, по которым наблюдатель не может получить какую-либо информацию о закодированном информационном векторе длины k в рамках информационно-аналитической модели, изображенной на рис. 1. Другими словами, при необходимости обеспечения совершенной защиты должны быть стерты не менее $n - \mu_0$ символов каждого кодового слова; в этом случае множество претендентов информационных векторов, которое может построить наблюдатель по имеющимся у него μ_0 кодовым символам, будет совпадать с множеством всех информационных векторов. Если же стерто меньше, чем $n - \mu_0$ кодовых символов, то множество претендентов на информационный вектор не совпадает с пространством всех возможных информационных векторов. Согласно [21], в общем случае при перехвате μ координат кодового слова мощность множества претендентов равна q^{k-r}, где q — мощность поля, а число r определяется из неравенств $d_r(C^\perp) \le \mu < d_{r+1}(C^\perp)$, где $(d_1(C^\perp), ..., d_k(C^\perp))$ — весовая иерархия $[n, k]$-кода $C^\perp$, дуального к коду C.

Пусть $N(k)$ — количество ячеек памяти (принимающих два значения), необходимое для хранения k информационных битов, M — количество модификаций данных без использования операции стирания ячеек памяти, $p(\delta)$ — вероятность восстановления информации по *одному* частичному блоку данных при доле стираний δ в этом блоке данных, $0 \le \delta \le 1$.

В таблице приведены сравнительные характеристики двух систем хранения, в которых для сокращения числа перезаписей применяется WOM/WAM/WEM-код, избыточность которого равна λ и который позволяет выполнять m модификаций данных без применения стирания ячеек памяти. Дополнительно в одной из систем для защиты от частичного наблюдения перед применением WOM/WAM/WEM-кода данные кодируются смежными классами $[n, n-k]$-кода C. Здесь предполагается, что единицей стирания данных на носителе является кодовое слово используемого WOM/WAM/WEM-кода. Из таблицы, в частности, видно, что без использования кодирования смежными классами при всех $\delta < 1$ вероятность $p(\delta)$ всегда будет больше вероятности угадывания блока из нулей и единиц длины k (совершенная защита не обеспечивается). В то же время при кодировании смежными классами для δ таких, что $(1-\delta)n < d_1(C^\perp)$, вероятность $p(\delta)$

Характеристики систем хранения с применением и без применения метода кодирования смежными классами

Система	$N(k)$	M	$p(\delta)$
Без использования $[n, n-k]$-кода C	$\dfrac{k}{(1-\lambda)}$	m	$2^{-\delta k}$
С использованием $[n, n-k]$-кода C	$\dfrac{n}{(1-\lambda)}$	m	2^{r-k}, $d_r(C^\perp) \le (1-\delta)n < d_{r+1}(C^\perp)$

будет равна вероятности угадывания, т. е. обеспечивается совершенная защита. Однако, очевидно, что платой за обеспечение совершенной защиты является увеличение избыточности: количество ячеек для хранения k битов данных увеличивается в n/k раз по сравнению с системой, в которой не используется кодирование смежными классами. Так как кодовые слова WOM/WAM/WEM-кода стираются полностью, характеристика M не зависит от представления данных, в частности не зависит от того, используется или нет кодирование смежными классами.

Согласно таблице, если $(1 - \delta n) \geq d_1(C^{\perp})$, то $p(\delta)$ всегда будет больше вероятности угадывания блока из нулей и единиц длины k. В этом случае говорят, что в теоретико-информационном смысле наблюдатель может получить ненулевую информацию о закодированном информационном векторе. В работе [19] рассмотрен способ использования ненулевой информации наблюдателем для *полного* снятия неопределенности о закодированном сообщении — метод многократного наблюдения частично стертых кодовых слов (возможно, разных), соответствующих одному неизвестному информационному вектору. Так, если наблюдателю известен формат данных и он может выделить те части блока закодированных данных, которые соответствуют одному и тому же информационному вектору $\mathbf{s}$ длины k, то имеется возможность сузить множество претендентов.

Атака путем многократного наблюдения характерна для EEPROM-носителей (рис. 2). Например, как показало исследование [12], одни и те же данные, возможно, с некоторыми изменениями могут быть записаны в 16 разных страниц флэш-памяти, только одна из которых содержит данные с последними изменениями и доступна для чтения штатными средствами операционной системы. В этом случае, если перед записью данные кодируются смежными классами, например, с использованием кодов из [13] (с последующим кодированием WOM/WAM/WEM-кодом), а на уровне трансляции EEPROM-носителя незадействованные области памяти стираются не полностью, а частично, то наблюдатель в обход штатного интерфейса операционной системы с большой вероятностью может найти до 16 областей памяти, соответствующих одинаковым информационным векторам, т. е. наблюдатель может провести атаку многократного наблюдения.

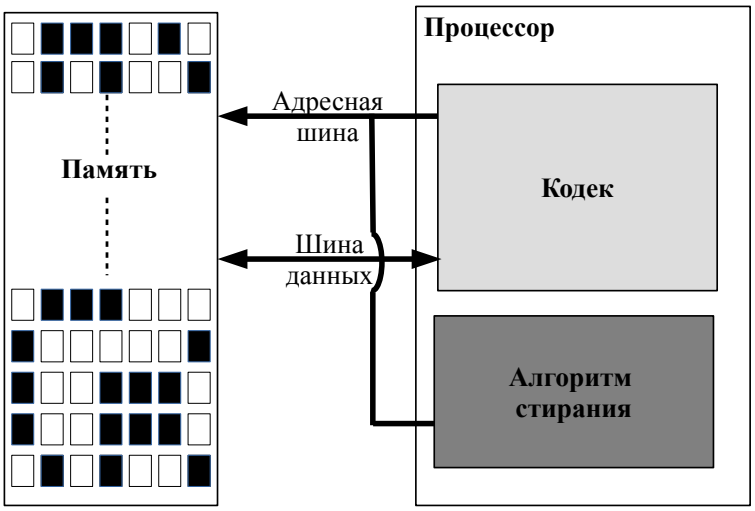

Рис. 2. Схема носителя, во флэш-памяти которого имеются частично стертые кодовые слова (показаны черными прямоугольниками), соответствующие одному информационному вектору

Отметим, что в найденных областях частичные кодовые слова могут быть разными, так как одному информационному вектору соответствует смежный класс кода C, но и в этом случае наблюдатель может сузить множество претендентов. В частности, в [19] показано, что если наблюдатель имеет возможность выбирать номера наблюдаемых координат в кодовых словах, то при кодировании смежными классами $[7, 4]$-кода Хэмминга и при стирании трех символов ($\mu_0 = 4$) наблюдателю достаточно трех кодовых слов для однозначного восстановления закодированного сообщения.

