УДК. 681.3.06:519.248.681

В. И. ДОЛГОВ, д-р. техн. наук, И. В. РУБАН, канд. техн. наук, С. В. ДУДЕНКО

ПОСТРОЕНИЕ НЕЛИНЕЙНЫХ СИСТЕМ НА ОСНОВЕ УСЕЧЕННОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ В КОНЕЧНЫХ ПОЛЯХ

В настоящий момент нелинейные преобразования в полях являются отдельной областью науки, представляющей не только теоретический, но и практический интерес. Они находят широкое применение в теории помехоустойчивого кодирования, в системах обработки информации и в последнее время широко используются при разработке криптографических протоколов.

Унитарные преобразования в полях относятся к классу теоретико-числовых преобразований, к этому классу можно отнести и преобразование Фурье в конечном поле. Следует заметить, что преобразование Фурье вектора длины n в поле $GF(q)$ существует только тогда, когда поле $GF(q)$ содержит элемент порядка n. Данный класс преобразований обладает свойством линейности. Интересным является тот факт, что при изменении порядка выполнения операций в теоретико-числовых преобразованиях связь между входным и выходным вектором становится нелинейной.

Представляется, что определенный научный и практический интерес может вызвать преобразование векторов, имеющих чётную длину, так как при этом повышается эффективность операций, выполняемых на современных микропроцессорах.

Вводится в рассмотрение усеченное преобразование Фурье (УПФ), которое в отличие от традиционного, существующего только для векторов длины n над полем Галуа, существует для векторов длины $n-1$. Данное преобразование при вычислениях в поле, являющемся расширением двоичного поля, имеет чётную длину всех входных векторов.

1 Преобразование Фурье над полем Галуа *GF (q)*

Для того чтобы ввести математические обозначения и облегчить восприятие материала, изложенного в статье, напомним определение и свойства преобразования Фурье в конечном поле, введенные Р. Блейхутом [1].

Определение. Пусть $v = \{v_i,\ i = 0, \ldots, n-1\}$ – вектор над $GF(q)$, где n делит $q^m - 1$ при некотором m, и пусть w – элемент порядка n в поле $GF(q^m)$. Преобразование Фурье в поле Галуа вектора v определяется как вектор $V = \{V_j,\ j = 0, \ldots, n-1\}$, задаваемый равенствами

$$V_j = \sum_{i=0}^{n-1} w^{ij} \cdot v_i, \qquad j = 0, \ldots, n-1,$$

где v_i – i-я точка входного вектора; V_j – j-я точка выходного вектора.

В качестве длины преобразования Фурье можно выбрать произвольный делитель числа $q^m - 1$, но наиболее важную роль играют примитивные длины $n = q^m - 1$. В последнем случае w является примитивным элементом поля $GF(q^m)$.

В [1] также приведена и доказана следующая теорема.

Над полем GF (q) характеристики p вектор и его спектр связаны соотношениями

$$V_j = \sum_{i=0}^{n-1} w^{ij} \cdot v_i, \qquad v_i = \left(1/n\right) \sum_{j=0}^{n-1} w^{-ij} \cdot V_j, \tag{1}$$

где n интерпретируется как число поля, т. е. по модулю p.

Например, в поле $GF(2^8)$ преобразование Фурье существует для $n = 3,5,15,17,51,85,255$. Для многих целей таких длин векторов оказывается достаточно.

Отметим два свойства преобразования Фурье, которые накладывают ограничения на его использование.

Первое заключается в том, что при проведении вычислений в поле, являющемся расширением двоичного поля, на вход преобразования Фурье мы можем подавать только векторы, имеющие нечётную длину. Любое число в формате ЭВМ может быть представлено в виде суммы степеней двойки. В связи с этим наибольший интерес вызывают именно преобразования в полях, являющихся расширением двоичного поля. Используемые же в современных ЭВМ микропроцессоры имеют разрядность 8, 32 и 64. Поэтому это свойство при программной реализации преобразования Фурье является недостатком, так как мы не можем в некоторых случаях полностью оптимизировать работу программы под аппаратную платформу. Особенно актуально вопрос о длине блоков, которые требуется преобразовать, стоит перед разработчиками криптографических протоколов, поскольку на современном уровне развития вычислительной техники система, стойкая к атакам криптоаналитика, должна поддерживать размер блока базовой функции шифрования, равный 128, 192, 256 бит.

