УДК 004.934

Модификация алгоритма на основе сети Фейстеля с добавлением элемента случайности в ключ шифрования

П. К. Чернов, А. П. Шкарапута
Пермский государственный национальный исследовательский университет
Россия, 614990, г. Пермь, ул. Букирева, 15
ch3rn0vpk@gmail.com, shkaraputa@psu.ru

Изучен метод построения шифров на основе сети Фейстеля, определены его достоинства и недостатки. Предложен алгоритм на основе сети Фейстеля с добавлением элемента случайности в ключ шифрования. Проведен анализ основных параметров алгоритма относительно сети Фейстеля: криптостойкость, время выполнения, увеличение объема зашифрованных данных. В результате анализа выявлены повышенные по сравнению с сетью Фейстеля характеристики криптостойкости, увеличенное время выполнения и объем зашифрованных данных. Также сделан вывод о наличии у алгоритма потенциала в качестве основы для построения блочных шифров.

Ключевые слова: *криптография; сеть Фейстеля; коды Хэмминга; блочные шифры.*

DOI: 10.17072/1993-0550-2021-1-81-88

Введение

За всю историю развития криптографии было создано множество алгоритмов и методов. Среди них можно отметить метод "Сеть Фейстеля", созданный в 1970-х гг. метод построения блочных шифров, который получил широкое распространение и стал основой для создания множества блочных шифров, таких как DES, Blowfish, "Магма" и др. В силу этого метод был широко изучен криптоаналитиками, были выявлены его сильные и слабые стороны.

Таким образом, благодаря широкой распространенности метода, наличию множества базирующихся на нем алгоритмов, а также возможности их модификации и модернизации можно говорить о том, что направление разработки и изучения алгоритмов на основе "Сети Фейстеля" является актуальным и в настоящее время.

В данной работе рассматриваются криптографические свойства сети Фейстеля, предлагается вариант модифицированного алгоритма с внесением элемента случайности и применением кодов Хэмминга, а также проводится анализ параметров алгоритма.

На основе результатов анализа делаются выводы о сильных и слабых сторонах предложенного алгоритма, а также о возможности его использования для конструирования блочных шифров.

1. Сеть Фейстеля

Обобщенная сеть Фейстеля – один из методов построения блочных шифров. Метод назван по имени Хорста Фейстеля – исследователя, работавшего в свое время в IBM, и одного из авторов стандарта DES [1].

Сеть Фейстеля подразумевает разбиение обрабатываемого блока данных на несколько субблоков (чаще всего – на два), один из которых обрабатывается некоей функцией f и накладывается на один или несколько субблоков [2].

На рис. 1 приведена наиболее часто встречающаяся структура алгоритмов на основе сети Фейстеля:

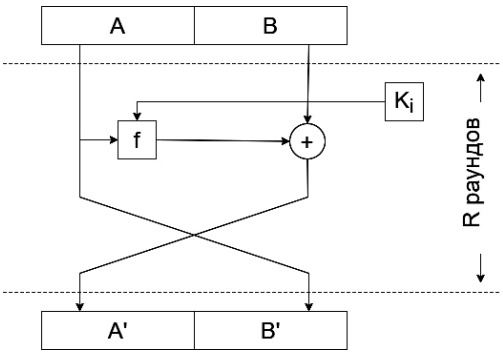

Рис. 1. *Сеть Фейстеля*

Дополнительный аргумент функции f, обозначенный на рис. 1 как K_i, называется *ключом раунда.* Ключ раунда является результатом обработки ключа шифрования процедурой расширения ключа, задача которой – получение необходимого количества ключей K_i из исходного ключа шифрования.

В простейших случаях процедура расширения ключа просто разбивает ключ на несколько фрагментов, которые поочередно используются в раундах шифрования; существенно чаще процедура расширения ключа является достаточно сложной, а ключи K_i зависят от значений большинства битов исходного ключа шифрования.

