

4. *Бабьева А. А., Кязжин С. Н.* Аддитивно связанные ключи подписи: взломать нельзя использовать // XXVI Научно-практич. конф. «РусКрипто'2024». https://www.ruscrypto.ru/resource/archive/rc2024/files/05_babueva_kyazhin.pdf.
5. *Morita H., Schuldt J. C. N., Matsuda T., et al.* On the security of the Schnorr signature scheme and DSA against related-key attacks // LNCS. 2016. V. 9558. P. 20–35.
6. *Krawczyk H.* The “SIGn-and-MAC” approach to authenticated Diffie — Hellman and its use in the IKE protocols // LNCS. 2003. V. 2729. P. 400–425.
7. *Jeong I. R., Katz J., and Lee D. H.* One-Round Protocols for Two-Party Authenticated Key Exchange. 2008. https://www.cs.umd.edu/~jkatz/papers/1round_AKE.pdf.
8. *Canetti R. and Krawczyk H.* Analysis of key-exchange protocols and their use for building secure channels // LNCS. 2001. V. 2045. P. 453–474.

УДК 519.7

DOI 10.17223/2226308X/17/14

АЛГЕБРАИЧЕСКИЕ АТАКИ НА НЕКОТОРЫЕ НИЗКОРЕСУРСНЫЕ ШИФРЫ НА ОСНОВЕ ФУНКЦИЙ С МАЛЫМ ЧИСЛОМ ВЫХОДНЫХ БИТ¹

К. В. Антонов, А. А. Семёнов, И. В. Отпущенников, А. Л. Павленко

Вводится новый класс атак, применимых к низкоресурсным функциям симметричной криптографии. Основная идея атак основана на использовании специальных функций, получаемых из оригинальных за счёт рассмотрения малого числа выходных бит. Задачи обращения специальных функций существенно более просты для используемых в алгебраическом криптоанализе комбинаторных алгоритмов (SAT-решателей), чем задачи обращения функций, из которых они строятся. Однако для специальной функции, ввиду того, что её выход существенно короче входа, задача обращения имеет не единственное решение. Для достижения единственности по ключу описывается процедура клонирования функции, в рамках которой специальные функции строятся для одного ключа и нескольких различных фрагментов открытых данных. Операция клонирования специальных функций выполняется на And-Inverter-графах (AIG), представляющих данные функции. Во многих случаях размер AIG может быть существенно уменьшен за счёт применения алгоритмов AIG-минимизации. Результатом совместного использования перечисленных техник являются алгебраические атаки, которые для ряда поточных шифров могут быть существенно более эффективными в сравнении с аналогичными атаками на функции, представляющие рассматриваемые шифры стандартным образом.

Ключевые слова: алгебраический криптоанализ, низкоресурсная криптография, SAT-решатели, булевы схемы.

1. Введение и постановка задачи

Атаки, о которых идёт речь, основаны на идее, опубликованной в [1, 2]. Работа [2] довольно широко цитируется, а описанный в ней сценарий известен как «кубическая атака» (cube attack), работоспособность данной атаки демонстрируется на вариантах известного шифра Trivium [3], ослабленных по числу шагов инициализационной фазы. Основным результатом [2] заявлен подход, в рамках которого зависимость меж-

¹Исследование выполнено в рамках госзадания Минобрнауки России по проекту «Теоретические основы, методы и высокопроизводительные алгоритмы непрерывной и дискретной оптимизации для поддержки междисциплинарных научных исследований», номер гос. регистрации 121041300065-9.

ду переменными, кодирующими ключ шифра, ищется в форме линейных уравнений над $\text{GF}(2)$, получаемых из так называемого мастер-полинома (master-polynomial). Сам мастер-полином предполагается неизвестным (black-box), однако в [2] предлагается специальная техника, которая позволяет выводить линейные соотношения на биты ключа, исходя из предположений о его степени: используется понятие «случайного полинома степени d » (d -random polynomial). Если d не слишком велика, то линейные соотношения на биты ключа можно получать, варьируя значения некоторых переменных в соответствующем гиперкубе.