Второе свойство – линейная связь входного вектора с выходным – является недостатком при использовании преобразования Фурье в криптографических целях. Так, например, в [2] Шнорр предлагает структуру хэш-функции, основанную на преобразовании Фурье в поле *GF* (65537), порядок которого есть простое число Ферма. Известно, что если результат сложения и умножения чисел не превышает значения модуля в таких полях, то соответствующие операции оказываются линейными. В данном случае линейность преобразования Фурье дала возможность доказать нестойкость такой конструкции к коллизиям [3].

Цель настоящей работы заключается в устранении указанных недостатков.

2 Способ достижения нелинейности теоретико-числовых преобразований

Идея способа может быть пояснена на примере достижения нелинейности преобразования Фурье. Обязательным условием является вычисления в поле, являющемся расширением двоичного поля.

Пусть имеется входной вектор

$$\boldsymbol{v} = \{v_0, v_1, \ldots, v_{n-1}\},$$

где $v_i \in GF(2^m)$ и w- элемент порядка n в $GF(2^m)$ при условии, что $n|(2^m-1)$.

Тогда первая точка выходного вектора может быть записана как

$$V_0 = OS\,[S(w^0 \otimes v_0) \oplus S(w^0 \otimes v_1) \oplus \ldots \oplus S(w^0 \otimes v_{n-1})], \qquad (2)$$

где $\otimes$ – операция умножения в поле, которая является линейной;

$\oplus$ – операция побитного ИСКЛЮЧАЮЩЕГО ИЛИ, которая также является линейной;

$S(v_i)$ – переход от десятичного представления элемента поля к двоичному представлению;

$OS(v_i)$ – переход от двоичного представления элемента поля к десятичному представлению.

Переходы $S(v_i)$ и $OS(v_i)$ являются нелинейными, однако в связи с тем, что они биективны (т.е. взаимообратны) и в выражении (2) применяются последовательно один за другим, то связь входного и выходного векторов оказывается линейной.

Суть достижения нелинейности состоит в изменении порядка выполнения операций в выражении (1). При этом первой точкой выходного вектора считается точка

$$V_0 = S(w^0 \otimes v_0) \oplus S(w^0 \otimes v_1) \oplus \ldots \oplus S(w^0 \otimes v_{n-1}), \qquad (3)$$

а переход $OS(v_i)$ осуществляется только при обратном преобразовании. В этом случае связь входного и выходного векторов для преобразования Фурье нельзя будет описать линейной

функцией в силу нелинейности перехода $S(\nu_i)$. Приведенные результаты говорят о возможности достижения нелинейности преобразования Фурье, однако ограничение на длины входных векторов остается в силе.

В третьем разделе рассмотрим унитарное преобразование, которое в полях характеристики два позволяет работать с векторами, имеющими чётную длину.

3 Усеченное преобразование Фурье (УПФ) над полем Галуа *GF (q)*

Прежде всего докажем теорему о существовании усеченного унитарного преобразования в конечном поле. Доказательство строится на основе выражения (1) путем удаления из традиционного преобразования Фурье нулевой компоненты (точки) входного вектора. Оно далее названо усеченным преобразованием Фурье. Обозначим символом $\Phi_l(\nu)$ – операцию усеченного преобразования входного вектора ν длины l. Результатом преобразования есть вектор длины l, который будем называть выходным вектором.

Теорема. *Над полем GF (q) характеристики* p *существует унитарное преобразование вида:*

$$V_j = \sum_{i=1}^{n-1} w^{ij} \cdot \nu_i, \tag{4}$$

$$\nu_i = \left(\frac{1}{n \bmod p}\right) \sum_{j=1}^{n-1} \left(w^{-ij} \oplus L\right) \cdot V_j, \tag{5}$$

где w *– элемент порядка* n *в поле GF (q);*
$\oplus$ – означает операцию сложения в поле; $L = -1$.

