6. *Погорелов Б. А., Пудовкина М. А.* Вариации ортоморфизмов и псевдоадамаровых преобразований на неабелевой группе // Прикладная дискретная математика. Приложение. 2019. № 12. С. 24–27.
7. *Погорелов Б. А., Пудовкина М. А.* О классе степенных кусочно-аффинных подстановок на неабелевой группе порядка 2^m, обладающей циклической подгруппой индекса два // Прикладная дискретная математика. Приложение. 2019. № 12. С. 27–29.
8. *Погорелов Б. А., Пудовкина М. А.* Неабелевость группы наложения ключа и свойство $\otimes_{\mathbf{W}}$-марковости алгоритмов блочного шифрования // Матем. вопр. криптогр. 2020. Т. 11. № 4. С. 3–22.
9. *Холл М.* Теория групп. М.: ИЛ, 1962. 468 с.
10. *Dixon J. D. and Mortimer B.* Permutation Groups. Berlin: Springer Verlag, 1996. 346 p.
11. *Grossman E.* Group Theoretic Remark on Cryptographic System Based on Two Types of Additions. Math. Sc. Dept. IBM Watson res. Center Yorktown Heights, 1974.
12. *Погорелов Б. А., Пудовкина М. А.* Надгруппы аддитивных регулярных групп порядка 2^m кольца вычетов и векторного пространства // Дискретная математика. 2015. Т. 27. № 3. С. 74–94.
13. *Бабаш А. В., Шанкин Г. П.* Криптография. М.: СОЛОН-Р, 2002. 512 с.
14. *Paterson K. G.* Imprimitive permutation groups and trapdoors in iterated block ciphers // LNCS. 1999. V. 1636. P. 201–214.
15. *Погорелов Б. А.* Примитивные группы подстановок, содержащие 2^m-цикл // Алгебра и логика. 1980. Т. 19. № 2. С. 236–247.

УДК 519.7 DOI 10.17223/2226308X/14/23

ПОРОЖДЕНИЕ ДОПОЛНИТЕЛЬНЫХ ОГРАНИЧЕНИЙ В ЗАДАЧАХ АЛГЕБРАИЧЕСКОГО КРИПТОАНАЛИЗА ПРИ ПОМОЩИ SAT-ОРАКУЛОВ[1]

А. А. Семёнов, К. В. Антонов, И. А. Грибанова

Описывается новая техника, предназначенная для дополнения исходной системы ограничений в задаче алгебраического криптоанализа новыми ограничениями. Порождаемые ограничения могут иметь форму линейных уравнений над полем из двух элементов в случае, если задача криптоанализа сведена к квадратичной системе над GF(2). Если же рассматриваемая задача сведена к SAT, то порождаемые ограничения имеют вид эквивалентностей или единичных резольвент. Для обеих ситуаций мы показываем, что порождаемые ограничения могут снижать оценки трудоёмкости криптоанализа.

Ключевые слова: *алгебраический криптоанализ, проблема булевой выполнимости (SAT), квадратичные системы уравнений над GF(2), SAT-оракул.*

1. Предварительные результаты

Везде далее под термином «ограничение» понимается либо линейное уравнение над GF(2), либо дизъюнкт. Рассматривается задача обращения функции вида

$$f : \{0,1\}^n \to \{0,1\}^m, \tag{1}$$

заданной некоторым алгоритмом A_f: то есть требуется, зная A_f и произвольное $\gamma \in \in \operatorname{Range} f \subseteq \{0,1\}^m$, найти такое $\alpha \in \{0,1\}^n$, что $f(\alpha) = \gamma$.

[1]Грибанова И. А. поддержана стипендией Президента РФ (СП-3545.2019.5).