Наложение обработанного субблока на необработанный чаще всего выполняется с помощью логической операции *исключающее* или (Exclusive OR, XOR), как показано на рис. 1. Достаточно часто вместо XOR здесь используется сложение по модулю 2^n, где n – размер субблока в битах. После наложения субблоки меняются местами, т.е. в следующем раунде алгоритма обрабатывается уже другой субблок данных.

1.1. Достоинства сети Фейстеля

У данного метода построения шифров выделяют следующие достоинства:

- Простота аппаратной реализации и возможность использования на платформах с ограниченными ресурсами [2].
- Простота программной реализации в силу того, что все значительная часть функций поддерживается на аппаратном уровне в современных компьютерах (например, сложение по модулю 2).
- Процедуры шифрования и дешифрования совпадают, с тем исключением, что ключевая информация при дешифровании используется в обратном порядке.
- Для построения устройств шифрования можно использовать те же блоки в цепях шифрования и дешифрования.
- Хорошая изученность алгоритмов на основе сетей Фейстеля [3].
- Возможность распараллеливания вычислений, поскольку блоки открытого текста не связаны между собой.

1.2. Недостатки сети Фейстеля

В то же время обобщенная сеть Фейстеля обладает следующими недостатками:

- На каждой итерации изменяется только половина блока обрабатываемого текста, что приводит к необходимости увеличивать число итераций для достижения требуемой стойкости [1].
- Уязвимость к частотному криптоанализу при небольших размерах блока [4].
- Уязвимость к "сдвиговой атаке" при использовании одинаковых или повторяющихся раундовых ключей [2, 5].

В отношении выбора F-функции каких-то четких стандартов не существует, однако, как правило, эта функция представляет собой последовательность зависящих от ключа нелинейных замен, перемешивающих перестановок и сдвигов [1].

2. Описание алгоритма

Модификация подразумевает число раундов кратное четырем. Структура раундов шифрования и дешифрования представляет собой сеть Фейстеля.

2.1. Алгоритм шифрования

Алгоритм принимает на вход блок данных размером N бит и использует для шифрования ключ длины L бит, где L=N/2.

Структура раундов шифрования представляет собой сеть Фейстеля. Функция F включает:

- Побитовое сложение по модулю два (XOR) половины шифруемого блока с раундовым ключом.
- Преобразование S (применение S-блока).

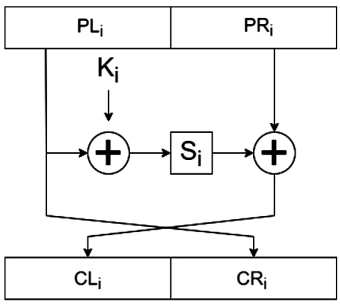

Рис. 2. *Схема i-го раунда шифрования*

Операции i-го раунда шифрования (рис. 2):

1. Сложение по модулю два (XOR) левой половины шифруемого блока с раундовым ключом.
2. S-преобразование.
3. Сложение по модулю два (XOR) с правой половиной шифруемого блока.
4. Конкатенация полученного на предыдущем шаге результата с левой половиной шифруемого блока.

На последнем раунде половины блока не меняются местами.

Модификация подразумевает число раундов кратное четырем. В простейшем варианте исходный блок данных проходит четыре раунда шифрования. На каждом раунде применяется соответствующий раундовый ключ K_i, i=1…4.

Для получения раундовых ключей K_1 и K_2 исходный ключ K длины L расширяется путем добавления контрольных n бит на основе кодов Хэмминга. Обозначим полученный ключ K'. Раундовые ключи K_1 и K_2 представляют собой первые и последние L бит ключа K' соответственно (рис. 3).

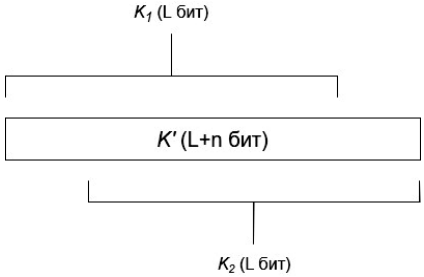

Рис. 3. *Образование ключей K_1 и K_2*

Для получения раундовых ключей K_3 и K_4 случайно выбранный бит исходного ключа K инвертируется. После этого ключ с инвертированным битом расширяется с помощью добавления n контрольных бит на основе кодов Хэмминга.