Однако не всегда наблюдатель может выбирать номера наблюдаемых координат. Например, при стирании данных на носителях процесс стирания находится под контролем владельца носителя (или разработчика носителя) и в общем случае может быть рандомизированным. При этом наблюдатель не может заранее выбрать оптимальные множества номеров подслушиваемых координат и, таким образом, не может заранее минимизировать количество частично стертых кодовых слов для снятия неопределенности о закодированной информации. Иначе говоря, при случайном стирании наблюдателю может потребоваться большее количество частично стертых кодовых векторов для снятия неопределенности, чем в ситуации, когда наблюдаемые координаты им выбираются. Следующие разделы посвящены задаче оценки необходимого количества наблюдения для снятия неопределенности.

3. Математическая модель атаки многократного наблюдения

Пусть $\mathbb{F} = \mathbb{F}_q$ — поле Галуа мощности q, $\underline{u} = \{1; ...; u\}$, C — $[n, n-k]$-код, $C \subset \mathbb{F}^n$, G — порождающая матрица кода C, представленная без потери общности в систематическом виде: $G = (I_{n-k}\, P)$, где I_{n-k} — единичная матрица размера $(n-k) \times (n-k)$, P — $(n-k \times k)$-матрица без нулевых столбцов. Сопоставим информационному вектору $\mathbf{s}(\in \mathbb{F}^k)$ кодовый вектор $\mathbf{c} = (\mathbf{v}, \mathbf{s} + \mathbf{v}P)(\in \mathbb{F}^n)$, где вектор $\mathbf{v}$ выбирается случайно и равновероятно из пространства $\mathbb{F}^{n-k}$. Несложно видеть, что при фиксированном информационном векторе $\mathbf{s}$ соответствующие кодовые векторы принадлежат одному смежному классу кода C, а для двух разных информационных векторов кодовые векторы будут разными. В связи с этим рассмотренный метод называется кодированием смежными классами [13, 17], а код C — базовым кодом [22]. Для извлечения информационного вектора $\mathbf{s}$ из кодового вектора $\mathbf{c}$ достаточно умножить слева вектор $\mathbf{c}$ на проверочную матрицу $H = \left(-P^\top\, I_{n-k}\right)$ базового кода C, где $P^\top$ — транспонированная матрица P.

Множество номеров наблюдаемых координат в кодовом векторе длины n обозначим через $\tau(\subseteq \underline{n} = \{1; ...; n\})$ и назовем *окном наблюдения*, а мощность множества τ назовем *размером окна наблюдения*, $\overline{\tau} = \underline{n} \backslash \tau$ — множество номеров ненаблюдаемых координат. Наблюдаемое значение при передаче по главному каналу вектора $\mathbf{c}$ обозначим $\mathbf{c}_\tau$.

Пусть наблюдатель может среди частично наблюдаемых векторов выделить набор

$$(\mathbf{c}_{\tau_i})_{i=1,...,l}\,, \quad |\tau_i| = \mu_i, \tag{1}$$

векторы в котором соответствуют одному информационному вектору $\mathbf{s}$; для $i \neq j$ в общем случае $\tau_i \neq \tau_j$ и $\mu_i \neq \mu_j$. Целью наблюдателя является восстановление неизвестного ему информационного вектора $\mathbf{s}$ по набору (1). Пусть $\mathcal{N}_{\tau_i}(\subseteq \mathbb{F}^k)$ — множество информационных векторов, которые позволяют наблюдать вектор $\mathbf{c}_{\tau_i}$, или иначе — множество претендентов. Наблюдатель с целью отыскания закодированного информационного вектора $\mathbf{s}$ по каждому вектору $\mathbf{c}_{\tau_i}$ независимо от ранее наблюдаемых векторов строит множество $\mathcal{N}_{\tau_i}$, а затем находит множество

$$\mathcal{N}_{\tau_1,\ldots,\tau_i} := \bigcap_{j=1}^{i} \mathcal{N}_{\tau_j} = \mathcal{N}_{\tau_1,\ldots,\tau_{i-1}} \cap \mathcal{N}_{\tau_i}, \tag{2}$$

тем самым пытаясь после i-го наблюдения сузить множество претендентов, которому принадлежит неизвестный вектор $\mathbf{s}$. Формулу (2) для удобства назовем *атакой многократного наблюдения*. Отметим, что атака (2) проста в реализации и может быть автоматизирована.

Так как все информационные сообщения для наблюдателя равновероятны, вероятность правильного восстановления информационного вектора $\mathbf{s}$ по набору (1) равна $1/|\mathcal{N}_{\tau_1,\ldots,\tau_l}|$. В [19] доказано, что

$$|\mathcal{N}_{\tau_1,\ldots,\tau_l}| = q^{\dim \bigcap_{i=1}^{l} \mathcal{L}_c(H_{\overline{\tau}_i})}, \tag{3}$$

где $H_{\overline{\tau}_i}$ — матрица, составленная из столбцов проверочной $(k \times n)$-матрицы H базового кода C с номерами столбцов из множества $\overline{\tau}_i$; $\mathcal{L}_c(H_{\overline{\tau}_i})$ — линейная оболочка, натянутая на столбцы матрицы $H_{\overline{\tau}_i}$, $i = 1, ..., l$.

Таким образом, если наблюдатель может выбирать множества стираемых кодовых символов в каждом кодовом слове, то он может, используя формулу (3), аналитически или вычислительно до проведения атаки (2) минимизировать мощность набора $(\tau_1, ..., \tau_l)$, т. е. минимизировать число l так, чтобы за l наблюдений информационный вектор $\mathbf{s}$ был однозначно восстановлен. (С точки зрения наблюдателя, чем меньше частичных кодовых векторов необходимо для снятия неопределенности об информационном векторе, тем лучше.) Также в случае, когда l фиксировано, наблюдатель может подобрать такие l окон наблюдения, что в результате множество претендентов $\mathcal{N}_{\tau_1,\ldots,\tau_l}$ будет иметь наименьшую из возможных мощность. Например, для кодов Хэмминга и Рида — Маллера задача поиска минимального набора окон наблюдений при фиксированном их размере решена в [19].