В алгебраическом криптоанализе [1] задача обращения (1) сводится к проблеме поиска решений алгебраических уравнений над некоторым конечным полем либо к проблеме выполнимости булевой формулы, обычно в конъюнктивной нормальной форме (КНФ). Везде далее рассматривается поле GF(2). Такая сводимость в теории является эффективной (следствие теоремы Кука — Левина [2, 3]). Более того, существует ряд программных систем, реализующих данную сводимость в отношении функций вида (1), заданных в виде императивных или функциональных программ. Упомянем здесь системы CBMC [4], Cryptol+SAW [5], Transalg [6], а также систему, описанную в [7]. Во всех представленных далее вычислительных экспериментах использовалась система Transalg. Данная система, получая на вход описание функции вида (1) на C-подобном языке, строит шаблонную КНФ C_f [8]. Неформально говоря, КНФ C_f представляет работу алгоритма A_f на всех возможных входах из $\{0,1\}^n$. Более точно, пусть $\alpha = (\alpha_1, \ldots, \alpha_n)$ — произвольный набор из $\{0,1\}^n$, рассмотрим следующую КНФ:

$$C_f(\alpha) = x_1^{\alpha_1} \wedge \ldots \wedge x_n^{\alpha_n} \wedge C_f, \tag{2}$$

где обозначение x^σ понимается в том же смысле, что и в [9]. Если применить к (2) правило распространения единичных дизъюнктов (Unit Propagation Rule, UP [10]), то за линейное от длины записи C_f время по UP будут выведены значения всех переменных, входящих в C_f, в том числе и значения переменных $y_1, \ldots, y_m$, кодирующих выход функции (1). Каждой выводимой по UP переменной соответствует некоторая элементарная операция, выполняемая алгоритмом A_f, и, таким образом, применение UP к (2) можно рассматривать как способ вычисления $f(\alpha)$.

Рассмотрим теперь произвольное $\gamma \in \operatorname{Range} f$, $\gamma = (\gamma_1, \ldots, \gamma_m)$, и построим следующую КНФ:

$$C_f(\gamma) = y_1^{\gamma_1} \wedge \ldots \wedge y_m^{\alpha_m} \wedge C_f.$$

Поскольку $\gamma \in \operatorname{Range} f$, то $C_f(\gamma)$ выполнима, а из выполняющего её набора эффективно извлекается такой $\alpha \in \{0,1\}^n$, что $f(\alpha) = \gamma$ [8].

Разберём процесс построения C_f в рамках процедур, используемых в системе Transalg. В данной системе вычисление функции f на произвольном входе из $\{0,1\}^n$ представляется в виде последовательности простых операций с ячейками памяти специальной абстрактной машины (АМ). Каждой такой операции сопоставляется булева формула следующего вида:

$$y \equiv g(\tilde{x}_1, \ldots, \tilde{x}_s). \tag{3}$$

Здесь y — булева переменная, которая вводится на рассматриваемом шаге трансляции программы A_f; $\tilde{x}_1, \ldots, \tilde{x}_s$ — переменные, введённые на предыдущих шагах работы программы. Формула (3) соответствует вычислению значения некоторого оператора от битов, находящихся в ячейках памяти АМ, с которыми связаны переменные $\tilde{x}_1, \ldots, \tilde{x}_s$.

Определение 1. Функцию $g(\tilde{x}_1, \ldots, \tilde{x}_s)$, находящуюся в правой части произвольного соотношения вида (3), будем называть t-гейтом. Переменные, фигурирующие в (3), будем называть переменными рассматриваемого t-гейта.

Современные символьные трансляторы типа CBMC или Transalg позволяют задать функцию (1) в виде специального графа G_f, который есть, по сути, схема из функциональных элементов над базисом $\{\neg, \wedge\}$. Граф G_f известен как AIG (And Invertor Graph) [11]. Если произвольный гейт g в G_f — AND-гейт, то с g связывается формула вида $\varphi_\wedge : y \equiv \tilde{x}_1 \wedge \tilde{x}_2$, если g — NOT-гейт (Invertor), то формула вида $\varphi_\neg : y \equiv \neg\tilde{x}_1$:

переменная y приписана рассматриваемому гейту, а переменные $\tilde{x}_1, \tilde{x}_2$ — гейтам, данные с которых подаются на вход g. Заданные так формулы имеют тот же смысл, что и (3). Если все формулы вида $\varphi_\wedge$ и $\varphi_\neg$ перевести в КНФ над множествами входящих в них переменных, а потом соединить все полученные КНФ конъюнкцией, то получится КНФ $\tilde{C}_f$, аналогичная по свойствам КНФ C_f. Будем называть $\tilde{C}_f$ шаблонной КНФ, порождённой графом G_f.