Доказательство. В поле *GF (q)* справедливо $\sum_{i=0}^{n-1} w^{ir} = 1 \oplus \sum_{i=1}^{n-1} w^{ir} = \begin{cases} n \bmod p, & \text{если } r = 0, \\ 0, & \text{если } r \neq 0 \end{cases}$

и следовательно $\sum_{i=1}^{n-1} w^{ir} = \begin{cases} n \bmod p \oplus L, & \text{если } r = 0, \\ L, & \text{если } r \neq 0. \end{cases}$ (6)

Подставив выражение (4) в (5), получим

$$\nu_i = \left(\frac{1}{n \bmod p}\right) \sum_{j=1}^{n-1} \left[\left(w^{-ij} \oplus L\right) \cdot \sum_{k=1}^{n-1} w^{jk} \cdot \nu_k \right] =$$

$$= \left(\frac{1}{n \bmod p}\right) \cdot \sum_{j,k=1}^{n-1} \left[\nu_k \cdot w^{jk} \cdot \left(w^{-ij} \oplus L\right) \right] = \left(\frac{1}{n \bmod p}\right) \cdot \sum_{j,k=1}^{n-1} \nu_k \cdot \left(w^{(k-i)j} \oplus w^{jk} \cdot L\right) =$$

$$= \left(\frac{1}{n \bmod p}\right) \cdot \left[\sum_{\substack{j=1 \\ k=i}}^{n-1} \nu_i \cdot \left(w^{(k-i)j} \oplus w^{jk} \cdot L\right) \oplus \sum_{\substack{j=1 \\ k \neq i}}^{n-1} \sum_{k=1}^{n-1} \nu_k \cdot \left(w^{(k-i)j} \oplus w^{jk} \cdot L\right) \right].$$

Воспользовавшись теперь соотношением (6), можем прийти к результату

$$\left(\frac{1}{n \bmod p}\right)\cdot\left[\sum_{\substack{j=1\\k=i}}^{n-1} v_i\cdot\left(w^{(k-i)j}\oplus w^{jk}\cdot L\right)\oplus\sum_{\substack{j=1\\k\neq i}}^{n-1}\sum_{k=1}^{n-1} v_k\cdot\left(w^{(k-i)j}\oplus w^{jk}\cdot L\right)\right]=$$

$$=\left(\frac{1}{n \bmod p}\right)\cdot\left[v_i\cdot\left[\sum_{j=1}^{n-1} w^{(k-i)j}\oplus L\cdot\sum_{j=1}^{n-1} w^{jk}\right]\oplus\sum_{\substack{k=1\\k\neq i}}^{n-1} v_k\cdot\left(\sum_{j=1}^{n-1} w^{(k-i)j}\oplus\sum_{j=1}^{n-1} w^{jk}\cdot L\right)\right]=$$

$$=\left(\frac{1}{n \bmod p}\right)\cdot\left[v_i\cdot\left[n \bmod p\oplus L\oplus L^2\right]\oplus\sum_{\substack{k=1\\k\neq i}}^{n-1} v_k\cdot\left(L\oplus L^2\right)\right]=v_i\cdot$$

Последнее следует из того, что $L \oplus L^2 = 0$. **Теорема доказана.**

Легко убедиться теперь, что использование данного преобразования в полях, являющихся расширением двоичного поля, в отличие от преобразования Фурье позволяет подавать на вход УПФ векторы, имеющие четную длину. Например, в поле $GF(2^8)$ преобразование (4) существует для векторов длины = 2,4,14,16,50,84,254, что соответствует двоичным блокам входных данных 16, 32, 112, 128, 400, 672 и 2032 бит. Блок длиной в 32 бита может быть обработан на 32-разрядном микропроцессоре за один такт, т.е. такие длины уже позволяют оптимально использовать существующую элементную базу вычислительной техники для программной и аппаратной реализации УПФ.

Интересным на наш взгляд является также то, что существует два способа нахождения выходного вектора для $\boldsymbol{\Phi_l(v)}$ в $GF(2^m)$:

1. Выполнение преобразования согласно выражению (4).
2. Табличный способ.

Первый способ заключается в выполнении всех математических операций, определенных выражением (4). В этом случае часто пользуются матричным представлением выполняемых преобразований. Так, например преобразование $\boldsymbol{\Phi_4(v)}$ в $GF(2^8)$ может быть записано выражением вида

$$\begin{vmatrix} V_1\\ V_2\\ V_3\\ V_4\end{vmatrix} = \begin{vmatrix} 52 & 103 & 154 & 205\\ 103 & 205 & 52 & 154\\ 154 & 52 & 205 & 103\\ 205 & 154 & 103 & 52\end{vmatrix}\cdot\begin{vmatrix} v_1\\ v_2\\ v_3\\ v_4\end{vmatrix}.$$

В соответствии с предлагаемым способом для входного вектора $v = \{2, 45, 178, 236\}$ выходной вектор будет иметь вид $V = \{53, 217, 187, 73\}$.