Однако, как отмечено выше, не всегда наблюдатель может контролировать модель стирания кодовых символов. Так, при случайном стирании кодовых символов ему заранее известна только вероятность стирания символов из заданного множества, что характерно при стирании кодовых слов на носителях информации. В [22], например, для моделей случайного стирания $\mathcal{M}^{U(\rho)}$ и $\mathcal{M}^{\mathrm{EC}(p)}$ аналитически получены формулы вычисления количества получаемой наблюдателем информации при однократном перехвате. Задача аналитической оценки стойкости метода случайного кодирования к атаке многократного наблюдения по схеме (2) представляется трудной даже в рамках простых моделей случайного стирания $\mathcal{M}^{U(\rho)}$ и $\mathcal{M}^{\mathrm{EC}(p)}$.

В следующем разделе строится алгоритм для экспериментальной оценки величины $p(\mathcal{M}, \alpha)$ — мощности набора вида (1) частично стертых в соответствии с моделью $\mathcal{M}$ кодовых векторов, по которым при использовании наблюдателем атаки по схеме (2) неизвестный информационный вектор может быть восстановлен с вероятностью α. Отметим, что $p(\mathcal{M}, 1)$ — это среднее количество наблюдаемых частично стертых в соответствии с моделью $\mathcal{M}$ кодовых слов, по которым гарантированно восстанавливается информационный вектор. С точки зрения отправителя и получателя, для заданного α максимальное количество g одинаковых информационных векторов в сообщении должно удовлетворять условию $g < p(\mathcal{M}, \alpha)$, чтобы наблюдатель не имел возможности восстановить информационный вектор с заданной им вероятностью α.

4. Алгоритм вычисления $p(\mathcal{M}, \alpha)$

Построим алгоритм экспериментальной оценки $p(\mathcal{M}, \alpha)$ в зависимости от модели случайного стирания $\mathcal{M}$. Для этого сначала построим алгоритм GetApplicantsCardinality для вычисления $|\mathcal{N}_{\tau_1,...,\tau_l}|$ в соответствии с (3). Для $i = 1, ..., l$ введем обозначения: $r_i = \mathrm{rank}(H_{\overline{\tau}_i})$, $\mathrm{Syst}(H_{\overline{\tau}_i}^\top)$ — алгоритм, который по матрице $H_{\overline{\tau}_i}^\top$ строит пару матриц (F_i, D_i), где F_i — $(n - \mu_i \times n - \mu_i)$-матрица, а D_i — перестановочная матрица, причем

$$F_i \times H_{\overline{\tau}_i}^\top \times D_i = \begin{pmatrix} I_{r_i} & P_i \\ O_{i,1} & O_{i,2} \end{pmatrix} =: G_i. \tag{4}$$

Здесь $O_{i,1}$ и $O_{i,2}$ — нулевые матрицы размера $((n - \mu_i - r_i) \times k)$ и $((n - \mu_i - r_i) \times k - r_i)$.

Исходные параметры: H — проверочная матрица базового кода C, τ_1,...,τ_l — набор окон наблюдений
Результат: $|\mathcal{N}_{\tau_1,...,\tau_l}|$
$M = (\mathbf{0})$ $(\mathbf{0} \in \mathbb{F}^k)$;
цикл $i = 1, ..., l$ **выполнять**
  $(F_i, D_i) := \mathrm{Syst}(H_{\overline{\tau}_i}^\top)$;
  построить матрицу G_i вида (4);
  $J_i := \left(-P_i^\top | I_{k-r_i}\right) \times D_i^{-1}$;
  $M := \begin{pmatrix} M \\ J_i \end{pmatrix}$;
  $i := i + 1$;
конец цикла
возвратить $q^{k-\det(M)}$.

Отметим, что так как в алгоритме GetApplicantsCardinality матрицы J_i вычисляются независимо друг от друга, этот алгоритм может быть распараллелен при программной реализации.

Лемма 1. *Алгоритм* GetApplicantsCardinality *по матрице H и множествам τ_1,...,τ_l вычисляет $|\mathcal{N}_{\tau_1,...,\tau_l}|$.*

Доказательство. Пусть $i \in \underline{l}$. Матрицу G_i вида (4) можно рассматривать как порождающую матрицу некоторого линейного $[k, r_i]$-кода C_i. Так как она имеет систематический вид, проверочная матрица кода C_i имеет вид $(-P_i^\top \ I_{k-r_i})$. Домножение на матрицу D_i^{-1} необходимо для сохранения порядка столбцов, зафиксированного в исходной матрице $H_{\overline{\tau}_i}$. Чтобы найти размерность пересечения линейных оболочек $\mathcal{L}_c(H_{\overline{\tau}_i})$ для всех $i = 1, ..., l$, необходимо найти размерность подпространства векторов, одновременно принадлежащих линейным кодам с проверочными матрицами J_i. То есть необходимо найти ранг $\left(kl - \sum\limits_{i=1}^{l} r_i \times k\right)$-матрицы вида $(J_1^\top \ J_2^\top \ ... \ J_l^\top)^\top$. □

Следствие 1. *Если в последовательности τ_1,...,τ_l хотя бы одно множество мощности n, то* $\mathrm{GetApplicantsCardinality}(H, \tau_1, ..., \tau_l) = 1$.

Доказательство. Без потери общности положим, что $\tau_1 = \underline{n}$. Тогда $\mathrm{rank}(H_{\overline{\tau}_1}) = 0$, следовательно, $J_1 = I_k$. Тогда построенная в алгоритме GetApplicantsCardinality матрица M будет иметь ранг k. □

Из следствия 1, в частности, получаем очевидный факт: в рамках вырожденной модели стирания, когда ни один кодовый символ не стирается, наблюдателю достаточно одного кодового слова для восстановления информационного вектора $\mathbf{s}$. Это действительно так, потому что $\tau_i = \underline{n}$ для всех i и, таким образом, $\mathrm{rank}(H_{\overline{\tau}_i}) = 0$. В более общем смысле: если модель стирания данных такая, что с большой вероятностью в кодовом слове не стирается ни один символ, то с большой вероятностью наблюдатель сможет раскрыть содержание закодированной информации уже при малой мощности набора (1). Кроме вырожденной модели стирания, очевидно, неподходящей с точки зрения защиты данных является модель $\mathcal{M}^{\mathrm{EC}(p)}$ при малых p (в частности, при $p<0.1$ для рассмотренных в настоящей работе кодов), так как высока вероятность события, когда в кодовом слове ни один кодовый символ не стирается. Также очевидно следующее

Следствие 2. *Если в последовательности $\tau_1, ..., \tau_l$ для каждого $i = 1, ..., l$ выполняется равенство* $\mathrm{rank}(H_{\overline{\tau}_i}) = k$, *то* $\mathrm{GetApplicantsCardinality}(H, \tau_1, ..., \tau_l) = q^k$.

