СВЕДЕНИЕ ЗАДАЧ ФАКТОРИЗАЦИИ, ДИСКРЕТНОГО ЛОГАРИФМИРОВАНИЯ И ЛОГАРИФМИРОВАНИЯ НА ЭЛЛИПТИЧЕСКОЙ КРИВОЙ К РЕШЕНИЮ АССОЦИИРОВАННЫХ ЗАДАЧ «ВЫПОЛНИМОСТЬ»

Дулькейт Владимир Игоревич (аспирант, e-mail: vidulkeyt@mail.ru),
Омский государственный университет им. Ф.М. Достоевского

Аннотация

В работе предлагаются алгоритмы консервативного сведения задач факторизации, дискретного логарифмирования и логарифмирования на эллиптической кривой к задаче «ВЫПОЛНИМОСТЬ», проводится анализ работы современных алгоритмов решения задачи «ВЫПОЛНИМОСТЬ» (SAT – решателей) на полученных КНФ. Исследуется стойкость рассматриваемых задач к восстановлению полного ключа по его известным фрагментам.

Ключевые слова: КНФ, факторизация, дискретное логарифмирование, логарифмирование на эллиптической кривой, «ВЫПОЛНИМОСТЬ».

Введение

Одним из ключевых направлений криптографического анализа является проверка криптографической стойкости алгоритмов асимметричного шифрования, поскольку на их базе работает подавляющее большинство криптографических протоколов обмена ключами, цифровой подписи и т.д. В настоящее время для реализации данной проверки применяются в основном методы решета в поле чисел общего вида и различные модификации ρ- и λ-методов Полларда, базирующиеся на псевдослучайном блуждании по группе. Последние достижения в этой области свидетельствуют о стойкости известных алгоритмов, поскольку для решения задач «рабочих» размерностей (т.е. регламентированных требованиями безопасности) требуется на несколько месяцев задействовать вычислительный кластер категории самых верхних позиций списка «Топ-500».

Однако относительно недавно возникло и получило развитие совершено новое, альтернативное алгебраическому подходу направление криптоанализа – логический криптоанализ. Суть подхода заключается в рассмотрении криптографического алгоритма как программы для машины Тьюринга. Подстановка открытого и шифрованного текстов в эту программу естественным образом приводит к задаче «ВЫПОЛНИМОСТЬ» для конъюнктивной нормальной формы (КНФ), часть выполняющего набора которой является ключом шифра. Идея такого подхода была впервые предложена Куком С. в 1997 году при поиске сложных задач для тестирования решателей КНФ [1].

Последующие исследования по логическому криптоанализу фокусировались на блочных и потоковых шифрах [2,3], генераторах двоичных последовательностей [4], а также некоторых аспектах криптоанализа RSA (криптостойкость основана на сложности задачи факторизации) [5,6]. При этом за границами исследований остались такие важные задачи, как дискретное логарифмирование и логарифмирование в группе точек эллиптической кривой, на основе которых строятся современные системы шифрования, протоколы обмена ключами и цифровой подписи (DSA, ECDSA и другие). В вопросе применения подходов логического криптоанализа для задачи факторизации недостаточно полно освещены такие аспекты, как использование параллельных алгоритмов для поиска выполняющего набора КНФ, кодирующей исходную задачу, и адаптация алгоритмов кодирования под требования современных алгоритмов поиска выполняющего набора КНФ.

Таким образом, целью данной работы является построение и исследование свойств алгоритмов консервативного сведения задач факторизации, дискретного логарифмирования и логарифмирования на группе точек эллиптической кривой к задаче «ВЫПОЛНИМОСТЬ»; анализ работы современных SAT - решателей на полученных КНФ.

Сведение к задаче «ВЫПОЛНИМОСТЬ» позволит не только применять для решения изначально алгебраических задач алгоритмы решения задачи «ВЫПОЛНИМОСТЬ», но и получать качественно новые результаты, недоступные для алгебраических методов. Так, например, выделять наиболее вероятные биты ключа, распознавать определенные последовательности бит [7,8] и полностью восстанавливать ключ по некоторым известным его фрагментам.