Прежде всего заметим, что операция сложения в поле, являющемся расширением двоичного поля – есть операция побитного ИСКЛЮЧАЮЩЕГО ИЛИ, а каждая точка входного вектора, умноженная на константу, влияет на все точки выходного вектора. Суть табличного способа нахождения выходного вектора для $\boldsymbol{\Phi_l(v)}$ в $GF(2^m)$ заключается в том, что строится $\boldsymbol{l}$ таблиц, состоящих из $\boldsymbol{2^m}$ элементов размерностью $\boldsymbol{m \cdot l}$ бит, а выходной вектор получается путем сложения элементов таблиц, соответствующих точкам входного вектора.

Например для $\Phi_4(v)$ в $GF(2^8)$ таблицы будут иметь вид

$$T_1[v_1]=\begin{bmatrix} S(52\cdot v_1) \\ S(103\cdot v_1) \\ S(154\cdot v_1) \\ S(205\cdot v_1) \end{bmatrix}, \quad T_2[v_2]=\begin{bmatrix} S(103\cdot v_2) \\ S(205\cdot v_2) \\ S(52\cdot v_2) \\ S(154\cdot v_2) \end{bmatrix},$$
$$T_3[v_3]=\begin{bmatrix} S(154\cdot v_3) \\ S(52\cdot v_3) \\ S(205\cdot v_3) \\ S(103\cdot v_3) \end{bmatrix}, \quad T_4[v_4]=\begin{bmatrix} S(205\cdot v_4) \\ S(154\cdot v_4) \\ S(103\cdot v_4) \\ S(52\cdot v_4) \end{bmatrix}, \tag{6}$$

где $S(v)$ означает нелинейный переход от десятичного представления элемента поля к двоичному представлению. Выходной вектор V может быть получен как

$$V = T_1[v_1] \oplus T_2[v_2] \oplus T_3[v_3] \oplus T_4[v_4] \tag{7}$$

Заметим, что для хранения таблиц (6) требуется 4 кБайта памяти, а выражение (7) можно рассматривать как преобразование входного блока размера 32 бита в выходной блок такой же размерности. Для сравнения в случае, когда необходимо построить полную таблицу замены 32-битного блока, потребуется $2^{32} \cdot 32$ бита памяти, что соответственно равно 16 Гбайтам.

Способ достижения нелинейности теоретико-числовых преобразований (описанный выше) может быть использован и для достижения нелинейности УПФ. Учитывая, что программная и аппаратная реализации УПФ могут быть оптимизированы под использование их на современных микропроцессорах и в тоже время УПФ позволяет производить нелинейную замену блоков при ограниченном размере памяти, на наш взгляд, УПФ может быть полезно при построении криптографических протоколов.

4 Использование усеченного преобразования Фурье над полем Галуа *GF (q)* в системах криптографической защиты информации

В разных источниках приводятся различные подходы к классификации блочных шифров, но чаще всего выделяют два основных класса: шифры, построенные с использованием цепи Фестеля (DES, DEAL, E2, LOKI97, RC6, Twofish, MARS, Гост 28147- 89), и шифры, которые построены на основе чередования процедур перестановок и подстановок (SPN – substitution-permutation network) (Square, Rijndael, SAFER+, Serpent, CRYPTON). При этом особенностью цепи Фестеля является то, что она дает возможность использовать на одном цикле необратимые операции в отличие от SPN – структуры, где биективность преобразований основное требование, в связи с тем, что процесс дешифрования состоит из операций, которые являются обратными к используемым операциям при шифровании. УПФ является биективным преобразованием, поэтому наиболее целесообразно будет его использование при проектировании симметричных блочных шифров на основе SPN – структуры.

Один цикл криптографического преобразования в шифре, построенном на основе SPN – структуры, может быть представлен тремя последовательно выполняемыми шагами:

1. Нелинейная замена входного блока.
2. Транспозиция входного блока.
3. Сложение с ключом.

