УДК 621.391:519.2

Белецкий А.Я., Белецкий А.А., Аксентий Е.А. (НАУ)

ДИФФЕРЕНЦИАЛЬНЫЕ ХАРАКТЕРИСТИКИ МИНИ-ВЕРСИЙ СИММЕТРИЧНОГО RSB-ШИФРАТОРА

Введение и постановка задачи

Мир компьютерных технологий, пронизавший все сферы человеческой деятельности, обуславливает необходимость высокоскоростного обмена между абонентами компьютерных сетей большими объемами цифровой информации, носящей, как правило, конфиденциальный характер. Целям защиты информации от несанкционированного доступа как нельзя лучше отвечают одноключевые блочные симметричные шифраторы (БСШ). К настоящему времени разработано огромное число симметричных шифраторов, причем каждый разработчик шифра старается убедить потенциальных потребителей в достоинствах и преимуществах именно своего продукта. Это естественно и было бы странно, если бы это было не так. Рынок потребителей шифраторов также громаден и каждый покупатель подбирает на нем товар по приемлемым для себя показателям.

Выделим на множестве критериев, по которым оценивается эффективность БСШ, два показателя, определяющие важнейшие потребительские качества шифра. А именно, криптостойкость и ресурсоемкость. Под *криптографической стойкостью* (или *криптостойкостью*) понимается способность криптографического алгоритма противостоять возможным атакам на него. Стойким считается алгоритм, который для успешной атаки требует от противника недостижимых вычислительных ресурсов, недостижимого объёма перехваченных открытых и зашифрованных сообщений или же такого времени раскрытия, что по его истечению защищенная информация будет уже не актуальна. Под *ресурсоемкостью* будем понимать совокупные вычислительные ресурсы, которые включают объем запрашиваемой памяти и число арифметико-логических операций, выполняемых компьютером (затраты машинного времени), необходимые для реализации шифра.

Основная задача шифратора состоит в том, чтобы преобразовать открытый (исходный) текст, который, как правило, является избыточным (коррелированным, сжимаемым), в выходной текст (шифротекст, криптограмму или шифрограмму), представляющий собой стохастическую последовательность битов, образующих процесс типа «белого» шума (некоррелированного и несжимаемого). Приближение шифрограммы к белому шуму осуществляется в БСШ за несколько итераций (раундов) криптопреобразования открытого текста. Принято считать, что качество шифратора тем выше, чем за меньшее число раундов преобразования криптограмма приближается по своим свойствам к свойствам, характерных для белого шума, и, тем самым, достигаются асимптотические характеристики шифра. Один из способов оценки минимального числа рандов, при котором обеспечивается отбеливание криптограммы, основан на применении так называемых дифференциальных характеристик шифраторов [1], рассчитываемых на этапе выполнения дифференциального криптоанализа шифров. Под *дифференциальной характеристикой* шифратора понимается зависимость *полных дифференциалов* шифра от числа раундов преобразования (более подробные пояснения введенных понятий будут даны ниже по тексту в разделе, посвященному описанию методики исследований). При этом шифр рассматривается как одна большая подстановка, т.е. дифференциальная характеристика покрывает весь шифр. Естественно, что вычисление полных дифференциалов шифров возможно лишь на его мини-версиях, в которых существенно укорочены основные параметры шифратора, такие как: размер ключа, длина блока и др. Как показано в работ [2], после некоторого числа раундов шифрующего преобразования дифференциальная характеристика шифра будет стремиться к дифференциальной характеристике случайной подстановки, при которой и достигается выше упоминавшееся отбеливание шифрограммы.

Целью исследования данной работы является, в основном, оценка криптостойкости RSB-шифра по дифференциальным характеристикам его мини-версий. Частично обсуждается также проблема ресурсоемкости шифратора. Но перед этим мы приведем основные сведения, касающиеся полного RSB-криптоалгоритма.

1. Общее описание RSB алгоритма

Аббревиатура RSB происходит от ключевых слов **R**ound, **S**tep, **B**lok – подчеркивая тем самым, что основными для криптоалгоритма являются *раундовые* преобразования, разбитые на определенное число *шагов*, а действие алгоритма осуществляется над *блоками* открытого или закрытого текстов. RSB – это итерационный шифр, который доставляет уникальную возможностью по изменению как размеров секретных ключей, так и числа шагов (соответственно, и раундов) шифрования. Отличительная особенность RSB алгоритма состоит в том, что в нем используется оригинальная, не встречающаяся ни в каком другом алгоритме, функция шифрования (криптопримитив) типа *скользящего кодирования,* которая обеспечивает не только глубокое перемешивание открытого текста, но и участвует в формировании *локального блочного раундового ключа* (определение дается ниже) для очередного шифруемого блока [3,4]. Тем самым все преобразования, выполняемые криптоалгоритмом, становятся зависимыми не только от секретного ключа, но и от шифруемых данных, т.е. относятся к классу «управляемых операций криптопреобразования», или «управляемых криптопримитивов» [5,6].

RSB шифр не в полней мере отвечает основным характеристикам, определяющим классические БСШ. Примитивом скользящего кодирования в RSB алгоритме осуществляется вложенное сцепление 32-битных элементов всего шифруемого текста. Тем самым оказываются сцепленными также блоки, на которые разбивается входной текст. Вследствие указанной причины в RSB алгоритме, во-первых, невозможно организовать параллельное криптопреобразование блоков и, во-вторых, затруднена, или вообще исключена, реализация типовых режимов шифрования, обычно используемых в БСШ.

Алгоритм RSB предусматривает различные варианты длины блока (для применения в различных классах безопасности) и переменную длину ключа шифрования (кратную 64-м битам) для каждого варианта длины блока. Структура RSB алгоритма (рис. 1) идентична для различных размеров блока, равных $64 \cdot 2^k$, где $k = 0, 1, 2, 3$ – коэффициент кратности длины блока, т.е. алгоритм поддерживает блоки длиной 64, 128, 256 и 512 бит.