Следствие 2 дает критерий совершенной защиты после l наблюдений, когда многократный (l-кратный) перехват не позволяет сделать множество претендентов меньшим по мощности, чем все пространство информационных векторов. Как показано в [17], если $|\tau| < d(C^\perp)$, где $C^\perp$ — код, ортогональный к базовому коду C, то $\mathrm{rank}(H_{\overline{\tau}}) = k$. Например, для $[7, 4]$-кода Хэмминга C имеем $d(C^\perp) = 4$. Следовательно, если в каждом кодовом слове останутся нестертыми любые три кодовых символа, то набор (1) из таких частичных кодовых слов для любого l не дает наблюдателю какой-либо информации об информационном векторе в рамках атаки (2). То же относится и к $[8, 4]$-коду Рида — Маллера C первого порядка, для которого $d(C^\perp) = 4$.

Таким образом, при использовании $[7, 4]$-кода Хэмминга и $[8, 4]$-кода Рида — Маллера для обеспечения совершенной защиты в каждом кодовом слове должны быть стерты (детерминированно или случайно) соответственно любые четыре и пять кодовых символов. Заметим, что как для кода Хэмминга, так и для кода Рида — Маллера первого порядка найдутся такие "подходящие" множества стираемых координат мощности соответственно 3 и 4, что наблюдение данных на нестертых координатах не даст какой-либо информации о закодированном векторе. В этом случае, к примеру, на EEPROM-носителях информации с целью сокращения числа стираемых ячеек для совершенной защиты от атаки (2) достаточно было бы стирать соответственно три и четыре координаты из таких "подходящих" наборов вместо соответственно четырех и пяти произвольных координат. Однако, как показано, например, в [20, 22], для $[8, 4]$-кода Рида — Маллера первого порядка таких "подходящих" наборов всего 14 из 70 возможных, что уменьшает период перезаписи ячеек памяти в пять раз по сравнению с тем, когда стираются любые четыре символа в каждом кодовом слове.

В следующем разделе экспериментально исследуется стойкость метода случайного кодирования на базе $[7, 4]$-кода Хэмминга и $[8, 4]$-кода Рида — Маллера в случаях, когда стирается небольшое количество кодовых символов при случайном и равновероятном выборе номеров стираемых координат либо когда количество стираемых кодовых символов является случайной величиной. Для этого разработан алгоритм CalcProbList экспериментального вычисления $p(\mathcal{M}, \alpha)$, который в общем случае может быть применим к любым кодам и моделям стирания.

Исходные параметры: H — проверочная матрица базового кода C, $\mathcal{M}$ — модель стирания, α_0 — предельное значение α, для которого оценивается $p(\mathcal{M}, \alpha)$, N — количество испытаний

Результат: список пар вида (l, α_l) для $l = 1, 2...$, $\alpha_l \geq \alpha_0$

$B = 0$, $l = 1$;

// список пар вида (l, α_l);

LIST $:= \emptyset$;

до тех пор, пока $B \leq \alpha_0$ **выполнять**

  $\Delta = 0$;

  цикл $i = 1, ..., N$ **выполнять**

    Выбрать в соответствии с моделью $\mathcal{M}$ множества τ_1,...,τ_l;

    $\Delta_i := \log_q(\text{GetApplicantsCardinality}(H, \tau_1, ..., \tau_l))$;

    $\Delta := \Delta + \Delta_i$;

  конец цикла

  $B := q^{-\Delta/N}$;

  LIST := LIST $\cup (l, B)$;

  $l := l + 1$;

конец цикла

возвратить LIST.

Отметим, что вычисление $p(\mathcal{M}, \alpha)$ теоретически возможно и без применения экспериментального алгоритма CalcProbList, если провести вычисления размерностей пересечений для всех возможных (в рамках заданной модели стирания) наборов $(\tau_1, ..., \tau_l)$. Однако сложность такого переборного способа растет экспоненциально по l. В частности, для модели $\mathcal{M}^{U(\rho)}$ потребуется перебрать $(C_n^\rho)^l$ последовательностей. Сложность же алгоритма CalcProbList имеет оценку $\mathcal{O}(N(l+1)k^3)$, где N — число опытов, а $\mathcal{O}((l+1)k^3)$ — вычислительная сложность алгоритма GetApplicantsCardinality, применяющего метод Гаусса l раз посредством обращения к алгоритму Syst и один раз при вычислении определителя. Количество опытов N определяется в соответствии с известной формулой для объема выборки при повторном отборе: $N = (\sigma/\phi)^2$, где σ — среднеквадратичное отклонение дисперсии случайной величины, моделирующей среднюю размерность пересечения подпространств (не превышает k), а ϕ — средняя ошибка повторной выборки (насколько величина Δ/N может отличаться от истинного значения средней размерности пересечения подпространств). Величина ϕ влияет на размер интервала, которому может принадлежать истинное значение вероятности определения информационного вектора после l перехватов. Другими словами, для каждой пары $(l, B) \in$ LIST истинная вероятность α угадывания информационного слова после l перехватов принадлежит интервалу $[B/q^\phi, B/q^{-\phi}]$. Например, для $q = 2$ при $\phi = 0.05$ для каждой пары $(l, B) \in$ LIST интервал будет иметь вид $[0.962B, 1.04B]$, при этом потребуется не более $(4/0.05)^2 = 6400$ опытов. Число опытов можно уменьшить за счет увеличения параметра ϕ, а также более тонкой оценки величины σ.