Обозначим полученный таким образом ключ K*. Раундовые ключи K_3 и K_4 представляют собой первые и последние L бит ключа K* соответственно (рис. 4).

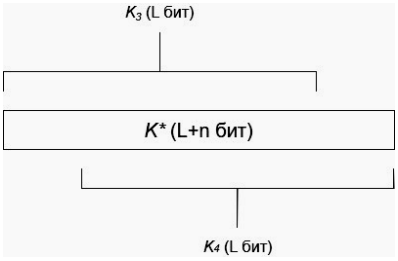

Рис. 4. *Образование ключей K_3 и K_4*

Полученные при формировании ключа K* контрольные биты (n бит) дописываются в конец полученного зашифрованного блока для возможности его дешифрования. Таким образом, итоговая длина зашифрованного блока составляет N+n бит.

2.2. Алгоритм дешифрования Сампл

Алгоритм дешифрования получает на вход зашифрованный блок данных размером N+n бит и использует тот же ключ K длины L бит, где L=N/2. Последние n бит зашифрованного блока используются только для восстановления раундовых ключей, при прохождении блоков раундов дешифрования они отбрасываются. В простейшем варианте зашифрованный блок данных проходит четыре раунда дешифрования.

Операции i-го раунда дешифрования (рис. 5):

1. Сложение по модулю два (XOR) левой половины дешифруемого блока с раундовым ключом.
2. S-преобразование.
3. Сложение по модулю два (XOR) с правой половиной дешифруемого блока.
4. Конкатенация полученного на предыдущем шаге результата с левой половиной дешифруемого блока

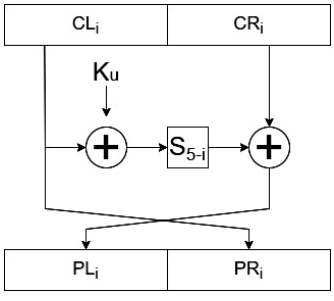

Рис. 5. *Схема i-го раунда дешифрования*

В данном случае номер раундового ключа, применяемого на *i*-м раунде шифрования, определяется формулой:

$$u = \frac{5i-14}{2i-5} \ . \qquad (1)$$

На последнем раунде половины блока не меняются местами. Количество раундов дешифрования совпадает с количеством раундов шифрования для данного блока. При дешифровании необходимо использовать те же раундовые S-блоки, которые применялись при шифровании, но в обратном порядке.

Для получения раундовых ключей, которые будут применяться на первых двух раундах дешифрования (ключи K_3 и K_4 соответственно) исходный ключ K расширяется согласно алгоритму кодов Хэмминга, но в качестве контрольных бит используются последние n бит дешифруемого блока. Далее для полученной последовательности применяется алгоритм исправления однократной ошибки кодов Хэмминга, в результате чего восстанавливается ключ K*, применявшийся для шифрования данного блока. Раундовые ключи K_3 и K_4 представляют собой первые и последние L бит ключа K* соответственно (рис. 4).

Для получения раундовых ключей, которые будут применяться на третьем и четвертом раундах дешифрования (ключи K_1 и K_2 соответственно), исходный ключ K длины L расширяется путем добавления контрольных n бит на основе кодов Хэмминга до ключа K'. Раундовые ключи K_1 и K_2 представляют собой первые и последние L бит ключа K' соответственно (рис. 3).

3. Анализ алгоритма

Для проведения анализа применялся описанный выше алгоритм со следующими параметрами:

- входной блок: 128 бит;
- ключ шифрования: 64 бит;
- количество раундов шифрования: в зависимости от исследования;
- размер зашифрованного блока: 135 бит.