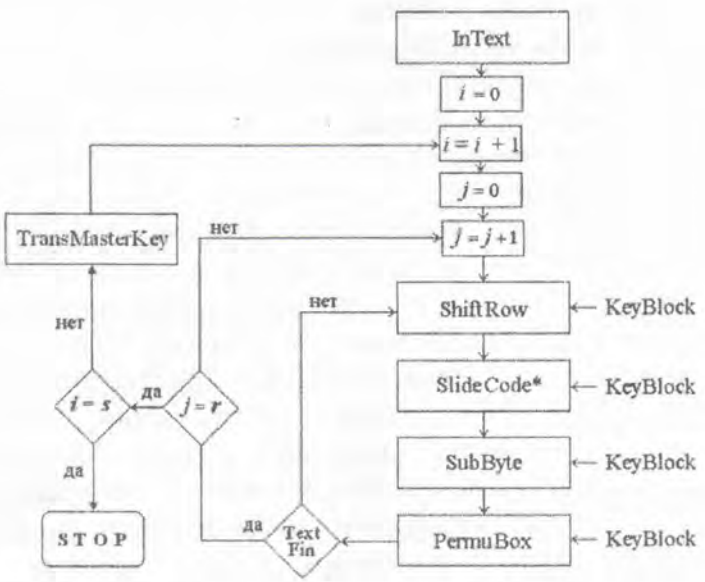

Рис. 1. Обобщенная структурная схема RSB криптоалгоритма.

Описание RSB алгоритма приводится далее для длины блока 256 бит и отмечаются некоторые особенности реализации шифра при других размерах блока. Приведем основные параметры алгоритма:

Размер (длина) блока - $N = 64, 128, 256$ или 512 бит;
Длина раундового ключа - 32 бита;
Длина общего (шагового, Common Key или Master Key) ключа - $32*r$, $r = 2, 4,\ldots$;
Число шагов шифрования - $s = 1, 2,\ldots$;
Общее число раундов шифрования - $r*s$;
Размер элемента скользящего кодирования - 32 бита;
Размер элемента нелинейной подстановки - 8 бит.

Каждый раунд зашифрования RSB алгоритма включает следующую совокупность последовательно выполняемых криптографических примитивов:

- стохастическая круговая прокрутка шифруемого блока (ShiftRow);
- скользящее кодирование 32-разрядных элементов блока (SlideCode);
- стохастическая нелинейная подстановка (замена) байтов блока (SubByte);
- стохастическая перестановка элементов (слов) блока (PermuBox).

Перечисленные выше примитивы параметризуются с помощью блочного раундового ключа, структурная схема которого приведена на рис. 2.

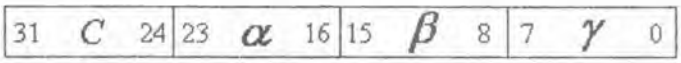

Рис.2. Структурная схема блочного раундового ключа

Локальные блочные раундовые ключи меняются каждый раз при переходе к очередному преобразуемому блоку. Такая модификация ключей достигается за счет операций скользящего кодирования, выполняемых в предыдущих блоках. В силу отмеченной особенности 32-разрядные компоненты общего ключа шифрования выше названы *базовыми раундовыми ключами*, а результат их преобразования функцией скользящего кодирования будем называть *локальными блочными раундовыми ключами.* Для первого блока шифруемого текста локальный раундовый ключ совпадает с базовым раундовым ключом. Далее приводится более подробное описание основных криптографических примитивов.

Стохастическая круговая прокрутка блока. Посредством данной операции осуществляется циклический сдвиг влево (круговая прокрутка по часовой стрелке) шифруемого блока на случайное нечетное число, которое задается восьмиразрядным двоичным байтом *C* (рис. 2). Восемь разрядов этого байта (для 512-битного блока) считываются из сектора *C* блочного раундового ключа (разряды 31–24 на рис. 2), а в младший разряд формируемой кодовой комбинации принудительно записывается единица. Тем самым код, которым определяется порядок циклического сдвига блока, будет содержать нечетное число в интервале от 1 до 255. Если длина блока составляет 256, 128 или 64 бита, то из байта *C* считываются, соответственно, семь, шесть или пять младших разрядов, причем в самый младший разряд считанной кодовой комбинации заносится единица.

Скользящее кодирование 32-разрядных элементов блока. Операция скользящего кодирования выполняет в RSB криптоалгоритме двойную роль. Во-первых, она обеспечивает достаточно глубокое перемешивание преобразуемого текста, цель которого состоит в том, чтобы сделать как можно более сложной зависимость между ключом и шифротекстом. И, во-вторых, с помощью такой операции осуществляется модификация локальных блочных раундовых ключей, под управлением которых выполняются криптографические преобразования блоков текста, начиная со второго. В результате

указанной модификации локальный блочный раундовый ключ i - го блока, $i > 1$, становится зависимым как от исходного базового раундового ключа, под управлением которого осуществляются преобразования первого блока текста, так и от шифруемых данных всех предыдущих $(i-1)$ - х блоков.

В RSB шифре реализованы два типа скользящего кодирования: лево- и правостороннее, причем *левостороннее скользящее кодирование* применяется на нечетных раундах шифрования, а *правостороннее* – на четных раундах. Структурная схема алгоритма прямого левостороннего скользящего кодирования (процесс преобразования текста осуществляется по направлению слева направо) на этапе зашифрования первого блока приведена на рис. 3, где $\oplus$ есть оператор поразрядного сложения по mod 2; R' – 32-разрядный исходный (базовый) раундовый ключ, а R'' – 32-разрядный локальный раундовый ключ для второго шифруемого блока.

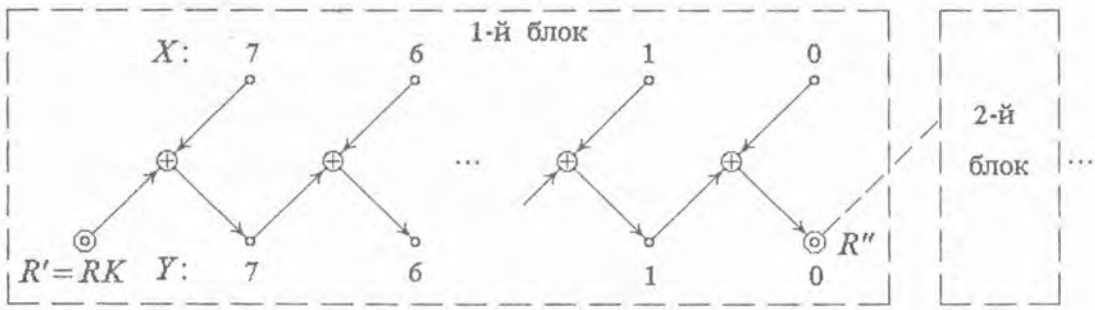

Рис.3. Структурная схема прямого левостороннего скользящего кодирования

Структурная схема алгоритма обратного левостороннего скользящего кодирования показана на рис. 4.