Для целей настоящей работы интересна не сама техника «tweakable black-box polynomial» [2], а функции, к которым она применяется. Мастер-полином, рассматриваемый в [2], представляет некоторую булеву функцию вида

$$f : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}, \quad (1)$$

где $\{0, 1\}^n$ — множество, из которого выбран секретный ключ, а $\{0, 1\}^m$ — множество, из которого выбирается известная информация: инициализирующая последовательность (IV) в случае поточного шифра или блок открытого текста в случае блочного. Заметим, что выходом функции (1) является 1 бит. По-видимому, впервые задача обращения функций вида (1) в контексте криптоанализа (кстати, также шифра Trivium) рассматривалась в [1]. Особо подчеркнём характерную особенность, отличающую функции вида (1) от функций, обычно рассматриваемых в задачах алгебраического криптоанализа и имеющих вид

$$g : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}^l, \quad (2)$$

где $\{0, 1\}^n$ — множество секретных ключей; $\{0, 1\}^m$ — множество открытых текстов или инициализирующих векторов; $\{0, 1\}^l$ — множество криптограмм. Важный факт состоит в том, что в (2) для достижения единственности по ключу предполагается, что $l \geq n$. Например, для шифра Trivium [3] «стандартная» атака направлена на обращение функции $g : \{0, 1\}^{80} \times \{0, 1\}^{80} \rightarrow \{0, 1\}^l$, $l \geq 80$: ставится задача найти 80 бит секретного ключа на основе известных l бит ключевого потока, сгенерированных алгоритмом Trivium по этому ключу и одному известному IV.

Далее задачи алгебраического криптоанализа решаются за счёт их сведения к проблеме булевой выполнимости (SAT) и использования современных SAT-решателей. В данном контексте задача обращения функций вида (2) для стойких шифров крайне сложна, тогда как обращение функций вида (1) не является сложным. Однако для функций вида (1) в общем случае, конечно, нет единственности по ключу. Далее мы несколько обобщаем исследуемый класс функций и рассматриваем функции вида

$$f : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}^k, \quad (3)$$

где $k \ll n$. Очевидно, что в таких условиях при обращении (3) снова нет единственности по ключу, однако можно исследовать проблему обращения функций, полученных «клонированием» некоторого числа функций вида (3) для различных известных входных данных. Иными словами, будем рассматривать задачу поиска $z \in \{0, 1\}^n$ на основе $x^i \in \{0, 1\}^m$, $i \in \{1, \dots, s\}$, таких, что для каждой пары вида (z, x^i) известно значение функции $f(z, x^i)$ вида (3). Таким образом, речь идет об обращении функций вида

$$F : \{0, 1\}^n \times \{0, 1\}^{sm} \rightarrow \{0, 1\}^{sk}, \quad (4)$$

где $k \ll n$, $sk \geq n$.

Например, для шифра Trivium элементарная функция (3) строится для пары (z, IV) , где z — неизвестный секретный ключ, $z \in \{0, 1\}^{80}$; IV — известная инициализирующая последовательность, $IV \in \{0, 1\}^{80}$; выход функции — некоторые фиксированные k бит ключевого потока, генерируемого шифром Trivium по входу (z, IV) . По крайней мере, для поточных шифров в роли значения (3) имеет смысл брать первые k бит соответствующего ключевого потока. Далее для s различных IV рассматриваются аналогичные функции и формируется функция вида (4).

Базовое предположение состоит в том, что для s , незначительно больших n , и для естественным образом подобранных различных IV при обращении (4) достигается единственность по ключу. Далее показано, что для ряда известных криптографических функций задачи обращения функций вида (4) существенно проще обращения функций вида (2).

Для задач обращения функций вида (4) построены атаки из класса «угадай и определяй» (guess-and-determine), относящиеся к алгебраическому криптоанализу [4]. В таких атаках рассматриваемая задача обращения сводится к SAT, а к формулам в КНФ, ослабленным подстановками угаданных бит, применяются SAT-решатели. Конкретно, использован IBS-метод [5], основное преимущество которого состоит в том, что получаемые им оценки имеют строгие гарантии точности.

2. Предварительные сведения

Итак, главным объектом изучения являются симметричные шифры, в основном, относящиеся к низкоресурсной (lightweight) криптографии, и алгебраические атаки на них, использующие SAT-решатели. Для сведения задачи обращения функций вида (2) и (4) к SAT используются методы, базовые принципы которых изложены, например, в [6, 7]. Кратко эти принципы можно описать следующим образом. Рассматривается функция вида $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$, определённая всюду на $\{0, 1\}^n$ и заданная некоторым быстрым алгоритмом A_f , который известен. По известному $\gamma \in \text{Range } f \subseteq \{0, 1\}^m$ требуется найти такой $\alpha \in \{0, 1\}^n$, что $f(\alpha) = \gamma$. По тексту A_f эффективно строится булева схема S_f (схема из функциональных элементов) над произвольным полным базисом, задающая f . По схеме S_f за линейное время от числа узлов (часто называемых гейтами) в этой схеме строится специальная КНФ C_f , называемая шаблонной (template CNF [7]). Для этой цели используются преобразования Цейтина [8].

