ИЗВЕСТИЯ РАН. ТЕОРИЯ И СИСТЕМЫ УПРАВЛЕНИЯ, 2020, № 5, с. 60–72

ОБРАБОТКА ИНФОРМАЦИИ И ИДЕНТИФИКАЦИЯ

УДК 003.26, 57.087.1, 612.087.1

ОПТИМИЗАЦИЯ ВСТРАИВАНИЯ КРИПТОГРАФИЧЕСКИХ КЛЮЧЕЙ В БИОМЕТРИЧЕСКИЕ ДАННЫЕ[1]

© 2020 г. Э. Т. Зайнулина[a],*, И. А. Матвеев[b],**

[a] *МФТИ, Долгопрудный, МО, Россия*

[b] *Федеральный исследовательский центр "Информатика и управление" РАН, Москва, Россия*

**e-mail: zaynulina.et@phystech.edu*

***e-mail: matveev@ccas.ru*

Поступила в редакцию 06.04.2020 г.
После доработки 20.04.2020 г.
Принята к публикации 25.05.2020 г.

Двумя важными компонентами современных систем контроля и управления доступом являются криптография и биометрия. Криптографические системы сами по себе высоконадежны, но требуют точного воспроизведения ключей доступа, что человек не в состоянии сделать самостоятельно, а соответствующие устройства могут быть утеряны или похищены. Биометрические данные у человека всегда при себе, но они изменчивы: невозможно повторно получить те же значения признаков. В работе предлагается способ объединения криптографического ключа и биометрических признаков радужной оболочки глаза. При этом получается ключ, из которого нельзя извлечь ни одну из двух исходных компонент, пока не будут предъявлены близкие к исходным биометрические признаки, т.е. данные того же человека. Метод объединения (кодер) и извлечения (декодер) состоят из нескольких отдельных последовательно исполняемых шагов. Подбор параметров осуществляется с помощью решения задачи дискретной оптимизации, состоящей в том, чтобы при некотором заданном пороге коэффициента ложного допуска минимизировать значение коэффициента ложного отказа в допуске. При этом есть ограничения в виде минимального размера кодируемого ключа и максимального размера итогового ключа. Проведены численные эксперименты на базах данных, находящихся в открытом доступе.

DOI: 10.31857/S0002338820050145

Введение. В настоящее время повсеместно применяется защита информации, основанная на криптографических алгоритмах. Таких алгоритмов, равно как и способов их применения, изобретено большое количество [1]. Здесь мы ограничимся случаем симметричного шифрования, при котором на этапе кодирования из *сообщения* M и секретного *ключа* K функцией-*кодером* Φ вычисляется *код* $C = \Phi(M, K)$, а на этапе декодирования функция-декодер Ψ восстанавливает сообщение: $M = \Psi(C, K)$. Из кода C без ключа K невозможно получить M, поэтому C будет открытым, не секретным. Свойством симметричного шифрования является то, что используемый ключ K должен быть повторен совершенно точно, например, в случае двоичной последовательности должны совпадать все ее биты. Острая проблема при этом состоит в том, что такие ключи легко отчуждаются (передаются, похищаются, теряются). Столь же насущная проблема – слабая способность человека запоминать парольные последовательности. Если придуманный лично им пароль человек в состоянии запомнить и воспроизвести (хотя уже здесь появляются трудности), то для автоматически сгенерированной последовательности из нескольких десятков псевдослучайных символов это практически невозможно [2].

В то же время человек располагает легко извлекаемыми, сложно отчуждаемыми и имеющими значительный информационный объем биометрическими признаками. Речь идет прежде всего о радужной оболочке глаза (РОГ) [3] и, в меньшей степени, об изображениях лица [4], отпечатках пальцев [5], подписи [6]. Биометрия редко утрачивается, сложно подделывается, легко предъявляется при наличии специального оборудования. Недостатком биометрических признаков в данном приложении является их изменчивость: невозможно в точности повторить результаты

[1] Работа выполнена при частичной финансовой поддержке РФФИ (проект № 19-07-01231).

измерения, можно лишь утверждать, что два набора признаков, полученные от одного человека, в некотором смысле различаются слабее,чем наборы, взятые от разных людей.

Представляет большой интерес объединение двух подходов, т.е. разработка методов точного воспроизведения (генерации) криптографического ключа из биометрических признаков, или замаскированного добавления существующего ключа к таким признакам, или иных методов и сценариев связывания строго определенных и имеющих необходимые статистические свойства данных, пригодных к использованию в криптографических протоколах, с наборами изменчивых биометрических признаков.

1. Выбор метода встраивания ключа в биометрию. Автоматическая биометрия уже достаточно долго используется для идентификации (поиска) и аутентификации (подтверждения) личности человека. Центральную задачу, решаемую такими системами, можно сформулировать как задачу создания оптимального классификатора: необходимо построить функцию расстояния между двумя наборами биометрических данных $\rho(D_1, D_2)$ и задать порог θ так, что как можно большее число пар D_1 и D_2, принадлежащих одному человеку, дает расстояние $\rho(D_1, D_2) < \theta$, а как можно большее число от разных людей дает $\rho(D_1, D_2) > \theta$. Функцию ρ можно представить как суперпозицию двух этапов: вычисление биометрического эталона T по данным $T = T(D)$, т.е. выделение признаков, устойчиво близких для одного человека и различающихся для разных людей, и расчет собственно расстояния $\rho(T_1, T_2)$. Эталон T представляет собой набор данных размером от нескольких сотен байт до нескольких килобайт, хотя элементы этих данных (в отличие от криптографических ключей) сильно коррелированы. Тем не менее, информационная емкость (энтропия) биометрического эталона сравнима или превосходит таковую для используемых в настоящее время криптографических ключей [7]. Например, для РОГ существует оценка в 249 степеней свободы [8], для отпечатка пальца разработана система, выделяющая 80 некоррелированных параметров [9]. Это дает основания считать, что возможно внедрить криптографический ключ в биометрию, не снизив его стойкости.

Следует отметить, что основное количество работ, посвященных применению криптографических методов в биометрии, развивают схему *отменяемой биометрии* (cancelable biometrics) [10]. Суть ее состоит в преобразовании биометрических данных в формат, из которого их нельзя извлечь в исходном виде, но в котором их можно сравнивать в системе распознавания. Отменяемая биометрия – ни что иное как разновидность нечеткого хеширования [11], что можно формально представить как введение дополнительного шага в расчет функции расстояния ρ. В этой схеме вычисляется $\rho(H_1, H_2)$ (в некоторых вариантах $\rho(H_1, T_2)$), где H – хэш-функция биометрического эталона $H = H(T)$. Очевидно, что по-прежнему решается задача распознавания.

Существует два подхода к тому, каким образом обрабатывать изменчивую биометрию, приводя ее к неизменному криптографическому ключу. В первом используются уже вычисленные биометрические признаки, которые исправляются различными вариантами избыточного кодирования с коррекцией ошибок. В русле данного подхода сделана и эта работа. Во втором подходе, развиваемом отечественными исследователями [12], биометрические признаки явно не вычисляются, вместо этого на “сырых” биометрических данных тренируется нейросеть, обучаемая выдавать тот или иной код [13]. Преимуществом этого подхода называется меньшая избыточность кодирования за счет использования на всех этапах непрерывных данных и квантования только в конце. Недостатки – непредсказуемость обучения нейросети, отсутствие гарантированного качества работы, в том числе отсутствие гарантии того, что результаты сохранят качество при работе с большим разнообразнем данных, чем использовалось при обучении.