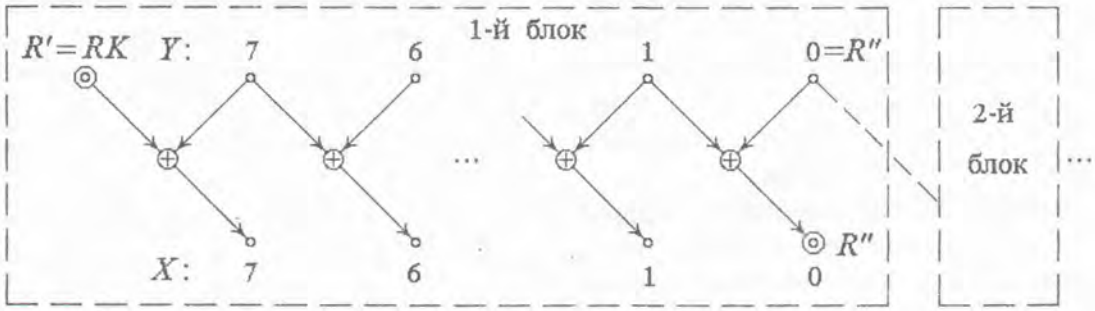

Рис.4. Структурная схема обратного левостороннего скользящего кодирования

Кроме левостороннего в RSB алгоритме на четных раундах применяется также правостороннее скользящее кодирование (для тех же целей, что и левостороннее); при этом процесс преобразования шифруемого текста осуществляется по направлению справа налево.

Обратим внимание на то, что предлагаемые схемы скользящего кодирования являются ничем иным, как *преобразованиями Грея* «наоборот» [7]. Это означает, что прямое скользящее кодирование представляет собой (по направлению, соответственно, лево- или правостороннее) *обратное преобразование Грея*, тогда как обратное скользящее кодирование есть не что иное, как *прямое* преобразование Грея.

Нелинейная подстановка байтов. Схема построения S-боксов [8], посредством которых реализуются операции нелинейной подстановки (замены) байтов в RSB алгоритме, наследует основные черты S-боксов, принятых в AES шифре [9], и имеет вид:

$$y = (x \oplus \alpha)_{\varphi}^{-1} \otimes A \oplus \beta, \qquad (1)$$

где α и β – байты блочного раундового ключа, показанного на рис. 2; A – невырожденная (0, 1)-матрица преобразования; x^{-1} – элемент, мультипликативно обратный байту x над выбранным неприводимым полиномом φ.

Алгоритм (1) оптимизирован по параметрам φ и A. В качестве критерия оптимальности выбран критерий минимума среднеквадратического отклонения (СКО) фактически реализуемого рассеивания откликов S-боксами o в квадрате XY размером 256 на 256 от равномерно плотного рассеивания. Как показали результаты машинных расчетов, выбранный критерий оптимизации является инвариантным к аддитивным компонентам α и β преобразования (1). Обнулив эти компоненты, приходим к более простому выражению для S-бокса.

$$y = x_{\varphi}^{-1} \otimes A. \qquad (2)$$

Суть оптимизации (2) сводится к следующему. Предварительно формировался полный набор, состоящий из 104968 симметричных инволютивных двоичных матриц восьмого порядка, на что было потрачено более семи суток машинного времени процессора средней производительности. Матрицы A выбраны инволютивными и симметричными из соображений минимизации затрат машинного времени на их генерацию. Программный алгоритм оптимизации организован в виде двух вложенных циклов. Внешним циклом производился перебор 30-ти неприводимых полиномов φ восьмой степени, а внутренним – перебор всего набора матриц А. Для каждой пары параметров φ и A по формуле (2) для $x \in \{0-255\}$ рассчитывались значения y, которые затем вносились в квадратную таблицу, состоящую из 64 ячеек. Математическое ожидание 256 точек (откликов y) для равномерно плотного их распределения, попадающих в каждую из 64 ячеек, составляет 4. Вычислялась частота n_{ij} попадания y в ячейку с координатами $i, j \in \{1-8\}$, а затем определялось среднеквадратическое отклонение σ по всем ячейкам таблицы. Как показали результаты машинных расчетов, минимального значения σ достигает при

$$\varphi = 100011011 \qquad (3)$$

и

$$\dot{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}. \qquad (4)$$

Обратим внимание на то, что полином (3) совпадает с полиномом, используемом в шифре AES. Отметим, кроме того, что матрица преобразования (4), доставляет более равномерное распределение откликов y, чем соответствующая циркулянтная матрица AES.

Стохастическая перестановка элементов блока. С помощью этого криптографического примитива осуществляется стохастическая перестановка (рассеивание)

двоичных элементов в пределах шифруемого блока. Алгоритм перестановки задается преобразованием, подобным преобразованию (1), т.е.

$$y = (x \oplus \beta)_{\varphi}^{-1} \otimes A \oplus \gamma, \qquad (5)$$

согласно которому элемент, находящийся в ячейке блока по адресу x, перемещается в ячейку этого же блока по адресу y [10]. Аддитивные компоненты β и γ выбираются из соответствующих секторов раундового ключа, показанного на рис. 2.

Элементом перестановки (табл. 1) может быть один, два или четыре бита, в зависимости от длины N шифруемого блока и степени n выбранного неприводимого полинома φ, участвующих в операции перестановки (5).

Таблица 1. Элементы перестановки блоков шифруемых текстов

Степень полинома (n)	$N = 64$	$N = 128$	$N = 256$	$N = 512$	Размерность S - боксов
8	–	–	Бит	2 бита	16x16
7	–	Бит	2 бита	Полубайт	16x8
6	Бит	2 бита	Полубайт	–	8x8
5	2 бита	Полубайт	–	–	8x4
4	Полубайт	–	–	–	4x4

Таким образом, перестановка элементов шифруемого блока осуществляется с помощью обобщенной нелинейной подстановки (S-бокса). Порядок матрицы и всех двоичных векторов преобразования в (5) совпадает со степенью n неприводимого полинома φ.