Следует отметить, что стойкость такой конструкции к атакам криптоаналитика полностью зависит от вводимой в цикле нелинейности, так как основой построения и реализации таких атак является использование характеристик, описывающих вероятность прохождения через циклы шифрования специфических пар открытых текстов. И если удается найти характеристику с высокой вероятностью, то можно ставить и решать задачи криптоанализа со сложностью меньшей, чем прямой перебор ключей.

Построению S-блоков (таблиц нелинейной замены) посвящено очень много работ как отечественных, так и мировых ученых. Наиболее прогрессивным подходом в этом направлении можно выделить использование для построения S-блоков бент-функций, которые являются максимально нелинейными в криптографическом смысле. Однако такой подход имеет одно ограничение, заключающееся в том, что для хранения S-боксов больших размеров (осуществляющих замену блока размера 32 и более бит) требуется недостижимый объем памяти. Как показывает анализ симметричных блочных шифров, которые были предложены в качестве кандидатов на новый стандарт в конкурсах AES и NESSIE, размер блока данных, обрабатываемый одним оператором, функции шифрования в одном цикле равен 4 (Rijndael), либо 16 байт (RC6). При этом такой размер блока для функции шифрования достигается за счет использования на шаге транспозиции входного блока линейных преобразований вместо битовых перестановок (DES) или битовых сдвигов (ГОСТ 28147-89). Сами S-блоки имеют небольшой размер, как правило, 8×8 бит. Среди линейных преобразований широкое применение получили преобразования на основе МДР-кодов. Подобные преобразования используются в шифрах Shark, Square, Rijndael, Khazad, Anubis. Главное преимущество линейных преобразований этого вида заключается в том, что число задействованных S-блоков в контексте дифференциального или линейного криптоанализа ("количество ветвлений") до и после такого линейного преобразования будет максимально возможным, т.е. равным $M+1$, где M – число S-блоков, покрываемых МДР-преобразованием.

Например, в симметричном блочном шифре AES-Rijndael [5] нелинейность обеспечивается за счет последовательного выполнения двух операций: замены элемента на его мультипликативно обратный элемент в поле $GF(2^8)$ и аффинного преобразования. Но при этом активизация одного S-блока на входе цикла за счет использования МДР-преобразования приводит к тому, что на следующем цикле задействованными являются уже четыре S-блока, что не позволяет криптоаналитику работать с одним активным S-блоком на каждом цикле и соответственно затрудняет задачу проведения криптоанализа.

На последнем этапе конкурса AES все кандидаты отмечаются как устойчивые к известным методам криптоанализа, но, согласно результатам исследований, опубликованных в [8], Rijndael имеет не лучший, в сравнении с остальными, показатель статистической безопасности. На наш взгляд выбор шифра AES-Rijndael в качестве нового стандарта во многом определила возможность его реализации на любой аппаратно-программной платформе вычислительной техники. Наилучший временной показатель обеспечивается при использовании 32-битных микропроцессоров.

Рассмотрим, каким образом удается достичь оптимизации при использовании 32-битного микропроцессора для шифрования одного блока в AES-Rijndael. В этом случае разработчики стандарта используют цикловое преобразование, описываемое выражением

$$V_j = T_1[v_1] \oplus T_2[v_2] \oplus T_3[v_3] \oplus T_4[v_4] \oplus K_j,$$

где K_j – ключ цикла;

$T_i(v_i)$ – таблицы вида: $T_1[v_1]=\begin{bmatrix} 2\cdot S(v_1) \\ 1\cdot S(v_1) \\ 1\cdot S(v_1) \\ 3\cdot S(v_1) \end{bmatrix}$, $T_2[v_2]=\begin{bmatrix} 3\cdot S(v_2) \\ 2\cdot S(v_2) \\ 1\cdot S(v_2) \\ 1\cdot S(v_2) \end{bmatrix}$,

$$T_3[v_3]=\begin{bmatrix} 1\cdot S(v_3) \\ 3\cdot S(v_3) \\ 2\cdot S(v_3) \\ 1\cdot S(v_3) \end{bmatrix}, \quad T_4[v_4]=\begin{bmatrix} 1\cdot S(v_4) \\ 1\cdot S(v_4) \\ 3\cdot S(v_4) \\ 2\cdot S(v_4) \end{bmatrix},$$

где $S(v)$ означает нелинейную замену байта, определенную для данного стандарта. При этом время выполнения такого преобразования будет зависеть от времени осуществления четырех поисков по таблице и времени выполнения трех операций сложения. Заметим, что складываются 32-битные числа, а их сложение на 32-битном микропроцессоре выполняется за один такт.