Теорема 1. *Пусть H — проверочная матрица базового кода C, $\mathcal{M}$ — фиксированная модель стираний кодовых символов,* LIST $=$ CalcProbList$(H, \mathcal{M}, \alpha_0, N)$, (l, α) — *элемент списка* LIST. *Тогда $p(\mathcal{M}, \alpha) = l$.*

Доказательство. Рассмотрим пару (l, α) — элемент списка, построенного по алгоритму CalcProbList$(H, \mathcal{M}, \alpha_0, N)$. В соответствии с алгоритмом CalcProbList $\alpha = \left(q^{\sum_{i=1}^{N} \Delta_l / N}\right)^{-1}$. Так как $q^{\sum_{i=1}^{N} \Delta_l / N}$ — средняя мощность множества претендентов после l наблюдений, α — вероятность найти информационное сообщение после l наблюдений в соответствии с моделью стирания $\mathcal{M}$. □

5. Вычисление $p(\mathcal{M}, \alpha)$ для кодов Хэмминга и Рида — Маллера

В рамках настоящей работы реализованы алгоритмы GetApplicantsCardinality и CalcProbList для оценки $p(\mathcal{M}, \alpha)$ в случае $\mathbb{F}_q = \mathbb{F}_2$. Проведены вычисления для моделей стирания кодовых символов $\mathcal{M}^{U(\rho)}$ и $\mathcal{M}^{\mathrm{EC}(p)}$, которые в рамках задачи гарантированного удаления данных на носителях информации просто реализуемы на программном или программно-аппаратном уровне. Отметим, что прикладной интерес, например, в системах хранения данных представляют коды малой длины [13, 23, 24]. Поэтому для исследования выбраны $[7, 4]$-код Хэмминга и $[8, 4]$-код Рида — Маллера, часто применяемые для борьбы с помехами в каналах передачи данных.

В обоих случаях в алгоритме GetApplicantsCardinality число опытов $N = 3000$; так как для выбранных кодов размерность пространства информационных векторов равна 4, средняя ошибка ϕ в этом случае будет не более $\sqrt{4^2/3000} = 0.07$. В рамках представленных ниже результатов предполагается, что кодовые символы кодируются независимо друга от друга с помощью подходящего WOM/WAM/WEM-кода. Однако отметим, что если кодовые символы случайного кода кодируются поблочно, то предлагаемый в настоящей работе способ также может быть применим путем использования той модели стирания, в которую преобразуются модели $\mathcal{M}^{U(\rho)}$ или $\mathcal{M}^{\mathrm{EC}(p)}$. Например, если для сокращения количества стираний используется $[3, 2]$-код Ривеста — Шамира, то модель $\mathcal{M}^{U(\rho)}$ преобразуется в модель случайного и равновероятного стирания ρ *соседних упорядоченных пар* кодовых символов, при этом первый символ в этой паре имеет нечетный номер, а второй четный.

Результаты вычислений в рамках модели $\mathcal{M}^{U(\rho)}$ для $[7, 4]$-кода Хэмминга показаны на рис. 3, *а*. На графике приведены три кривые для случаев, когда в каждом кодовом слове случайно и равновероятно стираются соответственно 3, 2 и 1 координаты или, что то же самое, случайно и равновероятно наблюдаются соответственно 4, 5 и 6 координат. Каждая кривая помечена числовым значением ρ — количеством стираемых координат. Случай, когда ни один кодовый символ не стирается, не показан, так как согласно следствию 1 в этом случае достаточно одного кодового вектора для снятия неопределенности об информационном векторе. Также не показаны случаи, когда стираются 4 и более координат, так как согласно следствию 2 в рамках такой модели стирания обеспечивается совершенная защита от атаки (2): размер окна наблюдения меньше $d(C^{\perp}) = 4$. На графике (см. кривую с меткой 3), например, видно, что при стирании трех координат в каждом кодовом векторе для правильного восстановления информационного вектора с вероятностью $\alpha = 0.95$ и более необходимо не менее $l = 33$ частичных кодовых слов.

Результаты вычисления в рамках той же модели стирания для случая, когда C — $[8, 4]$-код Рида — Маллера первого порядка, приведены на рис. 3, *б*. Графики для стирания пяти и более координат не показаны, так как в этом случае обеспечивается со-

вершенная защита от атаки (2): размер окна наблюдения меньше $d(C^{\perp}) = 4$ (см. следствие 2). На рис. 3, *б* видно, что при случайном и равновероятном стирании половины битов наблюдателю потребуется не менее $l = 45$ частично стертых кодовых слов для правильного восстановления вектора с вероятностью не менее $\alpha = 0.95$.

Таким образом, если для $[7, 4]$-кода Хэмминга и $[8, 4]$-кода Рида — Маллера при стирании соответственно произвольных четырех и пяти кодовых символов обеспечивается одинаковый уровень защиты (совершенная защита), то при стирании случайно и равновероятно соответственно произвольных трех и четырех кодовых символов предпочтительнее с точки зрения противодействия атаке (2) применять код Рида — Маллера либо комбинаторно-эквивалентный ему код. Также случайный код на основе кода Рида — Маллера обладает большей относительной скоростью передачи данных: по 4 бита информации на 8 кодовых битах, в то время как случайный код с базовым кодом Хэмминга кодирует каждые 3 бита информации в 7 кодовых битах.