Шаблонная КНФ обладает рядом важных свойств, одно из которых заключается в том, что интерпретация схемы S_f на произвольном входе $\alpha \in \{0, 1\}^n$ может быть промоделирована последовательностью применений правила единичного дизъюнкта (Unit Propagation rule, UP) к КНФ C_f и множеству литералов $x_1^{\alpha_1}, \dots, x_n^{\alpha_n}$, где $\alpha = (\alpha_1, \dots, \alpha_n)$; $X^{\text{in}} = \{x_1, \dots, x_n\}$ — множество переменных, приписанных входам схемы S_f . Обозначение x^σ понимается в смысле [9]: $x^0 = \neg x$, $x^1 = x$. Последнее свойство означает, что вычисление по схеме S_f на входе $\alpha = (\alpha_1, \dots, \alpha_n)$ можно представить как последовательность применений UP к формуле $x_1^{\alpha_1} \wedge \dots \wedge x_n^{\alpha_n} \wedge C_f$, итогом чего будет некоторый набор литералов $y_1^{\gamma_1}, \dots, y_m^{\gamma_m}$, где $Y = \{y_1, \dots, y_m\}$ — множество переменных, приписанных выходам схемы S_f ; $\gamma = (\gamma_1, \dots, \gamma_m) : f(\alpha) = \gamma$.

Обозначим через $C[\beta/B]$ КНФ, полученную из исходной КНФ C в результате подстановки в C набора значений β переменных из множества B , $B \subseteq X$ (X — множество, образованное всеми переменными в КНФ C). Подстановка понимается в стандартном смысле [10], то есть как результат замены вхождений переменной соответствующей константой с последующим выполнением всех возможных элементарных преобразований. Соответственно результат применения правила UP к формуле $x_1^{\alpha_1} \wedge \dots \wedge x_n^{\alpha_n} \wedge C_f$

можно интерпретировать как $C_f[\alpha/X^{\text{in}}]$. Результатом $C_f[\alpha/X^{\text{in}}]$ является набор значений всех переменных из X ; скажем, что такой набор индуцирован входом α .

Изложим кратко суть IBS-атак [5]. Для произвольного входа $\alpha \in \{0, 1\}^n$ и произвольного $B \subseteq X$ обозначим через β_α набор значений переменных из B , индуцированный входом α . Пусть $\gamma_\alpha = f(\alpha)$. Рассмотрим КНФ $C_f[\beta_\alpha/B, \gamma_\alpha/Y]$ (то есть результат подстановки в C_f наборов β_α и γ_α). К $C_f[\beta_\alpha/B, \gamma_\alpha/Y]$ применяется алгоритм A^t , время работы которого ограничено сверху величиной t (обычно t — некоторая константа); после достижения этого порога A^t прерывает работу. Предполагается, что для любой КНФ A^t выдаёт ответ из множества $\{\text{SAT}, \text{UNSAT}, \text{INDET}\}$. Ответ INDET означает, что A^t не смог определить выполнимость/невыполнимость формулы за время $\leq t$. Отметим, что алгоритм A^t — это фактически некоторый полиномиальный вспомогательный решатель из определения сильной лазейки (Strong Backdoor Set, SBS) [11]. Для произвольной КНФ C над переменными X , произвольного $B \subseteq X$ и алгоритма A^t обозначим через $C[\beta/B] \in \Sigma(A^t)$ ситуацию, когда результат применения A^t к $C[\beta/B]$ лежит в множестве $\{\text{SAT}, \text{UNSAT}\}$; в случае ответа INDET используем обозначение $C[\beta/B] \notin \Sigma(A^t)$.