Обратим внимание на то, что такая же схема реализуется и для нахождения выходного вектора в УПФ: выражения (6) – таблицы УПФ и выражение (7) – вычисление $\Phi_4(v)$ в $GF(2^8)$. Следовательно, мы можем использовать УПФ для построения симметричного блочного шифра по той же схеме, что использовалась разработчиками шифра Rijndael. Стойкость такого шифра к известным видам криптоанализа будет зависеть от введенной в УПФ нелинейности.

Нелинейность в УПФ определяется переходом от десятичного представления элемента поля к двоичному представлению. Так как поле строится на основе регистра сдвига с обратной связью и при этом допускается использование любого примитивного многочлена над данным полем, то вариантов построения, а, следовательно, и переходов будет несколько. Например, поле $GF(2^8)$ может быть построено на основе восьми примитивных многочленов, полный список которых из [7] представлен в табл. 1.

Таблица 1

№ п/п	Коэффициенты								
	X^8	X^7	X^6	X^5	X^4	X^3	X^2	X^1	X^0
1.	1	1	1	1	0	0	1	1	1
2.	1	0	1	1	0	0	0	1	1
3.	1	0	0	0	1	1	1	0	1
4.	1	0	0	1	0	1	0	1	1
5.	1	0	1	0	1	1	1	1	1
6.	1	0	1	1	0	1	0	0	1
7.	1	0	1	1	0	0	1	0	1
8.	1	1	1	0	0	0	0	1	1

Учитывая, что величина сдвига в регистре может варьироваться, количество вариантов построения поля будет еще больше. Поле $GF(2^{16})$ мы можем построить с использованием уже порядка 1000 примитивных многочленов [7].

При исследовании стойкости симметричного блочного шифра используют максимальные значения таблиц распределения дифференциалов и линейных аппроксимаций. Для примитивных многочленов из табл. 1 и величине сдвига, равной одному биту, соответствующие

им максимальные значения дифференциалов ($NS_D(\alpha,\beta)$) и линейных аппроксимаций ($NS_{LA}(\alpha,\beta)$) представлены в табл. 2. В табл. 2 также показаны соответствующие значения для нелинейных преобразований, используемых в Rijndael и Safer++. При нахождении значений *шифра Rijndael использовались таблицы нелинейной замены [5].*

Таблица 2

№ п/п	Вид используемого нелинейного преобразования	$NS_D(\alpha,\beta)$	$NS_{LA}(\alpha,\beta)$
1.	В поле, построенном на многочлене 1	[210,50] = 8	[222,90] = 30
2.	В поле, построенном на многочлене 2	[171,5] = 8	[187,251] =38
3.	В поле, построенном на многочлене 3	[217,49] = 8	[102,13] = 28
4.	В поле, построенном на многочлене 4	[85,149] =10	[110,14] = 28
5.	В поле, построенном на многочлене 5	[107,66] = 8	[255,16] = -32
6.	В поле, построенном на многочлене 6	[170,63] = 14	[102,255] = 30
7.	В поле, построенном на многочлене 7	[204,9] = 8	[238,207] = 30
8.	В поле, построенном на многочлене 8	[153,14] = 8	[255,155] = 30
9.	Rijndael	[40,9] = 6	[207,82] = 23
10.	Safer++ ($(45^X \bmod 257) \bmod 256$)	[128,253] = 128	[58,80] = -46
11.	Мультипликативно обратный элемент в поле GF (2^8)	[1,1] = 256	[2,2] = 128
12.	Мультипликативно обратный элемент в поле GF (2^8+1)	[253,253] = 72	[126,1] = -34

Из табл. 2 видно, что нелинейное преобразование на основе регистра сдвига с обратной связью имеет хорошие линейные и дифференциальные характеристики и допускает использование УПФ для нелинейной замены блоков в произвольном блочном шифре. Наилучшие характеристики достигаются при использовании примитивного многочлена вида

$$X^8 + X^4 + X^3 + X^2 + 1.$$

Экспериментальные исследования преобразования $\boldsymbol{\Phi_4(v)}$ в $GF(2^8)$ показали также, что число задействованных нелинейных переходов применительно к дифференциальному или линейному криптоанализу до и после такого преобразования равно 5 , т.е. максимально возможное. Это говорит о том, что УПФ в данном применении обеспечивает те же свойства, что и МДР-преобразование.