Обратимся к рис. 1. Внутренним циклом (по j) осуществляются последовательные криптопреобразования всех N-битных блоков расширенного (за счет возможного дополнения пробелами неполного последнего блока) исходного текста. При этом преобразования первого (самого левого) блока происходит под управлением первого базового раундового 32-битного ключа. В качестве локальных раундовых ключей последующих блоков выступают 32-битные векторы (см. рис. 3), сформированные на последнем этапе скользящего кодирования в предыдущем блоке. Когда закончено зашифрование всех блоков открытого текста (закончен первый раунд зашифрования), происходит переход к преобразованию текста, предварительно зашифрованного в первом раунде, но теперь уже под управлением второго (четного) базового раундового 32-битного ключа. Особенность зашифрования на втором раунде (как и на всех четных рандах) состоит в том, что в нем используется правостороннее скользящее кодирование. Это означает, что процесс зашифрования протекает справа налево, начиная с последнего (самого правого) блока. При этом преобразование первого блока (в данном случае - самого правого) осуществляются под управлением второго базового раундового ключа, а последующих блоков – под управлением локальных раундовых ключей, сформированных на этапе зашифрования предыдущих блоков. Примитивы ShiftRow, SubByte и PermuBox на втором (четном) раунде выполняются точно так же, как и на первом (нечетном) раунде.

Когда закончено криптопреобразование текста всеми базовыми раундовыми ключами (а таких ключей r), переходят к зашифрованию на втором шаге. При этом общий ключ Common Key (СК) подвергается модификации (в блоке TransMasterKey на рис. 1), которая сводится к циклическому сдвигу СК на семь разрядов влево (рис. 5).

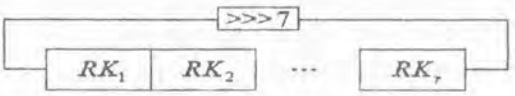

Рис.5. Способ модификации общего ключа зашифрования

Тем самым образуются обновленные базовые раундовые ключи, но процесс зашифрования выполняется точно так же, как и на первом шаге. И так далее, пока не исчерпаются все шаги зашифрования.

Инверсный шифр. На этапе расшифрования исходный Common Key (Master Key) сначала должен быть подвергнут циклическому сдвигу на $7*(s-1)$ бит влево, где s – число шагов шифрования. А затем на каждом новом шаге расшифрования СК циклически сдвигается на семь бит вправо. При этом если число раундов (для каждого шага) является четным числом (именно такая стратегия принята для RSB алгоритма), то процесс расшифрования начинается с последнего (самого правого) зашифрованного блока и протекает по направлению справа налево, а в качестве базового раундового ключа служит последняя в СК 32-битная кодовая комбинация. Естественно, что последовательность выполняемых примитивов на этапе расшифрования должна быть обратной последовательности примитивов зашифрования. Точно так же, как все примитивы на этапе расшифрования должны быть обратными примитивам зашифрования.

Предлагаемый RSB алгоритм закладывает реальную основу для создания новой технологии симметричной блочной криптографической защиты информации в компьютерных сетях. Реализация данного проекта позволит существенно повысить криптостойкость систем шифрования по сравнению с уже существующими продуктами. Достижение отмеченного качества (криптостойкости) базируется на таких предпосылках. В сложившейся мировой практике построения симметричных блочных криптографических алгоритмов в пределах раунда все блоки шифруемого текста подвергаются одинаковым преобразованиям. С одной стороны это обеспечивает возможность параллельной обработки информации, что повышает скорость шифрования. Вместе с тем если, например, в открытом тексте присутствуют одинаковые блоки, то одинаковыми будут также эти блоки после зашифрования, что облегчает работу криптоаналитиков. Отмеченный недостаток классических блочных шифраторов устраняется RSB технологией за счет применения двунаправленного скользящего кодирования, посредством которого каждый зашифровуемый блок текста становится управляемым своим *индивидуальным* локальным блочным раундовым ключом, зависящим не только от базового раундового ключа, но и всей истории шифрования, предшествующей преобразуемому блоку.

2. Краткое описание и пример вычислений в BABY-RSB алгоритме

Мини-версию RSB алгоритма будем в дальнейшем называть BABY-RSB алгоритмом или, для краткости, просто BABY. Как уже отмечалось выше размер блока N и длина базового раундового ключа r в BABY-алгоритме приняты равными 16 бит, а размеры элементов скользящего кодирования и нелинейной замены – 4 бита. Каждый раунд зашифрования включает ту же самую последовательность криптографических примитивов, что и полная версия RSB шифра. Данные преобразования параметризуются с помощью блочного раундового ключа, структурная схема которого приведена на рис. 6.

Рис.6. Структурная схема блочного раундового ключа BABY-RSB шифра

Если число шагов зашифрования $s>1$, то при переходе к очередному шагу криптопреобразований шаговый ключ шифрования (СК) подвергается модификации, которая сводится к циклическому сдвигу СК на три разряда влево.

Ниже приведен числовой пример, иллюстрирующий процесс зашифрования 16-битного текста BABY-RSB криптоалгоритмом для параметров шифрования $r=s=1$.

Итак, выберем в качестве открытого текста двоичный вектор

$$T=1010\ \ 0111\ \ 1101\ \ 0011. \tag{6}$$

Пусть, к тому же, ключ зашифрования (СК) имеет вид

$$\text{СК} = \boxed{C}\boxed{\alpha}\boxed{\beta}\boxed{\gamma} = 1100\ 1010\ 1100\ 0011 \tag{7}$$

Параметр круговой прокрутки (по часовой стрелке) блока на этапе зашифрования задан сектором C блочного ключа СК. Сначала считываются три младших разряда полубайта C, а затем в самый младший разряд трехбитного слова записывается единица. Сформированная кодовая комбинация как раз и определяет параметр (величину) круговой прокрутки.

Выполнив над T циклический сдвиг на пять разрядов влево, что предопределено параметром C в (7), получим вектор

$$T'=1111\ 1010\ 1101\ 0101 \tag{8}$$

Процесс левостороннего скользящего кодирования над T' отображен на рис. 7.