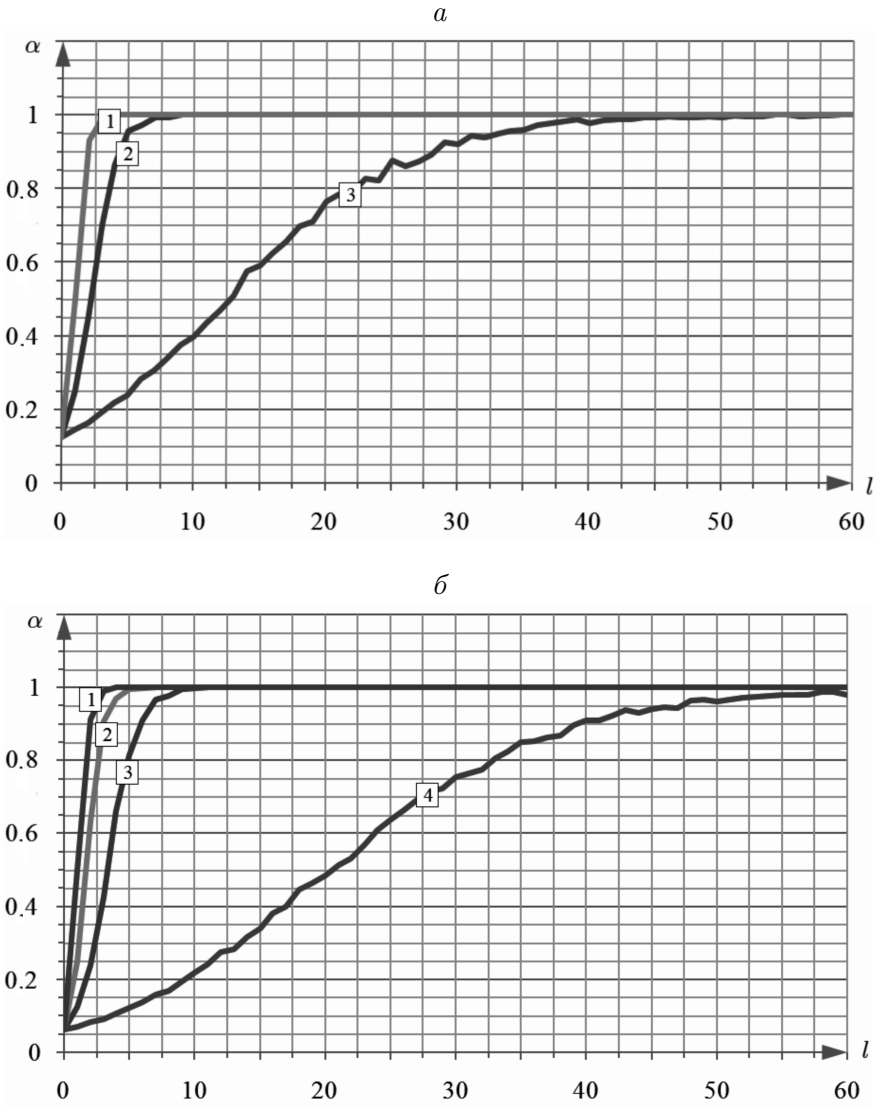

Рис. 3. Зависимость вероятности α восстановления информационного вектора от количества l частично стираемых кодовых слов для модели стирания кодовых символов $\mathcal{M}^{U(\rho)}$, $\rho \in \{1; 2; 3; 4\}$: *а* — $[7, 4]$-код Хэмминга; *б* — $[8, 4]$-код Рида — Маллера

Результаты вычислений для модели стирания $\mathcal{M}^{\mathrm{EC}(p)}$ при использовании тех же кодов показаны на рис. 4. Здесь случаи полного перехвата ($p = 0$, ни один кодовый символ не стирается) не показаны в силу их тривиальности. Каждая кривая помечена числом p, обозначающим вероятность стирания кодового символа в векторе. Заметим, что $[7, 4]$-код Хэмминга с точки зрения стойкости защиты от атаки (2) в рамках модели $\mathcal{M}^{\mathrm{EC}(p)}$ демонстрирует бóльшую стойкость, чем $[8, 4]$-код Рида — Маллера. Так, например, при $p = 0.6$ для успешного восстановления информационного вектора с вероятностью 0.95 в случае использования кода Хэмминга требуется наблюдение не менее 45 пар, а при использовании кода Рида — Маллера — не менее 30 пар. Вычисления также показали, что с ростом параметров базового кода количество наблюдений для восстановления информационного сообщения для заданного α увеличивается. Среднее количество l частично стертых кодовых слов в соответствии с моделями стирания $\mathcal{M}^{U(\rho)}$ (слева) и $\mathcal{M}^{\mathrm{EC}(p)}$ (справа), необходимое для снятия неопределенности об информационном век-

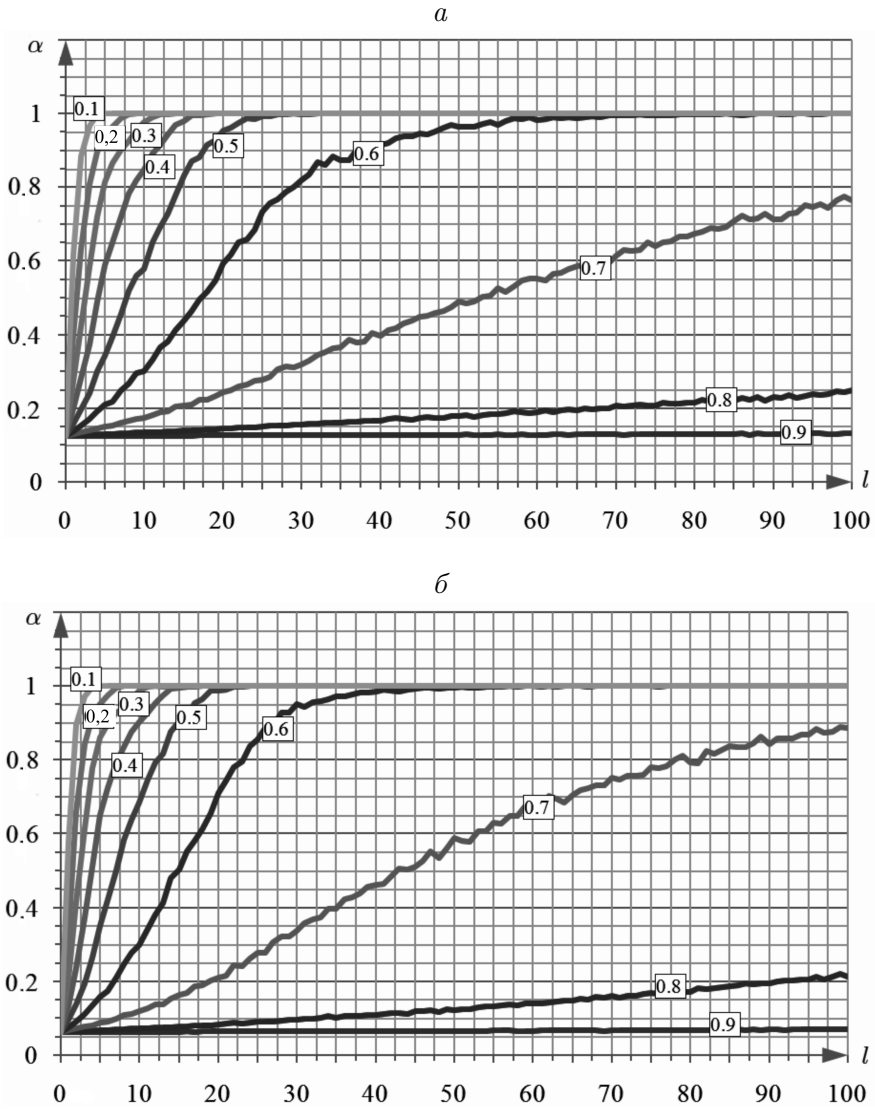

Рис. 4. Зависимость вероятности α восстановления информационного вектора от количества l частично стираемых кодовых слов для модели стирания кодовых символов $\mathcal{M}^{\mathrm{EC}(p)}$: *а* — $[7, 4]$-код Хэмминга; *б* — $[8, 4]$-код Рида — Маллера

торе с вероятностью $\alpha = 0.95$ для базового кода C — $[16, 11]$-кода Рида — Маллера, — приведено ниже.