При исследовании статистической безопасности для $\boldsymbol{\Phi_4(M)}$ и $\boldsymbol{\Phi_{16}(M)}$ в $GF(2^8)$ использовался следующий алгоритм:

1. Генерируется исходный блок $\boldsymbol{M_1}$ (нулевой, единичный или случайный).
2. Находится $\boldsymbol{C_1 = \Phi_1(M_1)}$.
3. Определяется блок $\boldsymbol{M_2 = M_1}$ **xor** $\boldsymbol{G}$, где $\boldsymbol{G}$ в цикле изменяет $\boldsymbol{M_1}$ в $\boldsymbol{g}$ битовых позициях.
4. Находится $\boldsymbol{C_2 = \Phi_1(M_2)}$.
5. Вычисляется расстояние Хэмминга $\boldsymbol{W(C_2, C_1)}$. Переходим на шаг 3.

Группированный статистический ряд для $W(C_i, C_j)$ как случайной величины при использовании $\Phi_4(M_l)$ в $GF(2^8)$, где M_i (случайный) и g в диапазоне [1÷5 бит], представлен в табл. 3.

Таблица 3

Интервал	Частота измеренная	Частота совокупная	Ожидаемая частота	Ожид. частота совокуп.	Разность
0	0	0	0	0	0
2	0	0	0	0	0
4	0	0	0	0	0
6	0	0	0	0	0
8	3	3	2	2	1
10	9	12	12	14	-3
12	37	49	46	61	-9
14	116	165	104	164	12
16	130	295	144	308	-14
18	129	424	125	433	4
20	73	497	67	500	6
22	27	524	22	522	5
24	2	526	4	527	-2
26	1	527	0	527	1
28	0	527	0	527	0
30	0	527	0	527	0

Проверка гипотезы о виде распределения $W(C_i, C_j)$ (на основе критерия Пирсона [6]) для результатов, представленных в табл. 2, показала, что распределение расстояния Хэмминга $W(C_i, C_j)$ как случайной величины подчинено биномиальному закону. Описательная статистика для $W(C_i, C_j)$ при использовании $\Phi_4(M_l)$ в $GF(2^8)$ представлена в табл. 4.

Таблица 4

Параметр	M_l (случайный) и g в диапазоне [1÷5 бит]	M_l (случайный) и g в диапазоне [7÷24 бит]	M_l (нулевой) и g в диапазоне [1÷5 бит]	M_l (единичный) и g в диапазоне [7÷23 бит]
Среднее	15,89184061	15,944	15,96875	16,032
Стандартная ошибка	0,122839257	0,12706699	0,04157825	0,123598159
Медиана	16	16	16	16
Стандартное отклонение	2,819957027	2,841304281	2,35202087	2,763738863
Дисперсия выборки	7,952157632	8,07301002	5,53200219	7,638252505
Интервал	17	16	11	16
Минимум	8	8	11	8
Максимум	25	24	22	24
Сумма	8375	7972	8012	8016
Количество опытов	527	500	500	500

Из результатов, представленных в табл. 4, видно, что при условии отличия двух входных векторов в g битовых позициях для $\Phi_4(M_l)$ в поле $GF(2^8)$ получаются выходные векторы, отличающиеся в среднем в половине битовых позиций, что говорит о хороших свойствах

статистической безопасности для данного преобразования. При увеличении количества циклов преобразования, т.е. когда выходной вектор подается на вход УПФ несколько раз, полученные характеристики улучшаются.

Группированный статистический ряд расстояний Хэмминга двух выходных векторов как случайной величины для $\Phi_{16}(M_i)$ в $GF(2^8)$, где M_i (случайный) и g в диапазоне [1÷5 бит], представлен в табл. 5.