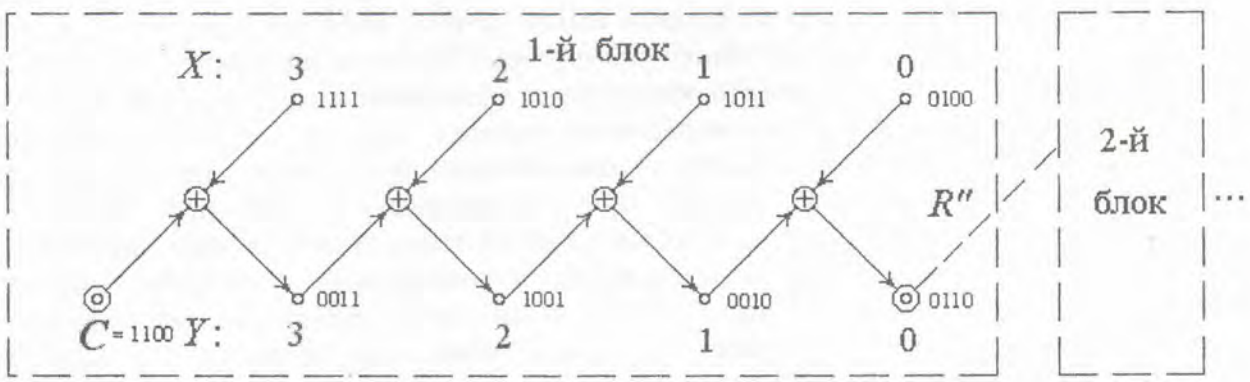

Рис.7. Процесс левостороннего скользящего кодирования над блоком T'

Данным криптопримитивом сформирован локальный раундовый ключ (вектор)

$$y_3y_2y_1y_0 = R'' = 0011\ 1001\ 0010\ 0110, \tag{9}$$

под управлением которого может выполняться преобразование второго блока текста, если таковой имеется.

Обратим внимание на то, что если в полной версии RSB алгоритма на вход схемы скользящего кодирования (рис. 3) подается весь блочный базовый раундовый ключ R', то в BABY-RSB (рис. 7) лишь полубайтная компонента C ключа (7).

Нелинейная подстановка (1) полубайтов (9) выполняется за два этапа. Сначала вычисляется компонента

$$y=(x\oplus\alpha)^{-1}_{\varphi_s}\otimes A_s, \tag{10}$$

в которой матрица преобразования A_s формируется составным кодом Грея (СКГ) $G = 1353$ и имеет вид:

$$A_s = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}. \qquad (11)$$

На основании выражений (10) и (11), выбрав в качестве неприводимого полинома $\varphi_s = 10011$, приходим к табличной форме (табл. 1) преобразования (10)

Таблица 1. Нелинейное преобразование (10)

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Y	B	7	4	9	F	A	3	1	C	2	0	6	E	8	D	5

На втором этапе вычисления нелинейной подстановки полубайты (10) переопределяются

$$y = y \oplus \beta, \qquad (12)$$

причем $\beta = 1100$ берется из ключа (7). С учетом (12) проходим к полной форме (1) нелинейной подстановки (табл. 2) в первом блоке

Таблица 2. Нелинейная подстановка (1)

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Y	7	B	8	5	3	6	F	D	0	E	C	A	2	4	1	9

Подвергая нелинейному преобразованию с помощью табл. 2 вектор (9), получим

$$Y = Y_3 Y_2 Y_1 Y_0 = 0101\ 1110\ 1000\ 1111 \qquad (13)$$

Перестановка битов (13) также реализуется за два этапа. Адрес ячейки y, в которую перемещается содержимое ячейки x, сначала определяется соотношением

$$y = (x \oplus \beta)^{-1}_{\varphi_p} \otimes \grave{A}_p, \qquad (14)$$

в котором $\varphi_p = 11001$, $\beta = 1100$, а матрицу A_p будем задавать СКГ = 1535, т.е.

$$A_p = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Таблица перестановки, вычисляемая программой Baby Per и отвечающая соотношению (14), имеет вид

Таблица 3. Нелинейное преобразование (14)

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Y	8	3	4	C	D	A	B	E	1	2	9	7	0	5	F	6

Затем фактический адрес перестановки переопределяется по формуле

$$y = y \oplus \gamma, \qquad (15)$$

причем параметр γ берется из базового раундового ключа (1), т.е. $\gamma = 0011$. Конечная форма таблицы перестановки (табл. 4) образуется поразрядным суммированием элементов Y табл. 3 с параметром $\gamma = 0011$.

Таблица 4. Финальный алгоритм перестановки битов

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Y	B	0	7	F	E	9	8	D	2	1	A	4	3	6	C	5

Выполнив перестановку вектора (13) по табл. 4, получим вектор

$$\hat{A} = 1011\ 0110\ 1100\ 1011,$$

что и завершает процесс криптопреобразования входного 16-битного текста (6).

3. Методика исследований

Как уже было отмечено выше, главная задача наших исследований состоит в том, чтобы представить RSB шифр в виде одной большой подстановки и оценить дифференциальную характеристику, покрывающую весь шифр. Другими словами, в качестве подстановки мы принимаем весь набор шифрующих преобразований для одного ключа. Будем рассматривать мини-версии RSB шифра с 16-битным ключом. В качестве входного (открытого) текста выберем двоичный массив, содержащий 2^{16} блоков всевозможных 16-битных кодовых слов, начиная с комбинации 00…00 и заканчивая комбинацией 11…11. Следовательно, общее число различных подстановок равно мощности ключевого пространства 2^{16} и для каждого ключа таблица дифференциалов имеет размер $2^{16}\times 2^{16}$, занимая 4 Гбайта памяти компьютера.