ρ	8	7	6	5	4	3	2	1
l	5200	550	105	28	8	4	3	2

p	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
l	150000	5700	630	115	33	13	6	3

Отметим, что целесообразность применения конкретных кодов зависит от рассматриваемой модели стирания и прикладной задачи, в рамках которой осуществляется выбор кода. Так, например, если выбор кода выполняется в рамках задачи гарантированного удаления данных на EEPROM-носителях, то возможно использование как $[7, 4]$-кода Хэмминга, так и $[8, 4]$- и $[16, 11]$-кодов Рида — Маллера. Действительно, если согласно [12] наблюдатель в модуле памяти может выделить до 16 кодовых слов, соответствующих одному информационному вектору, то для гарантированного стирания подходят обе модели: $\mathcal{M}^{U(\rho)}$ и $\mathcal{M}^{\mathrm{EC}(p)}$. В частности, для защиты от гарантированного восстановления информационных векторов могут применяться модели стирания $\mathcal{M}^{U(3)}$ и $\mathcal{M}^{\mathrm{EC}(0.5)}$ — для $[7, 4]$-кода Хэмминга, $\mathcal{M}^{U(4)}$ и $\mathcal{M}^{\mathrm{EC}(0.5)}$ — для $[8, 4]$-кода Рида — Маллера, $\mathcal{M}^{U(5)}$ и $\mathcal{M}^{\mathrm{EC}(0.4)}$— для $[16, 11]$-кода Рида — Маллера.

Заключение

Простота метода кодирования смежными классами позволяет применять его в системах, ограниченных по количеству потребляемой энергии. В последнее время этот метод исследуется в рамках защиты конфиденциальности данных не только в подслушиваемых каналах связи, но и в системах хранения информации. При этом, если в системах связи модель наблюдения может быть под контролем наблюдателя, то в системах хранения модель наблюдения определяется легитимным владельцем носителя информации. В настоящей работе разработаны и обоснованы алгоритмы для оценки стойкости метода кодирования смежными классами в рамках модели случайного стирания данных. Анализ результатов вычислений показал, что стойкость к атаке многократного наблюдения зависит от модели стирания и применяемых кодов.

Представляется, что полученные в работе результаты также могут быть использованы для экспериментальной оценки стойкости защиты данных от подмены, когда наблюдателю доступна часть пары $(\mathbf{s}, \mathbf{y})$, где $\mathbf{s}(\in \mathbb{F}^{n-k})$ — информационное сообщение, а $\mathbf{y} = \mathbf{s}\pi_t(P) + \chi$ — подпись вектора $\mathbf{s}$, вычисленная в момент времени t на секретном ключе $(\chi, \pi_t) \in \mathbb{F}^k \times S_k$, $\pi_t(P)$ — матрица, полученная из матрицы P путем перемешивания столбцов в соответствии с перестановкой π_t. В силу простоты этого метода он может найти применение при дополнительной защите EEPROM-носителей, для которых задача защиты данных от подмены также является актуальной [25]. Отметим, что такой способ защиты от подмены позволяет одновременно исправлять ошибки, возникающие при выходе из строя отдельных ячеек памяти. Если наблюдателю пары (сообщение, подпись) доступны частично, например, в результате стирания части символов, то в этом случае алгоритм CalcProbList возвращает список пар вида (l, α), где l — это оценка снизу на количество информационных сообщений, по соответствующим частичным парам которых может быть восстановлен вектор χ с вероятностью не менее α. Однако эта оценка снизу является довольно грубой и она может быть большей, так как наблюдателю неизвестна перестановка π_t.

Благодарности. Авторы выражают признательность рецензентам за конструктивные замечания основных результатов и советы по представлению их в статье.

Список литературы / References

[1] **Букашкин С.А.** Метод случайного кодирования // Радиотехника. 2014. № 4(184). С. 31–36.
Bukashkin, S.A. The random coding method // Radiotekhnika. 2014. No. 4(184). P. 31–36. (in Russ.)

[2] **Mohan, V., Gurumurthi, S., Stan, M.R.** FlashPower: A detailed power model for NAND flash memory // Design, Automation & Test in Europe Conference & Exhibition (DATE). Dresden: IEEE, 2010. P. 502–507.

[3] **Jiang, A., Bruck, J.** Data representation for flash memories. Data Storage Design. Ed. Balasa, F. Rijeka. Croatia: In Tech Europe, 2010. P. 53–74.

[4] **Rivest, R.L., Shamir, A.** How to reuse a write-once memory // Inf. Control. 1982. Vol. 55, No. 1-3. P. 1–19.

[5] **Tal, A.** Two flash technologies compared: NOR vs NAND. White Paper, 91-SR-012-04-8L REV. 1.0, 2002. 10 p. Available at: https://focus.ti.com/pdfs/omap/diskonchipvsnor.pdf

[6] **Yaakobi, E., Kayser, S., Siegel, P.H., Vardy, A., Wolf, J.K.** Codes for write-once memories // IEEE Transactions on Information Theory. 2012. Vol. 58, No. 9. P. 5985–5999.

[7] **Yadgar, G., Yaakobi, E., Schuster, A.** Write once, get 50 % free: Saving SSD erase costs using WOM codes // Proc. of the 13th USENIX Conf. on File and Storage Technologies (FAST '15). Santa Clara, CA, US, 2015. P. 256–271.

[8] **Jiang, A., Bohossian, V., Bruck, J.** Floating codes for joint information storage in write asymmetric memories //IEEE Intern. Symp. on Information Theory, ISIT 2007. P. 1166–1170.