Задача воспроизведения криптографического ключа решается *биометрическими криптосистемами* (biometric crypto system) [10], которые делятся на два класса, реализующих разные схемы работы: *генерации ключа* (key generation) и *встраивания ключа* (key binding).

В [14, 15] исследовались методы прямой генерации ключа из биометрических данных без использования какой-либо дополнительной информации. В этом случае при регистрации и распознавании вычисляется одна и та же функция, отображающая многообразие биометрических данных в пространство криптографических ключей (как правило, битовых строк): $K(D)$: $D \to \{0,1\}^n$, где n – битовая длина ключа. При этом должны выполняться условия

$$\begin{aligned} D_1, D_2 &- \text{данные одного человека} \;\Rightarrow\; K_1 = K_2, \\ D_1, D_2 &- \text{данные разных людей} \;\Rightarrow\; K_1 \neq K_2, \end{aligned} \tag{1.1}$$

где $K_1 = K(D_1)$, $K_2 = K(D_2)$. Невыполнение первого условия является ошибкой первого рода (ложный недопуск), ее вероятность обозначается FRR (false reject rate), невыполнение второго – ошибка второго рода (ложный допуск), ее вероятность обозначается FAR (false accept rate). Из-за изменчивости биометрических данных такой подход дает большое количество ошибок: FRR = 24% при FAR = 0.07% [15] даже на базе однородных качественных изображений CASIA-3-Lamp [16], что делает его малопригодным на практике.

Лучшие показатели имеет схема с использованием *вспомогательного кода* (helper code). В этом случае при регистрации кроме криптографического ключа $K_1 = K(D_1)$, который применяется для шифрования сообщения M и немедленно после этого уничтожается, при помощи кодера вычисляется вспомогательный код $h = \Phi(D_1)$. Он обладает следующими свойствами: (1) исходные данные D_1 из него восстановить невозможно; (2) при предъявлении биометрии D_2 с помощью декодера можно вычислить ключ $K_2 = \Psi(D_2, h)$, который удовлетворяет условиям (1.1). Таким образом, предъявив вспомогательный код h и биометрию, пользователь может получить ключ K_1, а значит, и сообщение M, зашифрованное этим ключом при регистрации. Нарушитель, даже зная h, не сможет получить K_1 [17].

Если в качестве ключа использовать сами данные $K_1 = D_1$, то восстанавливается исходная биометрия. Такая схема называется *защищенным оттиском* (secure sketch) [18]. С помощью вспомогательного кода можно работать даже с весьма изменчивыми биометрическими данными. Однако имеющиеся в печати работы показывают довольно скромные результаты. Например, в [19] делается предположение о внутриклассовой изменчивости в 10% признаков, хотя на практике она составляет больше 20%, что приводит к неработоспособности предлагаемого метода.

Схема встраивания ключа в заданных выше терминах выглядит как совокупность функций кодера $C = \Phi(K_1, D_1)$, декодера $K_2 = \Psi(C, D_2)$, удовлетворяющих условию (1.1). Ключ K_1 задается извне, что является преимуществом, поскольку не нужно учитывать природу алгоритма шифрования. С этой точки зрения ключ K_1 можно воспринимать как внешнее по отношению к системе кодирования сообщение M, что немедленно приводит к сценарию, сходному с симметричным шифрованием. Отличие состоит в том, что в симметричном шифровании секретный ключ K не должен изменяться, а биометрические признаки (тоже секретные) отличаются при кодировании и декодировании: $D_1 \neq D_2$. Такая схема называется *нечетким экстрактором* (fuzzy extractor) [18]. Схемы биометрической идентификации и нечеткого экстрактора приведены на рис. 1.

Если биометрические данные представляют собой набор действительных чисел $D \in \mathbb{R}^d$, а функции Φ и Ψ обратны друг другу: $K = \Psi(\Phi(K, D), D)$, то полученная схема называется *экранирующими функциями* (shielding functions) [20]. В современных системах регистрации и распознавания РОГ признаками являются целые числа или даже биты, поэтому использование экранирующих функций затруднено.

Очень популярное направление развития схемы встраивания ключа – *нечеткое хранилище* (fuzzy vault) [7]. Данный подход по сути есть приложение схемы разделения секрета Шамира [21]. При этом получены достаточно низкие, имеющие практический смысл значения ошибок: в [22] при нулевом FAR достигнуто значение FRR = 0.78%, в [23] FRR = 4.8%. Однако каждый из этих результатов показан на одной небольшой (меньше 1000 изображений) базе.

Наиболее перспективной для использования в системах биометрии РОГ является схема *нечеткого связывания* (fuzzy commitment) [24]. Для радужки можно построить простой метод, описанный в [25]. Эталон радужки в большинстве систем распознавания представляет собой битовое изображение (растр) определенных размеров, как правило, это $N = 256 * 8 = 2048$ бит, каждый бит – локальный признак, обычно знак свертки с фильтром в данной точке. Основой сравнения таких эталонов является нормированное расстояние Хэмминга – доля несовпадающих бит:

$$\rho(T_1, T_2) = \frac{1}{N}\sum_{i=1}^{N} T_1(i) \oplus T_2(i). \tag{1.2}$$

Если расстояние меньше некоторого порога p, который устанавливается разработчиками системы, то эталоны T_1 и T_2 признаются принадлежащими одному человеку.

Основная идея [25] состоит в использовании кодирования, исправляющего ошибки (error correcting code, ECC) [26]. Схема кодирования состоит из пары функций: кодера $R = \Phi_p(M)$,