В качестве таблиц постановок для S-блоков применяются первые четыре таблицы, применяемые в блочном алгоритме шифрования "Магма" в соответствии с ГОСТ 34.12-2018 [6].

Номер S-блока	Значение															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	C	4	6	2	A	5	B	9	E	8	D	7	0	3	F	1
2	6	8	2	3	9	A	5	C	1	E	4	7	B	D	0	F
3	B	3	5	8	2	F	A	D	E	1	7	4	C	9	6	0
4	C	8	2	1	D	4	F	6	7	0	A	5	3	E	9	B

Рис. 6. *Первые четыре таблицы подстановок в ГОСТ 34.12-2018*

Сравнение проводилось с результатами работы алгоритма, схемы раундов шифрования и дешифрования которого совпадают со схемами описанного ранее модифицированного алгоритма за исключением того, что все раундовые ключи одинаковы и совпадают с исходным ключом шифрования. Далее этот алгоритм будет обозначаться как "базовый".

Измерения проводились в системе со следующими характеристиками:

- оперативная память – 4 Гб;
- количество ядер процессора – 1;
- тактовая частота процессора – 2.9 ГГц.

3.1. Анализ времени выполнения

Было измерено время выполнения алгоритмов шифрования для базового алгоритма и модифицированного алгоритма с различным количеством раундов. В каждом исследовании измерялось время выполнения 10 процессов шифрования. Для каждого варианта алгоритма и количества раундов проводилось 100 измерений. При измерениях фиксировалось минимальное, максимальное и среднее время работы алгоритма.

Таблица 1. *Результаты измерений времени выполнения алгоритмов шифрования*

		Количество раундов		
		4	8	12
Базовый	$T_{мин}$, сек	0.007022142	0.015047789	0.023073435
	$T_{макс}$, сек	0.012038469	0.071227312	0.034108877
	$T_{ср}$, сек	0.008035631	0.016813633	0.024347665
Модифицированный	$T_{мин}$, сек	0.013041258	0.020028591	0.030043125
	$T_{макс}$, сек	0.017054558	0.045073748	0.046129227
	$T_{ср}$, сек	0.014215343	0.025840847	0.037316928

График на рис. 7 отражает соотношение средних значений времени шифрования для различного числа раундов базового и модифицированного алгоритмов (табл. 1):

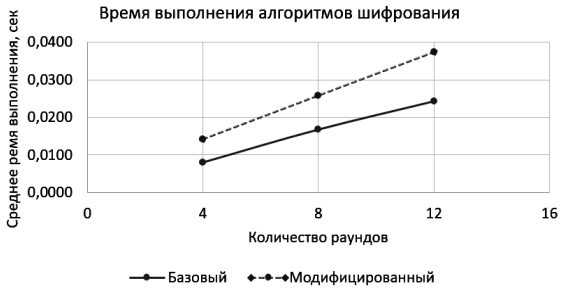

Рис. 7. *График средних значений времени выполнения алгоритмов шифрования*

Результаты измерений показывают, что время шифрования для модифицированного алгоритма в среднем в 1.61 раз больше, чем для базового.

Аналогичные измерения были проведены и для операций дешифрования.

Таблица 2. *Результаты измерений времени выполнения алгоритмов дешифрования*

		Количество раундов		
		4	8	12
Базовый	$T_{мин}$, *сек*	0.007022142	0.015047789	0.011017323
	$T_{макс}$, *сек*	0.021032333	0.024076939	0.043067455
	$T_{ср}$, *сек*	0.008624878	0.016783535	0.024371953
Модифицированный	$T_{мин}$, *сек*	0.013041258	0.024076700	0.035111904
	$T_{макс}$, *сек*	0.017054319	0.028089762	0.039124727
	$T_{ср}$, *сек*	0.013804030	0.024829199	0.035663407

График на рис. 8 отражает соотношение средних значений времени дешифрования для различного числа раундов базового и модифицированного алгоритмов (табл. 2).