[9] **Ahlswede, R., Zhang, Z.** Coding for write-efficient memory // Information and Computation. 1989. Vol. 83, No. 1. P. 80–97.

[10] **Ahlswede, R., Cai, N.** Models of multi-user write-efficient memories and general diametric theorems // Information and Computation. 1997. Vol. 135, No. 1. P. 37–67.

[11] **Kissel, R., Regenscheid, A., Scholl, M., Stine, K.** Guidelines for media sanitization. NIST Special Publication 800-88 Revision 1, 2006. 41 p.

[12] **Wei, M., Grupp, L.M., Spada, F.E., Swanson, S.** Reliably erasing data from flash-based solid state drives // Proc. of the 9th USENIX Conf. on File and Stroage Technologies, February 15–17, 2011, San Jose, California. 8 p.

[13] **Cassuto, Y., Bandic, Z.** Low-complexity wire-tap codes with security and error-correction guarantees // Proc. of the IEEE Information Theory Workshop, Dublin, Ireland, 2010. 5 p.

[14] **Qing, Li, Jiang, A.** Coding for secure write-efficient memories // 52nd Annual Allerton Conference on Communication, Control, and Computing. Allerton, 2014. Monticello, IL: IEEE, 2014. P. 505–512.

[15] **Wyner, A.D.** The wire-tap channel // Bell System Technical Journal. 1975. Vol. 54, No. 8. P. 1355–1387.

[16] **Коржик В.И., Яковлев В.А.** Неасимптотические оценки эффективности кодового зашумления одного канала // Пробл. передачи информации. 1981. Т. 17, вып. 4. С. 11–18.
Korzhik, V.I., Yakovlev, V.A. Non-asymptotic estimates for efficiency of code jamming in a wiretap channel // Problemy Peredachi Informatsii. 1981. Vol. 17, No. 4. P. 11–18. (in Russ.)

[17] **Ozarov, L.H., Wyner, A.D.** Wire-tap channel II // Bell System Technical Journal. 1984. Vol. 63. P. 2135–2157.

[18] **Косолапов Ю.В.** Коды для обобщенной модели канала с подслушиванием. // Пробл. передачи информации. 2015. Т. 51, № 1. С. 23—28.

Kosolapov, Yu.V. Codes for a generalized wire-tap channel model // Problemy Peredachi Informatsii. 2015. Vol. 51, No. 1. P. 23—28.

[19] **Деундяк В.М., Косолапов Ю.В.** Об одном методе снятия неопределенности в канале с помехами в случае применения кодового зашумления // Изв. ЮФУ. Техн. науки. 2014. № 2(151). С. 197–208.
Deundyak, V.M., Kosolapov, Y.V. One method of removing the uncertainty in the channel with errors in the case of code noising // Izvestiya SFedU. Engineering Sciences. 2014. No. 2(151). P. 197–208. (in Russ.)

[20] **Косолапов Ю.В., Никулин В.Э.** Способ организации распределенного хранилища, устойчивого к частичной утечке данных // Матер. XIII Междунар. науч.-практ. конф., ИБ-2013. Ч. I. Таганрог: Изд-во ЮФУ, 2013. С. 186–191.
Kosolapov, Y.V., Nikulin, V.E. Way of the organization of the distributed storage resistant against a partial data leakage // Materials of the XIII Intern. Scientific and Practical Conf. "IS-2013". Pt I. Taganrog: SFU Publishing House, 2013. P. 186–191. (in Russ.)

[21] **Wei, V.K.** Generalized hamming weights for linear codes // IEEE Transactions on Information Theory. 1991. Vol. 37, No. 5. P. 1412–1418.

[22] **Косолапов Ю.В., Курчев Н.О.** О вычислении меры стойкости кодового зашумления в канале со случайным частичным перехватом // Вычисл. технологии. 2014. Т. 19, № 6. С. 42–53.
Kosolapov, Y.V., Kurchev, N.O. Computation of the measure of resistance for Code Noising in channel with random partial interception // Comput. Technologies. 2014. Vol. 19, No. 6. P. 42–53. (in Russ.)

[23] **Huang, Q., Lin, S., Abdel-Ghaffar, K.** Error-correcting codes for flash coding // IEEE Transactions on Information Theory. 2011. Vol. 57, No. 9. P. 6097–6108.

[24] **Sunberg, C.W.** Erasure and error decoding for semiconductor memories // IEEE Transactions on Computers. 1978. Vol. c-27, No. 8. P. 696–705.

[25] **Elbaz, R., Champagne, D., Gebotys, C.H., Lee, R.B., Potlapally, N.R., Torres, L.** Hardware mechanisms for memory authentication: A survey of existing techniques and engines // Transactions on Computational Science. 2009. No. 4. P. 1–22.

Поступила в редакцию 4 июня 2015 г.,
с доработки — 25 сентября 2015 г.

On the experimental estimation of the lower bound for the maximum number of messages in a scheme aimed at data protection against spoofing

Gazaryan, Yury O., Kosolapov, Yury V.*

South Federal University, Rostov-on-Don, 344006, Russia
*Corresponding author: Kosolapov, Yury V., e-mail: itaim@mail.ru

Purpose: For a given protection scheme we estimate the lower bound of the maximum number of messages that may be signed by one fixed key (known for both sender and receiver). It is supposed that a channel between sender and receiver is noisy. The observer is supposed to get only the partial data from the pair ("message", "signature") and knows everything about message signing algorithm except of a signing key.

Methodology: To solve this problem we apply some well-known results of the analysis for the resistance of a code noising to multiple monitoring that uses the data channel with erasures, which allowed, together with methods of mathematical statistics, a method to obtain the lower estimate of the maximum number of messages.

Findings: We constructed and justified algorithms for experimental evaluation for the lower bound of maximum number of messages for two particular models of interceptions: the model of uniform random interception with fixed number for coordinates of the message-signature pairs, represented as a vector, and a model for interception of messages over channel with erasures with a fixed probability of interception for one symbol. We present the results of calculations for specific implementations of the considered protection scheme.

Originality/value: The results of the current research besides the considered spoofing protection scheme can be also used in another schemes of data protection. For example, these results could be useful in experimental analysis of the strength of scheme that protects confidentiality of the data with code noising against multiple interceptions.

Keywords: protection from spoofing messages, multiple partial interception, code noising.

Received 4 June 2015
Received in revised form 25 September 2015