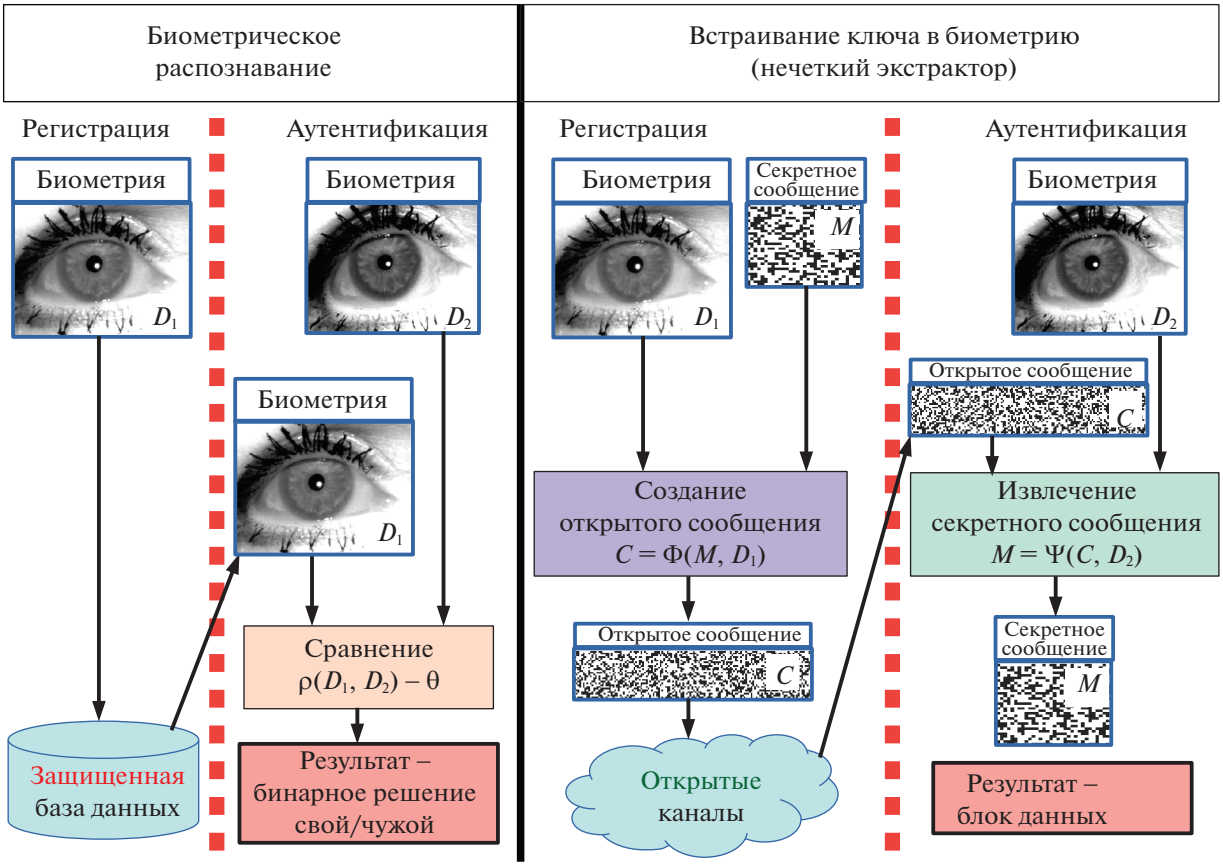

Рис. 1. Схемы биометрической идентификации и нечеткого экстрактора

отображающего сообщение в больший по размеру *код с избыточностью* R, и декодера $M = \Psi_p(R)$, где p – доля ошибок в коде R, которая может быть исправлена декодером. Секретное сообщение M кодируется, причем вероятность "ошибки передачи данных" устанавливается равной порогу классификации. Код $R_1 = \Phi_p(M)$ (в общем случае представляющий собой псевдослучайные числа) побитово складывается по модулю 2 (исключающее или) с эталоном радужки $C = R_1 \oplus T_1$. После вычисления C эталон и секретное сообщение уничтожаются. Полученные данные C также имеют вид случайных, из них нельзя извлечь ни эталон радужки T_1, ни сообщение M, их допустимо передавать по незащищенным каналам. Для декодирования регистрируется радужка и формируется новый эталон T_2, который, конечно, не совпадает с исходным. Однако если эталоны T_1 и T_2 принадлежат одному человеку, то они близки в смысле расстояния (1.2): $\rho(T_1, T_2) \le p$. Можно записать $R_2 = C \oplus T_2 = (R_1 \oplus T_1) \oplus T_2$, откуда $R_1 \oplus R_2 = T_1 \oplus T_2$, значит, справедливо $\rho(R_1, R_2) \le p$ и можно восстановить секретное сообщение $M = \Psi_p(R_2)$. Схема работы показана на рис. 2. Интересным развитием этого метода представляется обобщение операции сложения по модулю 2 до более широкого класса операторов поиска различия [27, 28].

В [25] используется каскад двух алгоритмов помехоустойчивого кодирования: Рида–Соломона [29] и Адамара [26]. Кодирование Рида–Соломона обрабатывает весь блок данных длины L, трактуя его как последовательность из L/s s-битовых символов, при этом любые как угодно расположенные символы (не биты!) могут отличаться, если их количество не больше pL. Это кодирование предназначено для того, чтобы компенсировать групповые ошибки, возникающие из-за наличия различных затенений (ресниц, бликов), закрывающих значительные области радужки. Кодирование Адамара исполняется для малых групп данных (несколько бит), и исправляет не более 25% ошибок в каждой группе. Другими словами, ошибки (различия эталонов T_1 и T_2) должны быть распределены равномерно по эталону с плотностью не выше 25%. Это кодирование предназначено для борьбы с точечными различиями, вызываемыми шумом камеры.

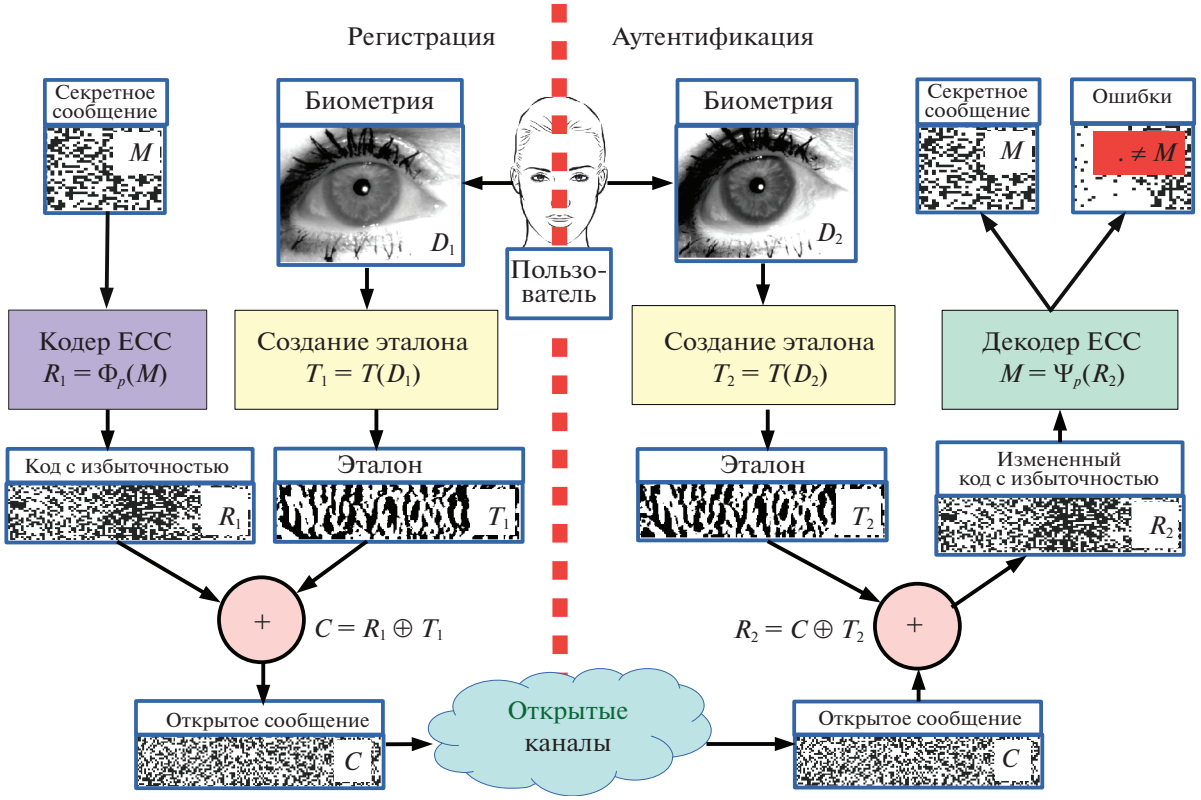

Рис. 2. Схема работы метода [25]

Сообщение M кодируется последовательно сначала кодом Рида–Соломона, потом результат – кодом Адамара.