Перейдем к развернутому изложению технологии вычисления таблиц дифференциалов для мини-версии RSB шифра. В программе, которую условно назовем DifAn, формируется входной текст X (для последующего зашифрования с помощью Baby RSB), представляющий собой одномерный массив, содержащий 2^16 двухбайтных слов, начиная с 0000000000000000 и до 1111111111111111. То есть, число блоков N открытого текста **X** равно 2^16. Именно этот входной текст X подвергается зашифрованию с помощью Baby RSB. На выходе шифратора образуется выходной текст Y (криптограмма или шифрограмма), также содержащий N = 2^16 двухбайтных слов. В DifAn резервируется матрица dXY, содержащая 2^16 строк и 2^16 столбцов. Строками матрицы являются дифференциалы (разницы) dX входных слов, а столбцами – разницы dY выходных слов (блоков шифрограммы) Y. Элементами матрицы dXY также являются 16-ти битные слова.

Дифференциалами (разницами) dX и dY называют поразрядную сумму по модулю два любой пары входных (X_i, X_j) и выходных (Y_k, Y_l) слов, т.е.

$$dX_{ij} = X_i \oplus X_j; \qquad dY_{kl} = Y_k \oplus Y_l.$$

В качестве примера рассмотрим вычисления дифференциалов для трехразрядных двоичных кодовых комбинаций. В первой слева колонке таблицы 5 приведены все трехбитные значения X_i входного текста X, во второй колонке – ранжированные (упорядоченные) значения всевозможных дифференциалов пар входных блоков, а в оставшихся столбцах указаны упорядоченные всевозможные значения дифференциалов dY пар выходных блоков шифрограммы.

Для упрощения набора введем другие обозначения. Вместо dX_{ij} будем писать X(i, j). Аналогично вместо dY_{kl} будем писать Y(k, l). Рассмотрим такой пример (табл. 6) криптопреобразования.

Таблица 5

X_i	dX	dY							
		000	001	010	011	100	101	110	111
X_0=000	000								
X_1=001	001								
X_2=010	010								
X_3=011	011								
X_4=100	100								
X_5=101	101								
X_6=110	110								
X_7=111	111								

Таблица 6

	X		Y	
X0	000		110	Y0
X1	001		010	Y1
X2	010		001	Y2
X3	011		111	Y3
X4	100		101	Y4
X5	101		000	Y5
X6	110		100	Y6
X7	111		011	Y7

Перед началом заполнения элементы матрицы dXY обнуляются.

Разность X(i,j) переходит в разность Y(i,j). Поясним на примере. Пусть i = 2, j = 5. В соответствии с табл. 6, X(2, 5) = X2 ⊕ X5 = 010 ⊕ 101 = 111. В свою очередь,

У(2, 5) = У2 ⊕ У5 = 001 ⊕ 000 = 001. Это означает, что входная разница dX = 111 переходит в выходную разницу dY = 001. Элемент матрицы (у нас это таблица 5), находящийся на пересечении строки dX = 111 и столбца dY = 001 увеличивается на единицу.

Вычислим X(0, i) и Y(0, i) для всех значений i от 0 до 7. Имеем

$$
\begin{aligned}
&X(0,0) = 000 \oplus 000 = 000 \Rightarrow Y(0,0) = 110 \oplus 110 = 000;\\
&X(0,1) = 000 \oplus 001 = 001 \Rightarrow Y(0,1) = 110 \oplus 010 = 100;\\
&X(0,2) = 000 \oplus 010 = 010 \Rightarrow Y(0,2) = 110 \oplus 001 = 111;\\
&X(0,3) = 000 \oplus 011 = 011 \Rightarrow Y(0,3) = 110 \oplus 111 = 001;\\
&X(0,4) = 000 \oplus 100 = 100 \Rightarrow Y(0,4) = 110 \oplus 101 = 011;\\
&X(0,5) = 000 \oplus 101 = 101 \Rightarrow Y(0,5) = 110 \oplus 000 = 110;\\
&X(0,6) = 000 \oplus 110 = 110 \Rightarrow Y(0,6) = 110 \oplus 100 = 010;\\
&X(0,7) = 000 \oplus 111 = 111 \Rightarrow Y(0,7) = 110 \oplus 011 = 101.
\end{aligned}
\qquad (16)
$$

Переносим эти данные в табл. 5. Получим

Таблица 7

X_i	dX	dY							
		000	001	010	011	100	101	110	111
X_0=000	000	1							
X_1=001	001					1			
X_2=010	010								1
X_3=011	011		1						
X_4=100	100				1				
X_5=101	101							1	
X_6=110	110			1					
X_7=111	111						1		

Выполним аналогичные вычисления для X(1, i) и Y(1, i)

$$
\begin{aligned}
&X(1,0) = 001 \oplus 000 = 001 \Rightarrow Y(1,0) = 010 \oplus 110 = 100;\\
&X(1,1) = 001 \oplus 001 = 000 \Rightarrow Y(1,1) = 010 \oplus 010 = 000;\\
&X(1,2) = 001 \oplus 010 = 011 \Rightarrow Y(1,2) = 010 \oplus 001 = 011;\\
&X(1,3) = 001 \oplus 011 = 010 \Rightarrow Y(1,3) = 010 \oplus 111 = 101;\\
&X(1,4) = 001 \oplus 100 = 101 \Rightarrow Y(1,4) = 010 \oplus 101 = 111;\\
&X(1,5) = 001 \oplus 101 = 100 \Rightarrow Y(1,5) = 010 \oplus 000 = 010;\\
&X(1,6) = 001 \oplus 110 = 111 \Rightarrow Y(1,6) = 010 \oplus 100 = 110;\\
&X(1,7) = 001 \oplus 111 = 110 \Rightarrow Y(1,7) = 010 \oplus 011 = 001.
\end{aligned}
\qquad (17)
$$

Из анализа предыдущей системы приходим к совершенно очевидным выводам:

$$1.\ X(i, i) = 000 \quad \Rightarrow \quad Y(i, i) = 000;$$
$$2.\ X(i, j) = X(j, i) \quad \Rightarrow \quad Y(i, j) = Y(j, i).$$

Из системы (16) следует, что нет никакой необходимости проводить вычисления указанного типа; можно сразу в верхнюю левую клетку Таблицы 7 (элемент матрицы dXY) записать 8, а в оставшиеся клетки (элементы) верхней строки и левого столбца таблицы (матрицы) записать нули. Это, во-первых, а во-вторых, из соотношений (17) также следует, что если проведено вычисление (i, j), то в соответствующем элементе матрицы можно сразу плюсовать двойку, не проводя вычислений (j, i), в силу того что они порождают одинаковые разности, т.е. достаточно ограничиться вычислениями (i, j), при $i < j$. Рекомендации, высказанные в предыдущем абзаце, позволяют вдвое сократить объем вычислений. В частности, табл. 7 на основании системы (1) можно представить в виде