С другой стороны, мы можем сопоставить формулам $\varphi_\wedge$ и $\varphi_\neg$ алгебраические уравнения над полем $\mathrm{GF}(2) = \langle\{0,1\}, \oplus, \wedge\rangle$: $e_\wedge$: $\tilde{x}_1 \wedge \tilde{x}_2 \oplus y = 0$ и $e_\neg$: $\tilde{x}_1 \oplus y = 1$ соответственно. Между множествами решений таких уравнений и множествами наборов, выполняющих формулы $\varphi_\wedge$ и $\varphi_\neg$, существует очевидная биекция: пусть α — набор, выполняющий формулу $\varphi_\wedge$ или $\varphi_\neg$; если рассматривать компоненты набора α как элементы поля $\mathrm{GF}(2)$, то α является решением уравнения $e_\wedge$ или $e_\neg$. Учитывая данный факт, везде далее будем обозначать переменные, входящие в формулы вида $\varphi_\wedge$, $\varphi_\neg$ и в уравнения $e_\wedge$, $e_\neg$, одинаковыми буквами.

Объединим все уравнения вида $e_\wedge$, $e_\neg$, построенные по графу G_f, в систему, которую обозначим через E_f и назовём шаблонной системой для функции f. Очевидно, что E_f состоит из уравнений над $\mathrm{GF}(2)$ степени не более 2 и, таким образом, является MQ-системой (Multivariate Quadratic equation system [12]). Для системы E_f справедливы свойства, похожие на свойства шаблонной КНФ C_f: 1) если подставить в E_f произвольный набор $\alpha \in \{0,1\}^n$ и применить простые правила преобразования уравнений [13], рассматриваемые как правила вывода, то итогом такого вывода будут значения всех переменных, входящих в E_f, и, в том числе, такой набор $y_1 = \gamma_1$, …, $y_m = \gamma_m$, что $f(\alpha) = \gamma$, $\gamma = (\gamma_1, \ldots, \gamma_m)$; 2) если подставить в систему E_f произвольный $\gamma \in \operatorname{Range} f$, то полученная система $E_f(\gamma)$ является совместной и из любого её решения можно эффективно выделить такой $\alpha \in \{0,1\}^n$, что $f(\alpha) = \gamma$.

2. Применение SAT-оракулов для порождения дополнительных ограничений в задаче обращения (1)

В данной части мы развиваем идеи из [14] в нескольких направлениях, а именно: мы описываем новый подход, в рамках которого SAT-оракулы используются для порождения новых соотношений между переменными, присутствующими как в C_f, так и в E_f: в случае C_f генерируемые соотношения имеют вид формул $x \equiv y$, $x \equiv \neg y$ либо единичных дизъюнктов. В случае E_f мы строим новые линейные уравнения над $\mathrm{GF}(2)$. В отличие от подхода из [14], информация о функции f используется более полно.

В основе всех приведённых далее результатов лежит следующее простое наблюдение: если применить какой-либо эффективный на практике SAT-решатель, основанный на алгоритме CDCL, к шаблонным КНФ вида C_f или $\tilde{C}_f$, то этот решатель очень быстро (обычно за доли секунды) сгенерирует такую пару (α, γ), что $f(\alpha) = \gamma$.