Однако данный каскад алгоритмов может использоваться лишь, если доля различающихся бит в эталонах одного человека не превышает 25%. На реальных базах данных и в приложениях это не так, и такое ограничение приводит к неприемлемо высокой (более 50%) ошибке первого рода. Для преодоления этого затруднения в [30] дополнительно вводится маскирование эталона: каждый четвертый бит эталонов РОГ устанавливается в 0. За счет этого доля различающихся бит в эталонах одного человека снижается до 20%. Критика этого метода, с точки зрения неустойчивости ко взлому, приведена в [31]. Атака осуществляется путем постепенного восстановления исходного эталона (приближения к нему).

В данной работе предпринята попытка доработать схему [25] более разумным образом и построить практически пригодный метод встраивания ключа. На основе системы выделения признаков РОГ проведены численные эксперименты на нескольких открытых базах изображений, определено пороговое расстояние Хэмминга. Вводятся дополнительные два шага: простое мажоритарное кодирование и псевдослучайное перемешивание. Параметры полученных четырех последовательных шагов кодера (и соответствующих им шагов декодера) подбираются решением дискретной оптимизационной задачи.

2. Определение пороговой вероятности. Алгоритм биометрического распознавания можно разбить на функцию выделения признаков (построения эталона) $T = T(D)$ и функцию сравнения эталонов $\rho(T_1, T_2)$. Биометрические системы распознавания прошли большой путь и достигли очень хороших показателей точности классификации (малых значений ошибок), в том числе за счет оптимизации методов выделения признаков. Поэтому обосновано использование признаков, получаемых имеющимися методами. В работе применялось выделение области радужки на изображении [32] и вычисление признаков РОГ [33], которые в совокупности реализуют функцию $T = T(D)$.

Согласно развиваемой в работе разновидности нечеткого экстрактора, предложенной в [25], для сравнения используется расстояние Хэмминга (1.2). В этой схеме если биометрический

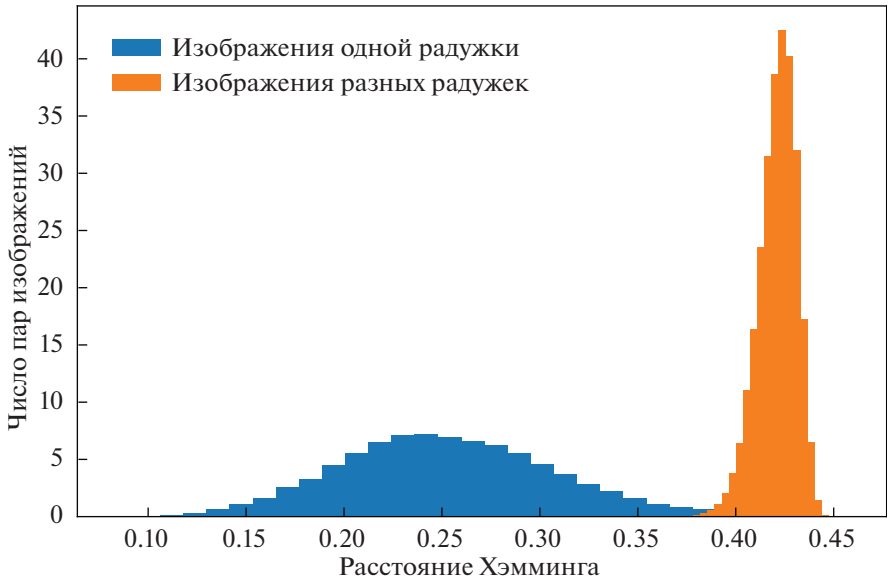

Рис. 3. Пример распределения сравнений “своих” и “нарушителей” по расстоянию

эталон T_1, использованный при регистрации, и предъявляемый эталон T_2 находятся на расстоянии меньше порога p, то зашифрованное сообщение M восстанавливается, иначе – нет. Другими словами, порог p разделяет “своих” и “нарушителей”. На рис. 3 показана типичная картина распределения сравнений по расстоянию $\rho(T_1, T_2)$. Левая группа соответствует случаю, когда T_1 и T_2 получены регистрацией одной радужки (“свои”). При этом исключены сравнения эталона с собой (дающие нулевое расстояние), как не встречающиеся на практике. Правая группа соответсвует случаю, когда T_1 и T_2 получены с разных радужек (“нарушители”). Необходимо определить величину порога.

Для экспериментов использованы базы данных (БД), находящиеся в открытом доступе: CASIA-4-Thousand [16], BATH [34], подмножество ICE базы NDIRIS [35], UBIRIS-1 [36]. Кроме того также участвовали базы изображений, собранных авторами [32, 33] на устройствах регистрации РОГ Panasonic BM-ET300 [37], LG IrisAccess-3000 [38] и Iritech Irishield MK2120U [39].

В табл. 1 приведен список семи использованных БД. Для каждой даны их собственные параметры (количество индивидуальных радужек и количество изображений) и значение порога, при котором ошибка второго рода равна 10^{-4}, что соответствует вероятности угадать четырехзначный пин-код банкомата. Из таблицы видно, что, взяв порог $\theta = 0.35$, можно добиться того, чтобы, предъявив чужую биометрию, получить доступ к секрету лишь с вероятностью, меньшей 10^{-4}.

Таким образом, значение $\theta = 0.35$ далее используется как базовое, т.е. трактуется как вероятность ошибки, которую должен исправлять код. В табл. 1 также приведены значения вероятностей

Таблица 1. Характеристики баз данных и пороги

БД	Количество		θ при FAR $= 10^{-4}$	FAR при $\theta = 0.35$, $\times 10^{-4}$	FRR при $\theta = 0.35$, $\times 10^{-2}$
	радужек	изображений			
BATH	1600	31988	0.402	0.03	4.46
CASIA	2000	20000	0.351	0.97	6.71
ICE	242	2953	0.395	0.011	7.13
Iritech	426	22954	0.356	0.79	3.88
LG	265	2864	0.386	0.07	2.36
Panasonic	277	2890	0.389	0.2	4.74
UBIRIS	240	1207	0.401	0.001	5.18

ошибок первого и второго рода для этого порога. Очевидно, вероятность ошибки второго рода лишь убывает на каждой из баз и тем сильнее, чем собственное значение порога при $\mathrm{FAR} = 10^{-4}$ отличается от 0.35. При этом максимальное значение ошибки первого рода не превышает 8%.

3. Описание методов. Описывать применяемые методы будем в последовательности их исполнения декодером, что также соответствует переходу “от простого к сложному”, так как вначале единицей данных является один бит, а в конце – все сообщение.

Итак, необходимо сконструировать код, восстанавливающий исходное сообщение из блока данных, каждый бит которых может быть изменен со средней вероятностью не более $p = 0.35$. Во-первых, это значение существенно больше $p = 0.25$, являющегося пороговым для возможности применения популярных методов Уолша–Адамара или Рида–Мюллера [40]. Во-вторых, возникновение этих ошибок нельзя считать независимым. Напротив, ошибки в близких элементах сильно коррелированы (возникают блоками).