Таблица 8

X_i	dX	dY							
		000	001	010	011	100	101	110	111
X_0=000	000	8							
X_1=001	001					2			
X_2=010	010								2
X_3=011	011		2						
X_4=100	100				2				
X_5=101	101							2	
X_6=110	110			2					
X_7=111	111						2		

Исключим из системы (17) две верхние строчки

$$\begin{aligned}
&X(1, 2) = 001 \oplus 010 = 011 \ \Rightarrow \ Y(1, 2) = 010 \oplus 001 = 011;\\
&X(1, 3) = 001 \oplus 011 = 010 \ \Rightarrow \ Y(1, 3) = 010 \oplus 111 = 101;\\
&X(1, 4) = 001 \oplus 100 = 101 \ \Rightarrow \ Y(1, 4) = 010 \oplus 101 = 111;\\
&X(1, 5) = 001 \oplus 101 = 100 \ \Rightarrow \ Y(1, 5) = 010 \oplus 000 = 010;\\
&X(1, 6) = 001 \oplus 110 = 111 \ \Rightarrow \ Y(1, 6) = 010 \oplus 100 = 110;\\
&X(1, 7) = 001 \oplus 111 = 110 \ \Rightarrow \ Y(1, 7) = 010 \oplus 011 = 001.
\end{aligned} \qquad (18)$$

Дополнив табл. 8 данными системы (18) получим

Таблица 9

X_i	dX	dY							
		000	001	010	011	100	101	110	111
X_0=000	000	8							
X_1=001	001					2			
X_2=010	010						2		2
X_3=011	011		2		2				
X_4=100	100			2	2				
X_5=101	101							2	2
X_6=110	110		2	2					
X_7=111	111						2	2	

Продолжив вычисления с учетом приведенных выше рекомендаций, приходим к окончательной таблице разностей (табл. 10), в которой исключены (затенены) верхняя строка и левый столбец, не участвующие в дальнейших вычислениях.

Таблица 10

X_i	dX	dY							
		000	001	010	011	100	101	110	111
X_0=000	000								
X_1=001	001					2	2	2	2
X_2=010	010		2		2		2		2
X_3=011	011		2		2	2		2	
X_4=100	100			2	2	2	2		
X_5=101	101			2	2			2	2
X_6=110	110		2	2		2			2
X_7=111	111		2	2			2	2	

Дальнейшая обработка массива разностей проводится по следующему алгоритму. Необходимо сформировать одномерную матрицу (столбец) MAX, содержащие N = 2^16 элементов. С этой целью вычисляется максимум в каждой строке таблицы 10 и его значение переносится в соответствующий элемент столбца MAX. Далее определяются абсолютный максимум по всей таблице (Max Max), минимум максимумов (Мин Max) и среднее значение максимумов (Ср. Max) столбца MAX, которые записываются в соответствующих трех правых нижних элементах таблицы MAX. Для данных, представленных в табл.10, окончательная форма таблицы MAX такова.

Таблица 11

X_i	dX	dY								MAX
		000	001	010	011	100	101	110	111	
X_0=000	000									
X_1=001	001					2	2	2	2	2
X_2=010	010		2		2		2		2	2
X_3=011	011		2		2	2		2		2
X_4=100	100			2	2	2	2			2
X_5=101	101			2	2			2	2	2
X_6=110	110		2	2		2			2	2
X_7=111	111		2	2			2	2		2
Максимум максиморум (Max Max)										2
Минимум максимумов (Мин Max)										2
Средний максимум (Ср. Max)										2

Результаты эксперимента

Дифференциальные характеристики рассчитывались для трех вариантов моделей BABY-RSB шифров, которые обозначим Baby-RSB1, Baby-RSB2 и Baby-RSB3 соответственно. Различие вариантов Baby-RSB1 и Baby-RSB2 состоит в следующем. В варианте 1 для всех шифруемых блоков в пределах раунда сохраняются фиксированными как аддитивные компоненты α (для примитива нелинейной подстановки), так и компоненты β (для примитива стохастической перестановки). Эти значения определяются соответствующими секторами базового раундового ключа. В варианте 2 компоненты α и β меняются от блока к блоку согласно конкретным значениям, образуемыми локальными

блочными раундовыми ключами шифрования примитивом скользящего кодирования. Общим для вариантов шифров Baby-RSB1 и Baby-RSB2 является то, что операция скользящего кодирования выполняется по всему шифруемому тексту по направлению слева направо (левостороннее кодирование) для всех нечетных раундов и справа налево (правостороннее кодирование) для всех четных раундов шифрования.

В варианте Baby-RSB3 скользящее кодирование выполняется только в пределах шифруемого блока, сохраняя, как и в первых двух вариантах Baby-шифра, направление кодирования в зависимости от того, четным или нечетным является раунд шифрования. Тем самым, вариант Baby-RSB3 в полной мере отвечает классификационным признакам, характерным симметричным блочным криптоалгоритмам. Следовательно, версия RSB шифратора, построенная по аналогии с вариантом Baby-RSB3, допускает реализацию известных режимов использования БСШ. К таким режимам относятся: *режим простой замены*, известный под названием *режима электронной кодовой книги – ECB* (Electronic Code Book), *режим сцепления блоков шифрования – CBC* (Ciphertext Block Chaining), *режим обратной связи по выходу – OFB* (Output Feedback) и др. Естественно, что ни один из этих режимов не может быть напрямую применен в основных версиях RSB, аналогичных вариантам шифров Baby-RSB1 и Baby-RSB2.

В ряде исследований предложен подход к оценке эффективности БСШ в виде минимального числа циклов алгоритма, при котором реализуется асимптотический показатель среднего значения максимума полных дифференциалов, который и был положен в основу наших исследований. В частности, как показано в [11], для 16-битного шифра, а именно этот порядок принят для мини-версий RSB, упомянутый выше асимптотический показатель лежит в пределах 18-20. Дифференциальная характеристика шифра как подстановки меняется от раунда к раунду. Однако после некоторого числа раундов шифрующего преобразования дифференциальная характеристика шифра будет стремиться к дифференциальной характеристике случайной подстановки. Для рассматриваемых мини-версий RSB – к асимптотическому значению максимума полных дифференциалов, равному 18-20. И чем за меньшее число раундов достигается асимптотика, тем выше считаются качество шифратора.