Пусть g — некоторый t-гейт либо гейт в графе G_f. Заметим, что если g — NOT-гейт в G_f и ему приписаны переменные x, y и связывающее их соотношение $x \equiv \neg y$, то мы можем заменить все вхождения литерала y ($\neg y$) на вхождения литерала $\neg x$ (x), избавившись тем самым от переменной y в $\tilde{C}_f$. С учётом сказанного будем рассматривать в графе G_f только AND-гейты. Пусть g — произвольный AND-гейт с приписанной ему переменной y и входам g приписаны переменные x_1, x_2. Свяжем с g булеву функцию δ_g, принимающую значение 1 на любом наборе переменных из $\{x_1, x_2, y\}$, на котором истинна формула $y \equiv x_1 \wedge x_2$. Как известно, вклад, который даёт гейт g в $\tilde{C}_f$ в результате

преобразований Цейтина, — это следующая КНФ:

$$C_g : (x_1 \vee \neg y) \wedge (x_2 \vee \neg y) \wedge (\neg x_1 \vee \neg x_2 \vee y).$$

Другой способ получения C_g заключается в построении СКНФ $\hat{C}_g$ по таблице, задающей функцию δ_g, и переходе от $\hat{C}_g$ к C_g в результате минимизации.

Пусть теперь g — произвольный AND-гейт в G_f либо произвольный t-гейт типа (3). Функция δ_g, таблица $T(\delta_g)$ и СКНФ $\hat{C}_g$ для произвольного t-гейта g определяются так же, как и для AND-гейта. Пусть $\tilde{T}(\delta_g)$ — таблица, образованная строками $T(\delta_g)$, оставшимися после вычёркивания строк, по которым была построена КНФ C_g.

Пример 1. Пусть g — AND-гейт, тогда таблица $\tilde{T}(\delta_g)$ имеет следующий вид:

x_1	x_2	y	δ_g
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

Несложно понять, что для любого $\alpha \in \{0,1\}^n$ применение UP к КНФ вида (2) выведет набор значений переменных гейта g (t-гейт или AND-гейт в G_f), являющийся одной из строк таблицы $\tilde{T}(\delta_g)$.

Зафиксируем некоторое k, $1 \leqslant k \leqslant m$, и выберем конкретные k выходов функции f, которым в C_f или $\tilde{C}_f$ соответствуют переменные $y_{j_1}, \ldots, y_{j_k}$, $J = \{j_1, \ldots, j_k\} \subseteq \subseteq \{1, \ldots, m\}$.

Определение 2. Для фиксированного $J = \{j_1, \ldots, j_k\}$ определим функцию $f^J : \{0,1\}^n \to \{0,1\}^k$ следующим образом: на вход f^J получает произвольный $\alpha \in \{0,1\}^n$, значение $f^J(\alpha)$ образовано компонентами с номерами $j_1, \ldots, j_k$ в наборе $\gamma = f(\alpha)$. Произвольную функцию вида f^J назовём селектором ранга k (или кратко k-селектором) для f.

Очевидно, что для произвольного k, $1 \leqslant k \leqslant m$, существует $\binom{m}{k}$ различных k-селекторов для f.

Основной технический результат работы [14] состоит в следующем. Рассмотрим произвольный AND-гейт g и КНФ

$$x_1^{\sigma_1} \wedge x_2^{\sigma_2} \wedge y^{\sigma_3} \wedge \tilde{C}_f, \tag{4}$$

где $(\sigma_1\sigma_2\sigma_3)$ — произвольный набор значений переменных x_1, x_2, y из таблицы $\tilde{T}(\delta_g)$. Применим какой-либо эффективный SAT-решатель к КНФ (4). Оказывается, что (4) не является трудной для современных полных SAT-решателей, даже если f — криптографически стойкая функция. Ещё одно наблюдение состоит в том, что для многих AND-гейтов для некоторых строк таблицы $\tilde{T}(\delta_g)$ SAT-решатель быстро доказывает невыполнимость КНФ (4). Как показано в [14], в этом случае мы можем добавить к системе E_f новое линейное уравнение над GF(2).