3.1. Декоррелирование псевдослучайным перемешиванием. Проектировать методы, исправляющие коррелированные ошибки, значительно труднее, и их показатели хуже, чем для случая некоррелированных ошибок. Значительная часть усилий в этом случае направлена именно на декорреляцию. В рассматриваемой системе весь блок данных доступен целиком (а не последовательно, как во многих системах, работающих с каналами передачи) и можно применить простой метод декорреляции – псевдослучайное перемешивание битов эталона радужки. Если ко всем эталонам применяется одинаковое перемешивание, то расстояние Хэмминга не изменяется и сохраняются все конструкции, основанные на нем. Но при этом соседние биты последовательностей, используемых в кодах на следующих этапах, берутся уже из разнесенных пикселей эталона и ошибки в них независимы.

3.2. Битовое мажоритарное кодирование. Уровень ошибок $p = 0.35$ слишком высок для большинства корректирующих кодов. Практически единственной возможностью здесь остается мажоритарное кодирование отдельных битов, исправляющее до 50% ошибок. При кодировании значение бита повторяется n раз, при декодировании подсчитывается сумма принятых n бит, и если она меньше $n/2$, то принимается значение 0, иначе 1. Если p – вероятность ошибки в единичном бите и искажения битов независимы, то вероятность ошибки при приеме составляет

$$p_D(p) = 1 - \sum_{l=0}^{(n-1)/2} C_n^l p^{n-l}(1-p)^l = 1 - (1-p)^n \sum_{l=0}^{(n-1)/2} C_n^l \left(\frac{p}{1-p}\right)^l, \tag{3.1}$$

где C_n^l – число сочетаний из n по l.

На рис. 4 приведены графики функции (3.1) при некоторых n.

Видно, что если вероятность ошибки одного бита кода равна $p = 0.35$, то, семикратно продублировав бит сообщения, можно передать его с вероятностью ошибки $p_D = 0.2$, что позволяет использовать коды Адамара. Большая кратность дублирования также допустима, но приводит к большему размеру кода.

Параметром этого метода является кратность повтора бита n.

3.3. Блоковое кодирование Адамара. Обозначим через $\mathbb{B} = \{0,1\}$, $\mathbb{B}^l = \{(b_1, \ldots, b_l)$: $b_l)$: $b_i \in \mathbb{B}$, $i = \overline{1,l}\}$ множество всех двоичных последовательностей длины l (l-мерный двоичный куб). Для любых двух элементов $u, v \in \mathbb{B}^l$ задано нормированное *расстояние Хэмминга* (1.2), равное отношению количества различающихся бит в последовательностях к их длине. Также обозначим через $b(u)$ число единичных бит в последовательности u.

Пусть M – исходное *сообщение* (message) длиной k бит, $M \in \mathbb{B}^k$, т.е. единицей кодирования является блок из нескольких бит. Алгоритм кодирования Адамара отображает сообщение в избыточный *код* длиной n бит из некоторого известного алфавита $\mathbb{H}$: $C^*(M) \in \mathbb{H} \subset \mathbb{B}^n$, $n > k$. Пусть $C \in \mathbb{B}^n$ – искаженный код, возникший при замене (инвертировании) некоторых бит исходного кода C^*. Вероятность искажения одного бита равна p_D (взята из предыдущего этапа (3.1)), искажения независимы. Далее в этом разделе для простоты переобозначим $p_D \to p$. Будем декодировать сообщение, предполагая, что исходный код искажен минимально, т.е. для C будем искать

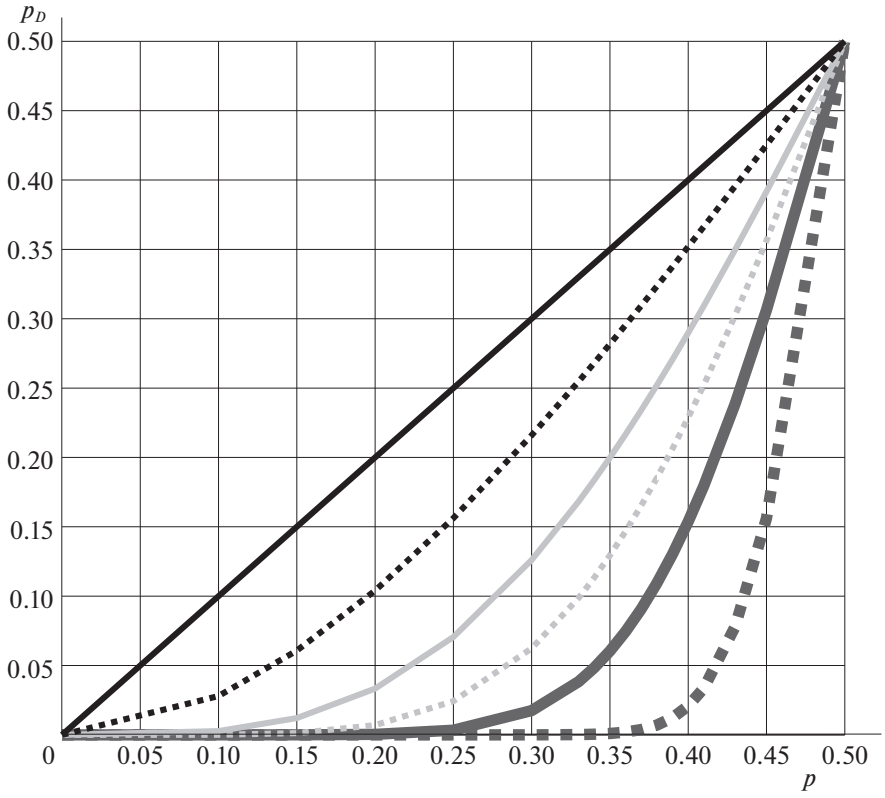

Рис. 4. Графики ошибок мажоритарного кодирования для значений $n = 1$ (прямая линия), 3, 7, 13, 25, 101

ближайший в смысле расстояния Хэмминга код из алфавита $\mathbb{H}$. Назовем его *аттрактором*. Аттракторов может быть несколько (несколько кодов могут иметь одинаковое минимальное расстояние до C). Если аттракторов несколько, то выбирается случайный. Обозначим множество аттракторов C как $A(C)$.

Известна простая оценка вероятности ошибки блокового кодирования Адамара, с длиной кодового слова n используемая в так называемой границе Хэмминга:

$$p_H \le 1 - P_{corr} = 1 - (1-p)^n \sum_{l=0}^{(n-1)/4} C_n^l \left(\frac{p}{1-p}\right)^l, \tag{3.2}$$

что совпадает с (3.1) за исключением верхнего предела суммирования.