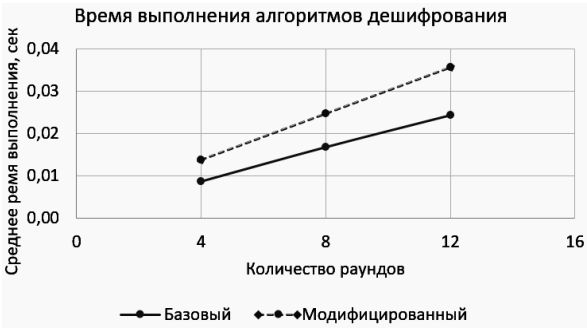

Рис. 8. *График средних значений времени выполнения алгоритмов дешифрования*

Результаты измерений показывают, что время дешифрования для модифицированного алгоритма в среднем в 1.51 раз больше, чем для базового.

3.2. Анализ увеличения объема зашифрованного блока

Хэммингом было выявлено, что минимальное число бит для кодирования блока данных вычисляется в соответствии со следующим неравенством:

$$k \geq 2^k - m - 1, \qquad (2)$$

где k – количество контрольных бит, m – длина исходного блока [7].

В случае шифрования 128-битных блоков увеличение размеров зашифрованного блока для различного числа раундов шифрования будет следующим (рис. 9, табл. 3):

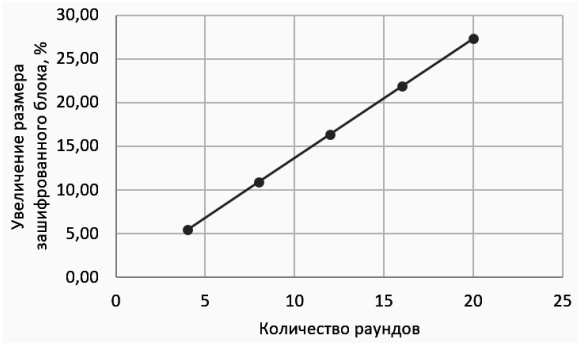

Рис. 9. *График увеличения размера блока шифротекста*

Таблица 3. *Увеличения размера блока шифротекста*

Кол-во раундов	Увел-е размера, %
4	5,46875
8	10,9375
12	16,40625
16	21,875
20	27,34375

3.3. Анализ информационной энтропии

Энтропия – мера неопределенности некоторого опыта, исход которого зависит от выбора одного элемента из множества исходных [8]. Об энтропии можно говорить как о количественной мере неопределенности появления на выходе источника сообщений буквы первичного алфавита. При анализе в качестве исходного алфавита выступало множество различных 8-битных блоков, встречающихся в исследуемом тексте.

Для исследования использовался файл, содержащий 744 блока по 128 бит. Значение информационной энтропии данного текста составило 4,481584641169272 бит.

Данный файл был зашифрован тремя вариантами базового алгоритма (с 4, 8 и 12 раундами шифрования) и для каждого полученного шифротекста было вычислено значение информационной энтропии. Затем исходный текст шифровался с применением трех вариантов модифицированного алгоритма (с 4, 8 и 12 раундами шифрования). Для каждого варианта шифрование и вычисление значения информационной энтропии было произведено 1000 раз, после чего определялось среднее значение информационной энтропии. Результаты исследования представлены в следующей табл. 4.

Таблица 4. *Значения энтропии шифротекста для разных вариантов алгоритмов*

Кол-во раундов	Базовый	Модифицированный
4	7.86222388280093	7.978919180625732
8	7.88038719573999	7.9803688031014355
12	7.894317989627951	7.984753003604013

Таким образом, результаты исследования показывают, что значение информационной энтропии шифротекста прямо пропорционально количеству раундов шифрования. Данная характеристика выше у модифицированного алгоритма, что обусловлено наличием в модифицированном алгоритме элемента случайности, повышающего неопределенность появления символа шифротекста.

3.4. Устойчивость к сдвиговой атаке

Описанный выше базовый алгоритм полностью соответствует параметрам алгоритма, уязвимого к простейшему случаю сдвиговой атаки, ввиду идентичности раундов, а также применения одного и того же ключа во всех раундах.