Основную новизну настоящей работы составляет идея добавлять к КНФ типа (4) ещё и значения некоторых селекторов малого ранга k. То есть, грубо говоря, мы подставляем в (4) значения некоторых, но не всех бит выхода и применяем к такой КНФ SAT-решатель. Если он доказывает невыполнимость рассматриваемой КНФ, то это

означает, что набор $(\sigma_1\sigma_2\sigma_3)$ значений переменных x_1, x_2, y ни при каком входе не совместим с конкретными значениями тех бит выхода функции f, которые присутствуют в выбранном селекторе. Таким образом, для конкретного AND-гейта g можно перебрать все селекторы малого ранга (например, для $k \in \{1, 2, 3\}$), вызывая для каждого значения конкретного селектора и каждой строки таблицы $\tilde{T}(\delta_g)$ SAT-оракул O. Каждый раз, когда O доказывает невыполнимость соответствующей КНФ, получаем новое линейное уравнение над GF(2). Рассмотрим произвольный $\gamma \in \operatorname{Range} f$. Очевидно, что для любого k можно представить γ как набор селекторов ранга k. Например, для $k = 2$ и $\gamma \in \{0,1\}^m$ можно рассмотреть $\binom{m}{2}$ различных селекторов и их значений (двухбитовых векторов), определяемых набором γ. Следовательно, рассматривая задачу поиска прообраза γ при отображении f, мы можем использовать все линейные уравнения, которые были построены SAT-оракулом для соответствующих селекторов.

По аналогии с использованием k-селекторов для вывода линейных уравнений, дополняющих систему E_f, можно выводить при помощи селекторов и новые соотношения на переменные, присутствующие в шаблонной КНФ C_f. Рассмотрим соотношение (3) для произвольного t-гейта g и пусть $\tilde{T}(\delta_g)$ — таблица (сокращение таблицы, задающей δ_g) указанного вида, построенная для g. Пусть $(\sigma_1, \ldots, \sigma_{s+1})$ — произвольная строка таблицы $\tilde{T}(\delta_g)$. Рассмотрим шаблонную КНФ C_f, в которую подставим произвольные значения некоторого k-селектора, и обозначим результирующую КНФ через C'_f. Применим SAT-оракул O к следующей КНФ:

$$\tilde{x}_1^{\sigma_1} \wedge \ldots \wedge \tilde{x}_s^{\sigma_s} \wedge y^{\sigma_{s+1}} \wedge C'_f. \tag{5}$$

Если O доказал невыполнимость (5), то это означает, что при фиксированном значении выбранного селектора ни на каком входе функции f переменные $\tilde{x}_1, \ldots, \tilde{x}_s, y$ не примут значения $\sigma_1, \ldots, \sigma_{s+1}$. Соответственно такая строка вычёркивается из таблицы $\tilde{T}(\delta_g)$. В рассмотренных тестах таким способом могут быть вычеркнуты несколько строк таблицы при конкретном значении селектора. Результатом может быть, например, следующая ситуация (оставшиеся строки таблицы $\tilde{T}(\delta_g)$):

x_{3265}	x_{2820}	x_{2998}	x_{3176}	δ_g
0	0	0	0	1
0	1	1	0	1
0	0	0	1	1
0	1	1	1	1

В данном примере рассмотрена $\tilde{T}(\delta_g)$, полученная для некоторого t-гейта функции $g^{\lambda_1}_{\text{MD4-48}}$ из [15]. Изначально таблица $T(\delta_g)$ состояла из 16 строк, а таблица $\tilde{T}(\delta_g)$ — из 8. После проверки SAT-оракулом всех строк таблицы $\tilde{T}(\delta_g)$ на совместность с шаблонной КНФ осталось 4 строки. Из этого фрагмента таблицы видно, что переменная x_{3265} равна 0 при любом входе рассматриваемой функции. Кроме этого, для любого входа истинна формула $x_{2820} \equiv x_{2998}$. Таким образом, все вхождения переменной x_{2998} можно заменить вхождениями x_{2820}.