Но это довольно грубая оценка, ухудшающаяся с ростом n. Найдем точную формулу. Вероятность правильного декодирования

$$P_{corr} = \sum_M P(M) \sum_C p(C^*|C)\, p(C|C^*),$$

где $P(M)$ – доля сообщений M во входном потоке (вероятность встретить сообщение M), $p(C|C^*)$ – вероятность получить искаженный код C при передаче сообщения M (которое кодируется кодом C^*), $p(C^*|C)$ – вероятность восстановить сообщение M из искаженного кода C. Считая все сообщения равновероятными, можно рассчитывать вероятность по одному из них, например нулевому ($M = 0$):

$$P_{corr} = \sum_C p(C^*|C)\, p(C|C^*).$$

Без ограничения общности в силу симметричности кода Адамара [26] можно считать, что C^* – нулевой код, т.е. последовательность нулевых бит соответствующей длины (так оно и есть при стан-

Таблица 2. Значения $H(b,a)$, $h(b)$ и $\overline{a}(b)$ для кода Адамара $k = 6$, $n = 31$

b	$H(b,1)$	$H(b,2)$	$H(b,3)$	$H(b,4)$	$H(b,6)$	$h(b)$	C_n^b	$\overline{a}(b)$
0	1	0	0	0	0	1	1	1.
1	31	0	0	0	0	31	31	1.
2	465	0	0	0	0	465	465	1.
3	4495	0	0	0	0	4495	4495	1.
4	31465	0	0	0	0	31465	31465	1.
5	169911	0	0	0	0	169911	169911	1.
6	736281	0	0	0	0	736281	736281	1.
7	2629575	0	0	0	0	2629575	2629575	1.
8	7291200	398040	0	0	0	7490220	7888725	0.974
9	11179840	5077800	238080	0	0	13798100	20160075	0.836
10	1833216	9114000	4999680	833280	5208	8265964	44352165	0.492
11	0	0	0	624960	1630104	427924	84672315	0.190

дартном кодировании). Тогда вероятность получить определенный код C из нулевого равна $p^{b(C)}(1-p)^{n-b(C)}$ и

$$P_{corr} = \sum_C p(0|C)p^{b(C)}(1-p)^{n-b(C)},$$

где $p(0 \mid C)$ – вероятность получить нулевой код из C:

$$p(0|C) = \begin{cases} 0, & \exists C' \in \mathbb{H}, \quad C' \neq 0\colon \rho(C',C) < \rho(0,C) = b(C), \\ 1/|A(C)|, & A(C) = \{C' \in \mathbb{H}\colon \rho(C',C) = b(C)\}. \end{cases} \tag{3.3}$$

Для небольших значений длины кода n можно построить гистограмму распределения точек пространства кодов $\mathbb{B}^n$ в зависимости от расстояния до нулевого кода $b = b(C)$ и количества аттракторов $a = |A(C)|$. Положим $a = 0$ в случае выполнения первого условия из (3.3). Тогда можно записать двумерное распределение

$$H(b,a) = \left|\{C : 0 \in A(C), b = b(C)\}\right|, \quad b \in [0;n], \quad a \in [1;2^k],$$

получить вероятность правильного декодирования

$$P_{corr} = \sum_{a\neq 0}\sum_b \frac{H(b,a)}{a} p^b(1-p)^{n-b} = \sum_b p^b(1-p)^{n-b}\sum_{a\neq 0}\frac{H(b,a)}{a}$$

и ошибку кодирования

$$p_H = 1 - P_{corr} = 1 - (1-p)^n \sum_b h(b)\left(\frac{p}{1-p}\right)^b, \quad h(b) = \sum_{a\neq 0}\frac{H(b,a)}{a}. \tag{3.4}$$

Запись совпадает с (3.2) за исключением используемых коэффициентов и пределов суммирования (в (3.4) суммирование производится по всем b). Таблица значений $H(b,a)$ и $h(b)$ для пополненного кода Адамара порядка 5 ($n = 2^5 - 1 = 31$, $k = 5 + 1 = 6$) дана ниже (табл. 2). Для сравнения приведены значения C_n^b. Также представлены средние значения вероятности выбора правильного аттрактора:

$$\overline{a}(b) = \frac{h(b)}{\sum_a H(b,a)}.$$

Для этого кода значения b изменяются от 0 до 31, приведены первые 12. Остальные значения нулевые, все последовательности, отличающиеся от нулевой более чем на 11 бит, “притягиваются” к другим кодовым векторам. Из таблицы видно, что вплоть до значений $b = 7$ происходит выбор

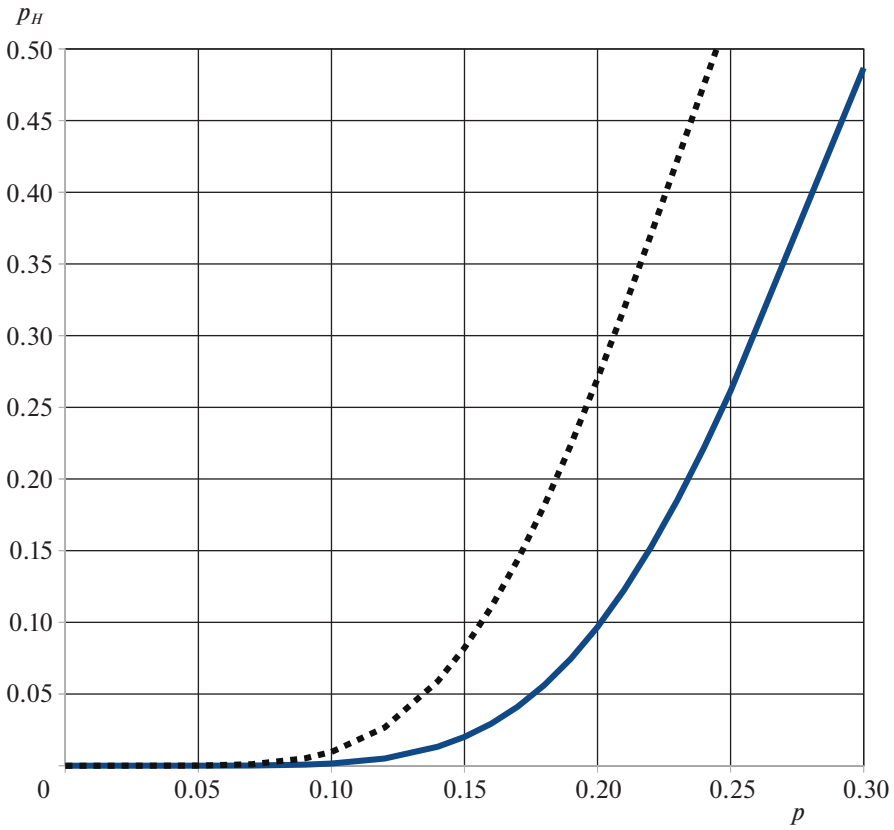

Рис. 5. Графики ошибки кодирования Адамара для $n = 31$, $k = 6$: оценка границей Хэмминга (3.2) (пунктирная линия) и точное значение (3.4) (сплошная линия)

лишь одного аттрактора, т.е. при таком или меньшем расхождении сообщение восстанавливается всегда. Это соответствует границе Хэмминга. Однако и при больших расхождениях существует значительная вероятность правильного восстановления. Это существенно, поскольку кодов за границей Хэмминга существует гораздо больше, чем внутри. Таким образом, оценка по Хэммингу вероятности ошибки декодирования сильно завышена. Например, для пополненного кода Адамара $k = 6$, $n = 31$ и ошибке $p = 0.250$ оценка по (3.2) дает $p_H = 0.527$, что, казалось бы, исключает использование такого кода. Однако расчет по формуле (3.4) дает $p_H = 0.261$, что вполне пригодно для использования на следующем шаге – коировании Рида–Соломона. На рис. 5 приведены графики функции (3.2) и (3.4) для $k = 6$, $n = 31$.

Параметр кодирования Адамара – длина слова k.