Таблица 5

Интервал	Частота измеренная	Частота совокупная	Ожидаемая частота	Ожид. частота совокуп.	Разность
50	100	100	1	1	99
52	0	100	6	7	-6
54	600	700	33	40	567
56	800	1500	134	174	666
58	500	2000	431	606	69
60	1000	3000	1072	1678	-72
62	1700	4700	2027	3705	-327
64	2000	6700	2849	6555	-849
66	1800	8500	2898	9453	-1098
68	1200	9700	2055	11508	-855
70	1100	10800	965	12473	135
72	1200	12000	279	12752	921
74	400	12400	44	12797	356
76	100	12500	3	12800	97
78	300	12800	0	12800	300

В ходе проведения экспериментов наблюдалось группирование значений $W(C_i, C_j)$ в интервале [50, 80] бит. В данном применении УПФ этот результат является положительным, так как одно из требований к базовой функции шифрования заключается в том, чтобы минимальные изменения входного блока вызывали максимальные изменения в выходном. Описательная статистика для $W(C_i, C_j)$ при использовании $\Phi_{16}(M_1)$ в $GF(2^8)$ представлена в табл. 6.

Таблица 6

Параметр	M_1 (случайный) и g в диапазоне [1÷5 бит]	M_1 (нулевой) и g в диапазоне [1÷5 бит]	M_1 (единичный) и g в диапазоне [1÷5 бит]
Среднее	64,2890625	63,8875	64,78125
Стандартная ошибка	0,050622545	0,04757158	0,045541853
Медиана	64	64	64,5
Стандартное отклонение	5,727287143	5,382109884	5,152472465
Дисперсия выборки	32,80181801	28,96710681	26,5479725
Интервал	28	25	17
Минимум	50	54	56
Максимум	78	79	73
Сумма	822900	815200	829200
Количество опытов	12800	12800	12800

Из табл. 6 можно увидеть, что $\Phi_{16}(M_1)$, как и $\Phi_4(M_1)$ в поле $GF(2^8)$, представленное в табл. 4, имеет хорошую статистическую безопасность.

Выводы

1. Предложен способ достижения нелинейности теоретико-числовых преобразований в полях характеристики 2.
2. Установлено, что нелинейность усеченного преобразования Фурье зависит от используемого для построения поля примитивного многочлена и величины сдвига в формирующем регистре.
3. Показано, что усеченное преобразование Фурье позволяет производить нелинейную замену блоков большой длины при ограниченном размере памяти. Так для организации нелинейной замены блока размера 128 бит требуется 2^{19} бит памяти, для сравнения стандартный подход требует $2^{128} \cdot 128 = 2^{135}$ бит памяти.
4. Подтверждено, что использование усеченного преобразования Фурье в поле $GF(2^8)$ для построения симметричного блочного шифра на основе SPN-структуры позволяет получить показатель статистической безопасности и цикловые дифференциальные и линейные характеристики, соизмеримые с соответствующими показателями кандидатов на новый стандарт в конкурсах AES и NESSIE.

Список литературы: 1. *Блейхут Р.* Теория и практика кодов, контролирующих ошибки. М.: Мир, 1986. 576 с. 2. *Schnorr C.P.* FFT-Hash 2, Efficient Cryptographic Hashing // Advances in cryptology – Eurocrypt'92. 1992. P. 45 – 54. 3. *Vaudenay S.* FFT-Hash 2 is not yet Collision-free // Proc. Of Crypto'92. 1992. P. 587 – 593. 4. *Ященко В.В.* О критерии распространения для булевых функций и бент-функций // Проблемы передачи информации. 1997. Вып. 1. С. 52 – 63. 5. *Горбенко И.Д., Скрыпник Л.В., и др.* Стандарт симметричного шифрования 21 века: свойства, режимы работы, реализация // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2001. Вып. 119. С. 22 – 35. 6. *Гмурман В.Е.* Теория вероятностей и математическая статистика. М.: Высш. шк., 1977. 479 с. 7. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки: Пер. с англ. М.: Мир, 1976. 600 с. 8. *Бондаренко М.Ф., Горбенко И.Д. и др.* Улучшенный стандарт симметричного шифрования XXI века: концепция создания и свойства кандидатов // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2000. Вып. 114. С. 5 – 14.

Харьковский национальный
университет радиоэлектроники
Харьковский военный университет

Поступила в редколлегию 15.01.2003