Если для конкретного селектора получены такого рода соотношения, мы можем хранить полученную информацию в специальной таблице и использовать её при обращении конкретных $\gamma \in \operatorname{Range} f$, рассматривая γ как набор значений соответствующих селекторов.

Для тестирования описанной техники проведён ряд вычислительных экспериментов. Во всех экспериментах задействовался кластер Иркутского суперкомпьютерного центра [16]. Рассмотрены две криптографические функции.

В первой серии экспериментов мы рассмотрели 10-раундовую версию легковесного шифра Simon [17], конкретно — 64-битный вариант Simon, сокращённый до 10 раундов (в оригинальной версии 32 раунда). Исследовалась задача поиска 64-битного ключа по трём известным блокам открытого текста и шифртекста (напомним, что в шифре Simon используются 32-битные блоки). Решая данную задачу, мы искали линеаризующие множества в том же стиле, как это сделано в работах [13, 18]. Перебирались селекторы ранга $k \leqslant 3$. Предварительный расчёт с вызовом SAT-оракула для конкретных гейтов и конкретных значений селекторов занимал несколько суток работы кластера [16]. Затем с использованием данной информации решалась задача оптимизации целевой функции, описанной в [18]. Как итог была построена атака на рассматриваемый шифр с оценкой трудоёмкости примерно в 150 раз меньше, чем атака, аналогичная [18]. Результаты приведены в следующей таблице:

Атака	Мощность линеаризующего множества	Оценка вероятности линеаризации	Оценка сложности атаки (число систем ЛУ)
1	62	1	$1{,}38 \cdot 10^{19}$
2	55	0,298	$3{,}63 \cdot 10^{17}$
3	54	0,557	$9{,}70 \cdot 10^{16}$

Здесь 1 — атака, использующая только технику из [18]; 2 — атака, использующая селекторные ограничения, но линеаризующее множество строится только на переменных ключа; 3 — атака, использующая селекторные ограничения, дополненная расширенным поиском линеаризующего множества (к переменным ключа добавляются некоторые переменные, функционально зависящие от ключа).

Во второй серии экспериментов мы рассмотрели задачу обращения полнораундовой функции сжатия алгоритма хеширования MD4 с использованием техники ослабляющих ограничений, описанной в [15, 19, 20]. Однако снижение трудоёмкости атаки за счёт использования информации от селекторов составило всего около 20 %. Отметим, что в этих экспериментах не использовалась информация от селекторов на этапе оптимизации функции, оценивающей трудоёмкость IBS-атаки. Возможно, именно поэтому достигнутые на данном этапе результаты не столь успешны, как с шифром Simon. Указанный недостаток мы планируем устранить в ближайшем будущем.

Авторы выражают глубокую благодарность Гладушу Андрею Игоревичу за помощь в проведении вычислительных экспериментов.

ЛИТЕРАТУРА

1. *Bard G.* Algebraic Cryptanalysis. Springer, 2009.
2. *Cook S. A.* The complexity of theorem-proving procedures // Proc. 3rd Ann. ACM Symp. Theory Comput. 1971. P. 151–158.
3. *Левин Л. А.* Универсальные задачи перебора // Проблемы передачи информации. 1973. Т. 9. № 3. С. 115–116.
4. *Clarke E., Kroening D., and Lerda F.* A tool for checking ANSI-C programs // LNCS. 2004. V. 2988. P. 168–176.
5. *Erkok L., Carlsson M., and Wick A.* Hardware/software co-verification of cryptographic algorithms using Cryptol // Proc. 9th Intern. Conf. FMCAD. IEEE, 2009. P. 188–191.
6. *Otpuschennikov I., Semenov A., Gribanova I., et al.* Encoding cryptographic functions to SAT using Transalg system // Frontiers Artificial Intell. Appl. 2016. V. 285. P. 1594–1595.
7. *Калгин К. В., Софронова Д. А.* Компактный транслятор алгоритмов в булевы формулы для применения в криптоанализе // Прикладная дискретная математика. Приложение. 2020. № 13. С. 135–136.