3.4. Кодирование Рида–Соломона. Единицей кодирования для алгоритма Рида–Соломона является все сообщение, которое разбивается на слова фиксированного размера, s бит каждое. Одно такое слово – запись целого числа в диапазоне $[0; 2^s - 1]$. Последовательность из L бит разбивается на $k = \lceil L/s \rceil$ слов, т.е. чисел. Эти числа трактуются как коэффициенты полинома над полем Галуа $GF(2^k)$. Оказывается, что если определенным образом расширить полином с k коэффициентами до полинома с $n > k$ коэффициентами, то можно среди полученных n коэффициентов произвольно исказить любые $t \le (n - k)/2$ из них и из полученной записи восстановить исходные коэффициенты.

Обозначим: L – длина сообщения (число кодируемых бит); s – размер слова кода Рида–Соломона (РС); k – число слов РС в сообщении, $k = \lceil L/s \rceil$; n – число слов РС в коде; N – длина кода (число бит в коде, полученном методом РС), $N = ns$; p – доля допустимых ошибочных слов в коде, при которой еще возможно восстановление сообщения. Код РС исправляет не более t

ошибок, где t равно половине избыточных слов в коде, добавленных к исходному сообщению, т.е. $n = k + 2t$, обозначив $p = t/n$, получим

$$p \le \frac{n-k}{2n}. \tag{3.5}$$

Вычисленная таким образом доля может служить оценкой допустимой вероятности ошибки слова кода. Также есть ограничение на максимальное число слов кода:

$$n \le 2^s - 1. \tag{3.6}$$

Зафиксировав длину сообщения L, выбрав длину слова s и зная вероятность ошибки декодирования предыдущего шага $p = p_H$, можно построить код РС. Вероятность p_H определяется извне, поэтому код РС параметризуется двумя величинами: размером слова s и длиной сообщения L.

3.5. Добавочная ошибка восстановления кода. Расстояние Хэмминга задает функцию зависимости ошибки первого рода от ошибки второго рода. Для некоторой базы данных и фиксированного порога θ рассмотрим количества различных исходов. Существует всего четыре исхода: истинный допуск (обозначим число таких событий “true positive” N_{TP}), ложный отказ (“false positive” N_{FR}), истинный отказ (“true negative” N_{TN}) и ложный отказ (“false negative” N_{FN}). Вероятности ошибки первого (false reject) и второго (false accept) рода определяются как

$$p_{FR} = \frac{N_{FN}}{N_{FN} + N_{TP}}, \quad p_{FA} = \frac{N_{FP}}{N_{FP} + N_{TN}}. \tag{3.7}$$

Следует отметить, что при работе с БД сумма $N_{FN} + N_{TP}$ постоянна и равна числу сравнений эталонов одного человека, а сумма $N_{FP} + N_{TN}$ постоянна и равна числу сравнений эталонов разных людей.

Восстановление сообщения из кода вносит дополнительную ошибку, оценим ее. Обозначим вероятность ошибки восстановления p_R. Соответственно вероятность правильного восстановления равна $1 - p_R$. Ожидаемое число событий истинного допуска в схеме с восстановлением уменьшается пропорционально этой вероятности: $N'_{TP} = N_{TP}(1 - p_R)$. Значит, увеличивается число ложных отказов: $N'_{FN} = N_{FN} + N_{TP}p_R$. Аналогично число ложных допусков уменьшается: $N'_{FP} = N_{FP}(1 - p_R)$, число истинных отказов увеличивается: $N'_{TN} = N_{TN} + N_{FP}p_R$. Рассчитывая новые вероятности ошибок первого и второго рода, получаем:

$$\begin{aligned} p'_{FR} &= \frac{N_{FN} + N_{TP}p_R}{N_{FN} + N_{TP}} = p_{FR} + \frac{N_{TP}}{N_{FN} + N_{TP}}p_R = p_{FR} + \left(1 - \frac{N_{FN}}{N_{FN} + N_{TP}}\right)p_R = \\ &= p_{FR} + (1 - N_{FR})p_R = p_{FR}(1 - p_R) + p_R, \\ p'_{FA} &= \frac{N_{FP}}{N_{FP} + N_{TN}}(1 - p_R) = p_{FA}(1 - p_R). \end{aligned} \tag{3.8}$$

Если $p_R \ll 1$, $p_{FR} \ll 1$, то можно приблизить: $p'_{FR} \approx p_{FR} + p_R$, $p'_{FA} \approx p_{FA}$. Таким образом, вероятность ошибки первого рода увеличивается на величину вероятности ошибки восстановления, вероятность ошибки второго рода можно считать неизменной.

4. Подбор параметров схемы кодирования. Описанные четыре последовательно исполняемых алгоритма имеют следующие параметры, влияющие на их характеристики: (1) псевдослучайное перемешивание не имеет таких параметров; (2) мажоритарное кодирование параметризуется кратностью повтора бита n; (3) кодирование Адамара – размером слова k, кодирование РС – размером слова s и длиной сообщения L. Различные комбинации значений (n, k, s, L) приводят к построению различных кодеков, имеющих разные отношения длины кода к длине сообщения. Можно формально записать зависимости ошибок первого и второго рода от этих параметров: $FRR(n, k, s, l)$ и $FFR(n, k, s, l)$. Фактически, эти значения находились экспериментально. Используемая биометрия [33] имеет размер $Z = 6656$ бит. Размер кода C не может быть больше, дублирование и маскирование недопустимы, так как делают тривиальным взлом такого кода. Также

следует учесть, что практически осмысленным является кодирование сообщения некоторой минимальной длины Y. Тогда можно записать задачу поиска оптимального кода:

$$\begin{gathered} FRR(n,k,s,L) \to \min, \\ s.t.\ FAR \le 10^{-4}, \quad C \le Z, \quad L \ge Y. \end{gathered} \tag{4.1}$$

Для $Y = 64$ решение найдено: $n = 13$, $k = 5$, $s = 5$, $L = 65$, $FRR = 10.4\%$. Такой размер считается достаточным для "обычных" пользовательских ключей. Для $Y = 128$ и выше решение данной задачи не получено. Следует отметить, что без псевдослучайного перемешивания (явным образом не участвующего в (4.1)) задача не решается и при $Y = 64$.

Заключение. В парадигме нечеткого экстрактора построен метод внедрения криптографического ключа в биометрию радужной оболочки глаза. Успешно внедряется ключ размером до 65 бит, для больших размеров решение не получено. Использование дополнительных шагов псевдослучайного перемешивания и мажоритарного битового кодирования позволило решить проблемы изменчивости и локальной коррелированности биометрических признаков. Метод протестирован на нескольких базах изображений радужной оболочки глаза. Требуется изучение криптостойкости представленного метода.