Зададим на $\{0, 1\}^n$ равномерное распределение и рассмотрим величину

$$\rho_B = |\{\alpha : C_f[\beta_\alpha/B, \gamma_\alpha/Y] \in \Sigma(A^t)\}|/2^n. \quad (5)$$

Величина (5) — это вероятность того, что решатель A^t обратит $\gamma_\alpha \in \text{Range } f$ при условии известного β_α , играющего роль правильной подсказки. Множество B с $\rho_B > 0$ называется Inverse Backdoor Set (IBS). Если B — некоторый IBS, то на его основе можно построить атаку [5], которая обращает хотя бы один выход из множества $\gamma_1, \dots, \gamma_k$, $\gamma_i \in \text{Range } f$, $i \in \{1, \dots, k\}$, с вероятностью $> 0,95$ за время $\approx 2^{|B|} \cdot 3t/\rho_B$.

Шаблонные КНФ для функций вида (4), построенные стандартным образом [7], для относительно большого числа s (скажем, для $s \geq 100$) получаются очень большими. Для нивелирования этого эффекта использованы алгоритмы минимизации булевых схем, реализованные в библиотеке ABC [12]. С этой целью схема S_F , задающая функцию (4), представляется в виде And-Inverter-графа (And-Inverter Graph, AIG) [13], после чего части входных полюсов S_F приписываются значения, соответствующие известным инициализирующим векторам либо блокам открытого текста (для поточных и блочных шифров соответственно). К такой схеме с частично означенным множеством входов (соответствующих открытым входным данным) применяется программа ABC. В итоге строится схема, задающая функцию следующего вида:

$$\tilde{F} : \{0, 1\}^n \rightarrow \{0, 1\}^{sk}. \quad (6)$$

В некоторых случаях схема $S_{\tilde{F}}$ по размеру (по числу вершин) в десятки раз меньше схемы S_F . Для функций вида (6) строятся IBS-атаки в соответствии со сценарием [5].

3. Вычислительные эксперименты

В экспериментах рассматривались ослабленные по числу шагов инициализации варианты известных поточных шифров Trivium [3] и Grain [14] (Grain v1.0), а также блочный низкоресурсный шифр Simon [15], ослабленный по числу раундов.

Для шифра Trivium выбирались значения числа шагов инициализации $M \in \{288, 320, 352, 384, 480, 576\}$. Задача обращения функции вида (2) является сложной уже для $M = 288$. Так, например, IBS-атака на данную функцию [16] даёт IBS B с $|B| = 63$ и оценкой трудоёмкости в секундах $\approx 2^{68}$ (на одном ядре процессора Intel E5-2695).

Для Trivium мы построили несколько вариантов функций вида (3), (4). Для каждой функции выбирались случайно и независимо $s = 100$ векторов IV из $\{0, 1\}^{80}$. Для $M = 288$ сначала рассматривалась функция (3) с $k = 1$, однако в этом случае функция (4), хоть и обрабатывается быстро обычным последовательным SAT-решателем Kissat [17], не обладает свойством единственности ключа. Единственность ключа для $M = 288$ была достигнута при параметрах $k = 8, s = 100$. Что оказалось весьма неожиданным, в таком варианте функция (4) не является сложной: секретный ключ находится за несколько минут решателем Kissat. Таким образом, даже на этом примере видно, насколько обращение функций вида (4) проще, чем функций вида (2). Обратив при помощи Kissat функции (4) для Trivium- M удалось для $M \in \{288, 320, 352\}$: для Trivium-352 Kissat обратил (4) с параметрами $k = 8, s = 100$ за 2 ч 40 мин.

Задача обращения Trivium-384 уже оказалась сложной для последовательного Kissat (не решилась за сутки); для M , начиная с 384, для функций вида (4) строились IBS-атаки. Результаты приведены в табл. 1, тут же дано сравнение построенных атак с атаками для Trivium- M на основе функций вида (2).

Т а б л и ц а 1

IBS-атаки для Trivium- M

M	Функции (2), параметры: $k \geq 80, s = 1$			Функции (4), параметры: $k = 10, s = 100$		
	Размер КНФ, Мбайт	$ B $	Оценка трудности атаки, с (одно ядро Intel E5-2695)	Размер КНФ, Мбайт	$ B $	Оценка трудности атаки, с (одно ядро Intel E5-2695)
288	0,99	49	$2^{59,8}$	0,52	—	≤ 300
384	1,20	55	$2^{66,1}$	2,71	42	$2^{57,5}$
480	1,41	57	$2^{68,6}$	10,0	56	$2^{71,5}$
576	1,62	58	$2^{69,6}$	24,3	61	$2^{75,5}$