8. *Semenov A., Otpuschennikov I., Gribanova I., et al.* Translation of algorithmic descriptions of discrete functions to SAT with application to cryptanalysis problems // Log. Methods Comput. Sci. 2020. V. 16. Iss. 1. P. 29:1–29:42.
9. *Яблонский С. В.* Введение в дискретную математику. М.: Наука, 1986.
10. *Dowling W. F. and Gallier J. H.* Linear-time algorithms for testing the satisfiability of propositional horn formulae // J. Log. Program. 1984. No. 1(3). P. 267–284.
11. *Biere A.* The AIGER And-Inverter Graph (AIG) format version 20071012. Tech. Report 07/1. Institute for Formal Models and Verification, Johannes Kepler University. 2007.
12. *Kipnis A. and Shamir A.* Cryptanalysis of the HFE public key cryptosystem by relinearization // LNCS. 1999. V. 1666. P. 19–30.
13. *Семёнов А. А., Антонов К. В., Отпущенников И. В.* Поиск линеаризующих множеств в алгебраическом криптоанализе как задача псевдобулевой оптимизации // Прикладная дискретная математика. Приложение. 2019. № 12. С. 130–134.
14. *Антонов К.В., Семёнов А.А.* Применение SAT-оракулов для генерации дополнительных линейных ограничений в задачах криптоанализа некоторых легковесных шифров. Прикладная дискретная математика. Приложение. 2020. № 13. С. 114–119.
15. *Грибанова И. А., Семёнов А. А.* Об аргументации отсутствия свойств случайного оракула у некоторых криптографических хеш-функций // Прикладная дискретная математика. Приложение. 2019. № 12. С. 95–98.
16. ЦКП Иркутский суперкомпьютерный центр СО РАН. `http://hpc.icc.ru`.
17. *Beaulieu R., Shors D., Smith J., et al.* The Simon and Speck lightweight block ciphers // Proc. 52nd Ann. Design Automation Conf. New York, USA, 2015. P. 175:1–175:6.
18. *Антонов К. В., Семёнов А. А.* Применение метаэвристических алгоритмов псевдобулевой оптимизации к поиску линеаризующих множеств в криптоанализе криптографических генераторов // Материалы 6-й Междунар. школы-семинара «Синтаксис и семантика логических систем». Иркутск: ИГУ, 2019. С. 13–18.
19. *Gribanova I. and Semenov A.* Using automatic generation of relaxation constraints to improve the preimage attack on 39-step MD4 // Proc. 41st Intern. Convention Inform. Commun. Technol. Electr. Microelectr. (MIPRO). IEEE, 2018. P. 1174–1179.
20. *Gribanova I. and Semenov A.* Parallel guess-and-determine preimage attack with realistic complexity estimation for MD4-40 cryptographic hash function // Материалы конф. «Параллельные вычислительные технологии (ПаВТ) 2019» (Калининград, 2–4 апреля 2019). С. 8–18.

УДК 621.391.7 DOI 10.17223/2226308X/14/24

CHOOSING PARAMETERS FOR ONE IND-CCA2 SECURE McEliece MODIFICATION IN THE STANDARD MODEL

Y. V. Kosolapov, O. Y. Turchenko

The paper is devoted to choosing parameters for one IND-CCA2-secure McEliece modification in the standard model. In particular, the underlying code, plaintext length and one-time strong signature scheme are suggested. The choice of parameters for the scheme was based on efficiency, on the one hand, and security, on the other. Also, experiments for the suggested parameters are provided using the NIST statistical test suite.

Keywords: *post-quantum cryptography, McEliece-type cryptosystem,* IND-CCA2-*security, NIST statistical test suite.*