СПИСОК ЛИТЕРАТУРЫ

1. *Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В.* Основы криптографии. М.: Гелиос АРВ, 2005. 480 с.
2. *Чмора А.Л.* Маскировка ключа с помощью биометрии // Проблемы передачи информации. 2011. Т. 47. № 2. С. 28–143.
3. *Daugman J.G.* Information Theory and the IrisCode // IEEE Trans. Information Forensics and Security. 2016. V. 11. № 2. P. 400–409.
4. *Gong S., Boddeti V.N., Jain A.K.* On the Capacity of Face Representation // 2017. URL: https://arxiv.org/abs/1709.10433.
5. *Yankov M.P., Olsen M.A., Stegmann M.B., Christensen S.S., Forchhammer S.* Fingerprint Entropy and Identification Capacity Estimation Based on Pixel-Level Generative Modelling // IEEE Trans. Information Forensics and Security. 2020. V. 15. P. 56–65.
6. *Иванов А.И., Ложников П.С., Сулавко А.Е.* Оценка надежности верификации автографа на основе искусственных нейронных сетей, сетей многомерных функционалов Байеса и сетей квадратичных форм // Компьютерная оптика. 2017. V. 41. № 5. С. 765–774.
7. *Juels A., Sudan M.* A fuzzy vault scheme // Designs, Codes and Cryptography. 2006. V. 38. P. 237–257.
8. *Daugman J.* How Iris Recognition Works // Proc. Int. Conf. Image Processing. Lake Buena Vista, Orlando, USA, 2012. V. 1. P. 33–36.
9. *Shekar B.H., Bharathi R.K., Kittler J., Vizilter Y.V., Mestestskiy L.* Grid Structured Morphological Pattern Spectrum for Off-line Signature Verification // Proc. 2015 Int. Conf. Biometrics. Phuket, Thailand, 2015. V. 8. P. 430–435.
10. *Rathgeb C. Uhl A.* A Survey on Biometric Cryptosystems and Cancelable Biometrics // EURASIP J. Information Security. 2011. V. 3. P. 1–25.
11. *Майоров А.В.* Нейросетевая хеш-функция // Нейрокомпьютеры: разработка, применение. 2009. № 6. Р. 45–48.
12. *Иванов А.И., Сомкин С.А., Андреев Д.Ю., Малыгина Е.А.* О многообразии метрик, позволяющих наблюдать реальные статистики распределения биометрических данных "нечетких экстракторов" при их защите наложением гаммы // Вестн. УрФО. Безопасность в информационной сфере. 2014. V. 2. № 12. С. 16–23.
13. Защита информации. Техника защиты информации. Автоматическое обучение нейросетевых преобразователей биометрия-код доступа. ГОСТР 52633.5-2011. М.: Стандартинформ, 2012.
14. *Sutcu Y., Sencar H.T., Memon N.* A Secure Biometric Authentication Scheme Based on Robust Hashing // Proc. 7th Workshop Multimedia and Security. N.Y., USA, 2005. P. 111–116.
15. *Rathgeb C., Uhl A.* Privacy Preserving Key Generation for Iris Biometrics // Communications and Multimedia Security. Eds. *De Decker B., Schaumueller-Bichl I.* Berlin, Heidelberg: Springer, 2010. P. 191–200.
16. CASIA Iris Image Database, Institute of Automation, Chinese Academy of Sciences. 2010. http:// biometrics.idealtest.org/ findTotalDbByMode.do?mode=Iris.
17. *Davida G., Frankel Y., Matt B.* On the Relation of Error Correction and Cryptography to an Offline Biometric Based Identification Scheme // Proc. Workshop on Coding and Cryptography. Paris, France, 1999. P. 129–138.
18. *Dodis Y., Ostrovsky R., Reyzin L., Smith A.* Fuzzy extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data // SIAM J. Computing. 2008. V. 38. № 1. P. 97–139.

19. *Yang S., Verbauwhede I.* Secure Iris Verification // Proc IEEE Int. Conf. Acoustics, Speech and Signal Processing. Honolulu, USA, 2007. V. 2. P. 133–136.
20. *Linnartz J.-P., Tuyls P.* New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates // Proc. 4th Int. Conf. Audio- and Video-Based Biometric Person Authentication. Guildford, UK, 2003. P. 393–402.
21. *Shamir A.* How to share a secret // Commun. ACM. 1979. V. 22. № 11. P. 612–613.
22. *Lee Y.J., Bae K., Lee S.J., Park K.R., Kim J.* Biometric Key Binding: Fuzzy Vault Based on Iris Images // Proc. 2nd Int. Conf. Biometrics. Seoul, Korea, 2007. P. 800–808.
23. *Wu X., Qi N., Wang K., Zhang D.* An Iris Cryptosystem for Information Security // Proc. Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing. Harbin, China, 2008. P. 1533–1536.
24. *Juels A., Wattenberg M.* A Fauzzy Commitment Scheme // 6th ACM Conf. Computer and Communications Security. Singapore, 1999. P. 28–36.
25. *Hao F., Anderson R., Daugman J.* Combining Crypto with Biometrics Effectively // IEEE Trans. Computers. 2006. V. 55. № 9. P. 1081–1088.
26. *Морелос-Сарагоса Р.* Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. М.: Техносфера, 2005.
27. *Рубис А.Ю., Лебедев М.А., Визильтер Ю.В., Выголов О.В., Желтов С.Ю.* Компаративная фильтрация изображений с использованием монотонных морфологических операторов // Компьютерная оптика. 2018. V. 42. № 2. С. 306–311.
28. *Лебедев М.А., Рубис А.Ю., Визильтер Ю.В., Выголов О.В., Желтов С.Ю.* Выделение отличий на изображениях с помощью референтных EMD-фильтров // Математические методы распознавания образов. 2017. V. 18. № 1. С. 116–117.
29. *Reed I.S., Solomon G.* Polynomial Codes over Certain Finite Fields // J. Society for Industrial and Applied Mathematics. 1960. V. 8. № 2. P. 300–304.
30. *Kanade S., Camara D., Krichen E., Petrovska-Delacretaz D., Dorizzi B.* Three Factor Scheme for Biometric-based Cryptographic Key Regeneration Using Iris // Proc. Biometrics Symposium. Tampa, FL, USA, 2008. P. 59–64.
31. *Иванов А.И.* Нечеткие экстракторы: проблема использования в биометрии и криптографии // Первая миля. 2015. № 1. С. 40.
32. *Ганькин К.А., Гнеушев А.Н., Матвеев И.А.* Сегментация изображения радужки глаза, основанная на приближенных методах с последующими уточнениями // Изв. РАН. ТиСУ. 2014. № 2. С. 80–94.
33. *Novik V., Matveev I., Litvinchev I.* Enhancing Iris Template Matching with the Optimal Path Method // Wireless Networks. 2018. P. 1–8.
34. *Woodard D.L., Ricanek K.* Iris Databases. Eds: Li S.Z., Jain A. Encyclopedia of Biometrics. Boston, MA, USA: Springer, 2009.
35. *Phillips P., Scruggs W., O'Toole A. et al.* Frvt 2006 and Ice 2006 Large-scale Experimental Results // IEEE PAMI. 2010. V. 5. № 32. P. 831–846.
36. *Proenca H., Alexandre L.* UBIRIS: A Noisy Iris Image Database // 13th Int. Conf. Image Analysis and Processing. Cagliari, Italy, 2005. P. 970–977.
37. https://www.sovereignccctv.com/panasonic-bm-et300-iris-reader.html Дата обращения 13.04.2020.
38. https://www.sourcesecurity.com/lg-iris-irisaccess-3000-technical-details.html Дата обращения 13.04.2020.
39. 9 http://www.iritech.com/products/hardware/ Дата обращения 13.04.2020.
40. *Reed I.S.* A Class of Multiple-error-correcting Codes and the Decoding Scheme // Transactions IRE Professional Group on Information Theory. 1954. V. 4. № 4. P. 38–49.