Модифицированный алгоритм можно свести к простейшему случаю сдвиговой атаки. Для этого набор раундовых ключей K_1, K_2, K_3, K_4 объявляется единым раундовым ключом, а последовательность этапов шифрования, использующих этот набор, считается отдельным раундом (рис. 10).

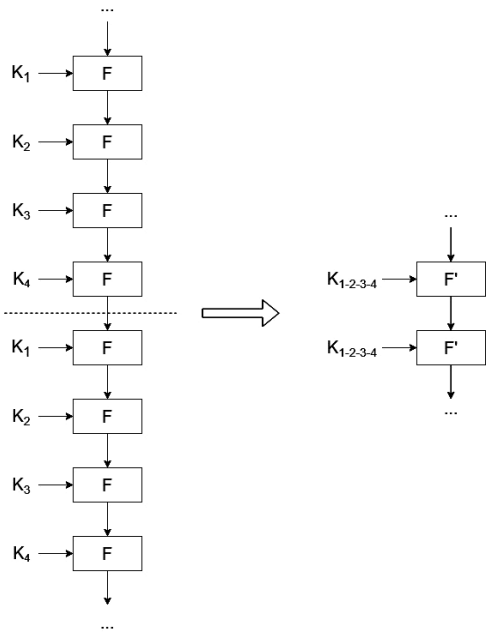

Рис. 10. *Схема объединения раундов для модифицированного алгоритма*

Однако модифицированный алгоритм все равно не будет уязвим к сдвиговой атаке, так как за счет внесения элемента случайности – инвертирования случайного бита исходного ключа шифрования при формировании раундовых ключей K_3 и K_4 – вероятность того, что объединяемые в наборы раундовые ключи K_1, K_2, K_3, K_4 на различных этапах будут идентичны, оказывается равной 1/L, где L – размер ключа шифрования.

3.5. Устойчивость к линейному и дифференциальному криптоанализу

Внесение элемента случайности приводит к тому, что одному исходному тексту может соответствовать множество шифротекстов. По этой причине методы линейного и дифференциального криптоанализа, работающие с парами открытых и закрытых текстов, должны быть менее эффективны в отношении подобных алгоритмов шифрования. К подобному выводу приходит и С.А. Демин в своей работе "Вероятностное шифрование" [9] при изучении алгоритма с внесением элемента случайности в блок данных в процессе шифрования.

Таким образом, можно говорить о том, что предложенный алгоритм является более устойчивым к линейному и дифференциальному анализу по сравнению с классической сетью Фейстеля.

Заключение

По результатам проведенного исследования можно сделать выводы о достоинствах и недостатках разработанного алгоритма по сравнению с классической сетью Фейстеля.

Достоинства:

1. Повышенная стойкость алгоритма шифрования к линейному и дифференциальному криптоанализу.
2. Лучшее значение информационной энтропии зашифрованного текста.
3. Устойчивость к "Сдвиговой атаке".
4. Устойчивость к частотному криптоанализу зашифрованного текста при сохранении возможности распараллеливания процессов шифрования отдельных блоков
5. Повышенная устойчивость алгоритма к атаке грубой силы.

Недостатки:

1. Повышенная сложность реализации ввиду необходимости применения ГПСЧ.
2. Повышенное время шифрования и дешифрования.
3. Увеличение размера зашифрованного блока.

Стоит отметить, что помимо изменения числа раундов, криптостойкость алгоритма можно повысить и иным способом: можно применять внесение элемента случайности – инвертирование случайного бита ключа шифрования – при формировании каждого раундового ключа, а не только двух из четырех. Это, однако, приведет к еще большему повышению времени шифрования и дешифрования, а также к еще большему увеличению размера зашифрованного блока. Вдобавок к этому использованные в предложенной схеме алгоритма преобразования S (S блоки) могут использовать иные таблицы или же могут быть заменены на другие функции, что также окажет влияние на криптостойкость алгоритма.