Финальные оценки зависимости выбранных дифференциальных характеристик для всех трех вариантов Baby-RSB таковы. В вариантах Baby-RSB1 и Baby-RSB2 асимптотика достигается за два раунда шифрования, тогда как для варианта Baby-RSB3 – за шесть раундов.

Выводы

Малое число раундов шифрования, за которое достигаются асимптотические значения максимума полных дифференциалов в мини версиях алгоритмов Baby-RSB1 и Baby-RSB2, является свидетельством того, что базовый вариант RSB шифра, в котором реализовано скользящее кодирование всего шифруемого текста, обеспечивает достаточно высокую криптостойкость к атакам, основанных на дифференциальном анализе. Вместе с тем, платой за повышение криптостойкости являются более высокие вычислительные ресурсы (по сравнению с другими БСШ – участниками открытого конкурса [12] на разработку национального стандарта блочного симметричного шифрования Украины [13-16]), необходимые для реализации RSB.

Список литературы

1. L. J. O'Connor. On the Distribution of Characteristics in Bijective Mappings. Advances in Cryptology. EUROCRIPT 93, Lecture Notes in Computer Science, vol. 795, T. Helleseth ed., Springer-Verlag, pages 360–370, 1994.

2. Долгов В.И., Кузнецов А.А., Лисицкая И.В., Сергиенко Р.В., Олешко О.И. Исследование криптографических свойств нелинейных узлов замены уменьшенных версий некоторых шифров // Прикладная радиоэлектроника. – Харьков: ХНУРЭ. – 2009. – Том 8. №3. – С.268 – 277.

3. Белецкий А.Я., Белецкий А.А. Симметричный блочный RSB-32 криптоалгоритм // Захист інформації. – Київ: ДУІКТ. – 2006, № 2. – С. 42 – 51.

4. Белецкий А.Я., Белецкий А.А., Кузнецов А.А. Семейство симметричных блочних RSB криптографических алгоритмов с динамически управляемыми параметрами шифрования // Електроніка та системи управління. - 2007. - № 1 (11). - С. 5-16.

5. Молдавян А.А., Молдавян Н.А., Гуц Н.Д., Иванов Б.В. Криптгграфия: скоростные шифры. СПб.: БХВ-Петербург, 2002. – 496 с.

6. Молдавян Н.А., Молдавян А.А., Еремеев М.А. Криптгграфия: от примитивов к синтезу алгоритмов. СПб.: БХВ-Петербург, 2004. – 448 с.

7. Белецкий А.Я., Белецкий А.А., Белецкий Е.А. Преобразования Грея. – К.: Изд-во НАУ. – Т.1 Основы теории, 2007. – 412 с.

8. Белецкий А.Я., Аксентий Е.А. Программный комплекс для исследования криптографических примитивов типа нелинейной подстановки. // Сучасний захист інформації. – Київ: ДУІКТ. – 2010, № 2. – С. 30 – 41.

9. Зензин О.С., Иванов М.А. Стандарт криптографической защиты – AES. Конечные поля. – М.: КУДИЦ-ОБРАЗ, 2002. – 176 с.

10. Белецкий А.Я., Аксентий Е.А. Программный комплекс для исследования статистических характеристик криптографических примитивов типа перестановки элементов шифруемого блока. // Сучасний захист інформації. – Київ: ДУІКТ. – 2010, № 1. – С. 43 – 53.

11. Kuznetsov A., Sergienko R., Isaev S., Laptii P. Differential Properties of Mini-Versions of Block Symmetric Ciphers // The materials of International Conferense «Statiatical Methods of Signal and Data Processing (SMSDP-2010).– Publishing of National Aviation University «NAU-Druk», Kiev, 2010. – P. 147-150.

12. Положення про проведення відкритого конкурсу криптографічних алгоритмів. http://dstszi.gov.ua/dstszi/control/uk/publish/

13. Горбенко І.Д., Долгов В.І., Олійников Р.В. та ін. Перспективний блоковий симетричний шифр «КАЛИНА». Основні положення та специфікація. // Прикладная радиоэлектроника, 2007. – Т. 6, № 2. - С. 195-208.

14. Горбенко І.Д., Долгов В.І., Олійников Р.В. та ін. Перспективний блоковий симетричний шифр «Мухомор». Основні положення та специфікація. // Прикладная радиоэлектроника, 2007. – Т. 6, № 2. – С. 147-157.

15. Головашич С.А. Спецификация алгоритма блочного симметричного шифрования «Лабиринт». //Прикладная радиоэлектроника, 2007. – Т. 6, № 2. - С. 230-240.

16. Кузнецов А.А., Сергиенко Р.В., Наумко А.А. Симметричный криптографический алгоритм ADE (Algorithm of Dynamic Encryption) // Прикладная радиоэлектроника. – Харьков: ХНУРЭ. – 2007. – Т. 6. №2. – С.241 – 249.

Наведено загальний опис RSB шифратора. Викладена методика оцінки диференціальних характеристик міні-версії алгоритму шифрування. Показано, що асимптотичного значення максимуму повних диференціалів у міні-версіях RSB досягається за два раунди криптоперетворень.
Ключові слова: шифри, криптоаналіз, диференціальний метод.

Приведено общее описание RSB шифратора. Изложена методика оценки дифференциальных характеристик мини-версии алгоритма шифрования. Показано, что асимптотическое значение максимума полных дифференциалов в мини-версиях RSB достигается за два раунда криптопреобразований.
Ключевые слова: шифры, криптоанализ, дифференциальный метод.

A general description RSB encoder. The technique of assessing the differential characteristics of mini-version of the encryption algorithm. It is shown that the asymptotic value of the maximum total differentials in the mini-versions of the RSB is achieved by two rounds of encryptions.
Keywords: ciphers, cryptanalysis, differential method.

Рецензент: Шелест М.Є.
Надійшла 20.10.2010