Комментарии к табл. 1. В столбце «Размер КНФ» в случае функции (2) содержится размер формулы, сгенерированной системой Transalg, в случае функции (4) — размер формулы, полученной по AIG, оптимизированному при помощи программы ABC. Можно заметить, что для относительно небольшого числа шагов инициализации задача обращения функций вида (4) проще, чем для функций (2), при том что для функций вида (4) размер кодировок существенно растёт с ростом числа шагов. Для ситуаций, когда кодировка функции (4) по размеру превосходит кодировку функции (2) в 10 раз и более, на работу решателю A^t на формулах для (4) давалось в 10 раз больше времени, чем для (2) (конкретно, 300 и 30 с соответственно).

В табл. 2 приведены результаты аналогичных экспериментов для поточного шифра Grain v1.0, который, как и Trivium, является одним из победителей конкурса eSTREAM. Данный шифр также имеет стадию инициализации, которая в соответствии со спецификацией алгоритма состоит из 160 шагов. Версии Grain v1.0 с числом шагов инициализации, равным M , обозначаются через Grain-v1- M ; $M \in \{90, 100, 110, 120\}$.

Комментарии к табл. 2. В случае Grain v1.0 выигрыш по времени при обращении функций (4) в сравнении с (2) существенно больше, чем для Trivium- M . Следует отметить также, что на кодировках функций вида (4) для Grain-v1- M более явным выглядит эффект от AIG-оптимизации, чем для Trivium- M .

Наконец, в последней серии экспериментов рассмотрены атаки аналогичного типа на ослабленный по числу раундов блочный шифр Simon 32/64, конкретно — на первые его 10 раундов. К сожалению, атаки на функции вида (4), задаваемые данным шифром, не продемонстрировали большей эффективности в сравнении с IBS-атаками,

IBS-атаки для Grain-v1-M

M	Функции (2), параметры: $k \geq 80, s = 1$			Функции (4), параметры: $k = 10, s = 100$		
	Размер КНФ, Мбайт	B	Оценка трудности атаки, с (одно ядро Intel E5-2695)	Размер КНФ, Мбайт	B	Оценка трудности атаки, с (одно ядро Intel E5-2695)
90	1,56	52	$2^{67,5}$	4,23	29	$2^{42,7}$
100	1,68	49	$2^{65,5}$	7,34	31	$2^{44,3}$
110	1,81	52	$2^{66,9}$	11,3	41	$2^{56,5}$
120	1,94	53	$2^{69,5}$	15,0	44	$2^{60,5}$

построенными для функций вида (2). Лучшая атака для функции вида (2) в случае Simon 32/64 даёт IBS из 37 переменных и оценку трудоёмкости $2^{47,6}$ секунд. В то же время для функции вида (4) с параметрами $k = 1$ и $s = 100$ имеем IBS B с $|B| = 45$, трудоёмкость $2^{59,5}$. Для работы с функциями (4) решателю A^t выделялось в 10 раз больше времени, чем для работы с функциями (2) (300 с против 30 с), поскольку кодировки (4) для Simon 32/64 превосходят кодировки (2) по объёму более чем в 10 раз.

Все вычисления проводились на кластере «Академик В. М. Матросов» Иркутского суперкомпьютерного центра [18].

З а к л ю ч е н и е

Описан новый класс алгебраических атак, базирующийся на задаче обращения специальных функций, использовавшихся ранее в кубических атаках. Такая функция имеет выход, длина которого существенно меньше длины секретного ключа, и соответственно прообраз такого выхода в общем случае не единственный. Предлагается специальная техника, которую можно условно назвать «клонированием» функций данного типа: рассматриваются несколько функций, построенных для общего секретного ключа и различных открытых входных данных (фрагментов ключевого потока или криптограмм в случае поточных и блочных шифров соответственно). Пожалуй, основной новизной предлагаемого подхода является представление таких клонированных функций в виде And-Inverter-графов с последующим применением к ним инструментов AIG-оптимизации. В вычислительных экспериментах на примерах ослабленных по числу шагов инициализации низкоресурсных поточных шифров Trivium и Grain v1.0 продемонстрировано, что алгебраические атаки на специальные функции могут быть существенно более эффективными, чем атаки на стандартные функции, задаваемые этими криптографическими алгоритмами. В ближайших планах расширить класс описанных атак за счёт сценариев, использующих выбранные открытые входные данные (chosen plaintext).