В статье "Модификация алгоритмов на основе сети Фейстеля посредством внесения избыточности с помощью кодов Хэмминга" [10] описывается модификация алгоритма на основе сети Фейстеля с использованием кодов Хэмминга и элемента случайности, а также проводится анализ основных характеристик алгоритма относительно классической сети Фейстеля. Основой описываемой модификации алгоритма является внесение в шифруемый блок данных случайной ошибки, исправление которой возможно при дешифровании блока благодаря проверочным битам кодов Хэмминга. В результате проведенного анализа авторы статьи приходят к выводу, что предложенная ими модификация алгоритма:

- устойчива к атаке грубой силы и частотному анализу;
- обладает лавинным эффектом;
- уступает классической сети Фейстеля во времени шифрования и дешифрования в среднем в 2 раза;
- увеличивает объем зашифрованного файла в среднем в 1,21 раз при 8 раундах шифрования.

В сравнении с этими параметрами модификация алгоритма с добавлением элемента случайности в ключ шифрования при аналогичном размере блока и при том же количестве раундов:

- обладает меньшими потерями во времени шифрования и дешифрования;
- имеет меньший прирост размера зашифрованных данных;
- обладает более широкими возможностями для распараллеливания.

С учетом описанных достоинств, недостатков и потенциала для модификации представленная схема алгоритма может стать основой для построения блочных шифров.

Список литературы

1. *Баричев С.Г., Гончаров В.В., Серов Р.Е.* Основы современной криптографии. М.: Горячая линия – Телеком, 2001. С. 8–9.
2. *Панасенко С.* Алгоритмы шифрования. Специальный справочник. СПб.: БХВ-Петербург, 2009. С. 9.
3. *Кучерик А.О., Лексин А.Ю., Бухаров Д.Н., Шагурина А.Ю.* Курс лекций по дисциплине "Защита информации". Владимир: Изд-во ВлГУ, 2017. 104 с.
4. *Лапонина О.Р.* Основы сетевой безопасности: криптографические алгоритмы и протоколы взаимодействия. М.: Бином. 2-е изд. 2007. С. 531.

5. *Сборник* статей III Всерос. науч.-техн. конф. молодых ученых, аспирантов и студентов. Ростов-на/Дону: Изд-во Южного федерального ун-та, 2017. С. 24.
6. *ГОСТ* 34.12-2018. М.: Стандартинформ, 2018. 16 с.
7. *Поисов Д.А.* Коды Хемминга // Все о Hi-Tech, 2010.
8. *Цымбал В.П.* Теория информации и кодирования. К.: Издательское объединение "Вища школа", 4-е изд. 1992. 263 с.
9. *Демин С.А.* Вероятностное шифрование // Вестник Балтийского федерального университета им. И. Канта. Серия: Физико-математические и технические науки, 2005. Вып. 1–2. С. 107–110.
10. *Александрова Е.И., Шкарапута А.П.* Модификация алгоритмов на основе сети Фейстеля посредством внесения избыточности с помощью кодов Хэмминга // Вестник Пермского университета: Математика. Механика. Информатика, 2018. Вып. 3(42). С. 95–103.

Modification of the algorithm based on the Feistel network by adding an element of randomness into the encryption key

P. K. Chernov, A. P. Shkaraputa
Perm State University; 15, Bukireva st., Perm, 614990, Russia
ch3rn0vpk@gmail.com, shkaraputa@psu.ru

The article revealed the research of methods for constructing block ciphers and its advantages and disadvantages. The modified algorithm based on the Feistel network using Hamming codes and adding an element of randomness into the encryption key was proposed. Analysis of the main parameters of the algorithm in comparison with Feistel network was performed: resistance to cryptanalysis, execution time, increase in the volume of encrypted data. The analysis revealed the stronger resistance to cryptanalysis than the Feistel network, increased execution time and volume of encrypted data. The potential for building block ciphers based on the algorithm was explored.

Keywords: *cryptography; Feistel network; Hamming codes; block ciphers.*