Л И Т Е Р А Т У Р А

1. Fischer S., Khazaei S., and Meier W. Chosen IV statistical analysis for key recovery attacks on stream ciphers // LNCS. 2008. V. 5023. P. 236–245.
2. Dinur I. and Shamir A. Cube attacks on tweakable black box polynomials // LNCS. 2009. V. 5479. P. 278–299.
3. De Canniere C. Trivium: a stream cipher construction inspired by block cipher design principles // LNCS. 2006. V. 4176. P. 171–186.
4. Bard G. Algebraic Cryptanalysis. Springer, 2009.
5. Semenov A., Zaikin O., Otpuschennikov I., et al. On cryptographic attacks using backdoors for SAT // Proc. AAAI. Palo Alto, USA, 2018. P. 6641–6648.

6. *Otpuschennikov I., Semenov A., Gribanova I., et al.* Encoding cryptographic functions to SAT using Transalg system // Proc. ECAI. Hague, Netherlands, 2016. P. 1594–1595.
7. *Semenov A., Otpuschennikov I., Gribanova I., et al.* Translation of algorithmic descriptions of discrete functions to SAT with applications to cryptanalysis problems // Logical Methods Computer Sci. 2020. V. 16. Iss. 1. P. 29:1–29:42.
8. *Цейтин Г. С.* О сложности вывода в исчислении высказываний // Записки научных семинаров ЛОМИ. 1968. Т. 8. С. 234–259.
9. *Яблонский С. В.* Введение в дискретную математику. М.: Наука, 1986.
10. *Чень Ч., Лу Р.* Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983.
11. *Williams R., Gomes C., and Selman B.* Backdoors to typical case complexity // Proc. IJCAI'03. Acapulco, Mexico, 2003. P. 1173–1178.
12. *Bryton R. and Mishchenko A.* ABC: An academic industrial-strength verification tool // LNCS. 2010. V. 6174. P. 24–40.
13. *Kuehlmann A. Paruthi V., Krohm F., and Ganai M.* Robust Boolean reasoning for equivalence checking and functional property verification // IEEE Trans. Computer-Aided Des. Integr. Circuits Systems. 2002. V. 21. No. 12. P. 1377–1394.
14. *Hell M., Johansson T., and Meier W.* Grain: a stream cipher for constrained environments // Intern. J. Wireless Mobile Computing. 2007. V. 2. Iss. 1. P. 86–93.
15. *Beaulieu R., Shors D., Smith J., et al.* The Simon and Speck lightweight block ciphers // 52nd Ann. Design Automation Conf. San Francisco, USA, 2015. P. 175:1–175:6.
16. *Semenov A., Antonov K., Otpuschennikov I., and Pavlenko A.* Using linearizing sets to solve multivariate quadratic equations in algebraic cryptanalysis // IEEE Access. 2023. V. 11. P. 120319–120333.
17. *Biere A., Fazekas K., Fleury M., and Heisinger M.* CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT Competition 2020 // Proc. SAT Competition. Helsinki, 2020. P. 50–53.
18. Иркутский суперкомпьютерный центр. <https://hpc.icc.ru/hardware/index.html>.

УДК 519.7

DOI 10.17223/2226308X/17/15

О СТОЙКОСТИ НЕКОТОРЫХ АЛГОРИТМОВ НАД ГРУППОЙ ТОЧЕК ЭЛЛИПТИЧЕСКИХ КРИВЫХ

А. О. Бахарев, К. Д. Царегородцев

Приводятся результаты анализа схемы VKO и комбинированной схемы VKO+подпись в моделях обобщённой группы и биективного случайного оракула. Получена верхняя оценка сложности задачи различения выхода схемы VKO от случайной равновероятной строки (в эвристике обобщенной группы), а также показано, что возможность получения подписи сообщений по алгоритму genGOST не даёт никакой дополнительной информации противнику в этой задаче (в эвристике биективного случайного оракула).

Ключевые слова: доказуемая стойкость, VKO, электронная подпись.

Введение

В ходе изучения различных криптографических протоколов возникают вопросы, связанные со стойкостью алгоритмов, в основе которых лежат вычисления в различных группах (например, в группе точек эллиптической кривой). Существует несколько возможных подходов к анализу подобных задач.