1. Алгоритмы сведения к задаче «ВЫПОЛНИМОСТЬ»

Задача факторизации

Для перехода от задачи факторизации (разложения целого числа на два различных целых сомножителя) к задаче «ВЫПОЛНИМОСТЬ» (поиска выполняющего набора логической формулы) основной операцией, которую необходимо закодировать в виде КНФ, является операция умножения двух чисел, в основе которой лежит операция вычисления суммы по модулю 2 и переноса в старший разряд. Для построения соответствующих фрагментов КНФ предлагается использовать правило де-Моргана в сочетании со следующими выражениями, являющимися прямым следствием свойств совершенных КНФ:

$$\bigoplus_{i=1}^{N} x_i = \bigwedge_{\{\sigma_i\}\in M_N} \left(x_1^{\sigma_1} \vee x_2^{\sigma_2} \vee \cdots \vee x_N^{\sigma_N} \right), \qquad (1)$$

где в левой части сумма по модулю 2, M_N – множество двоичных векторов длины N, содержащих чётное число нулей.

$$\bigvee_{i=1}^{N} x_i^{\delta_i} \vee \bigwedge_{j=1}^{L} y_j^{\sigma_j} = \bigwedge_{\substack{\{\pi_k\}\in \\ 2^L/\{0,\ldots,0\}}} \left(\bigvee_{i=1}^{N} x_i^{\delta_i} \vee \bigvee_{j=1}^{L} \left(y_j^{\sigma_j} \right)^{\pi_j} \right). \qquad (2)$$

При этом операция вычисления переноса представляется как сумма по модулю 2 в следующем виде:

$$c = \neg \left(s \oplus x \wedge y \wedge z \oplus \bar{x} \wedge \bar{y} \wedge \bar{z} \right), \qquad (3)$$

где s – литерал, равный сумме $x \oplus y \oplus z$, а c – бит переноса от этой суммы.

Построено три различных алгоритма кодирования операции умножения чисел в виде КНФ:

1. алгоритм на основе умножения «столбиком»;
2. алгоритм, использующий датчик псевдослучайных чисел для генерации нескольких КНФ, представляющих одну задачу факторизации;
3. алгоритм генерации 3-КНФ.

Первый алгоритм фактически является записью обычного алгоритма умножения «столбиком», в котором на каждом шаге вместо вычисления промежуточной суммы или переноса происходит генерация соответствующего фрагмента КНФ.

Второй алгоритм генерирует несколько КНФ, представляющих одну задачу факторизации, что позволяет использовать параллельные схемы SAT - решателей [9]. В таких схемах параллельно работает несколько независимых решателей, каждый из которых ищет выполняющий набор своей индивидуальной КНФ. По окончании каждой итерации происходит циклический обмен решениями между решателями и начинается следующая итерация, в которой полученные решения являются начальными приближениями.

Генерация КНФ происходит в два этапа:

1. попарное сложение промежуточных векторов, составленных из произведений вида

$$y_i \sum_{j=1}^{N} 2^{j-1} x_j :$$

$$\sum_{j=1}^{N+1} 2^{j-1} u_{ij} = y_i \sum_{j=1}^{N} 2^{j-1} x_j + 2 y_{i+1} \sum_{j=1}^{N} 2^{j-1} x_j, \qquad (4)$$

2. сложение двух произвольно выбранных векторов $\overline{u_i}$, $\overline{u_k}$.

Представление операции умножения в виде 3-КНФ также базируется на алгоритме умножения «столбиком», при этом для кодирования сложения в одном разряде используется следующая система уравнений, каждое уравнение которой представляется в виде фрагмента 3-КНФ:

$$\begin{cases} s_1 = x \oplus c, \\ c_1 = x \wedge c, \\ s = s_1 \oplus y \\ c_2 = s_1 \wedge y \\ c' = c_1 \oplus c_2 \end{cases} \qquad (4)$$

Отсутствие дополнительных, искусственных (т.е. не имеющих смысла, с точки зрения исходной задачи) литералов в построенных КНФ является достоинством представленных схем кодирования.

Консервативность сведения задачи факторизации к задаче «ВЫПОЛНИМОСТЬ» обеспечивается путем включения в итоговую КНФ фрагментов, фиксирующих порядок сомножителей ($q \geq p$) и устраняющих тривиальные решения ($1 \times n$).

Все рассмотренные алгоритмы обладают следующими свойствами:

1. трудоемкость есть $O\left(N^2\right)$;
2. количество литералов в КНФ есть $O\left(N^2\right)$;
3. количество дизъюнктов в КНФ есть $O\left(N^2\right)$.

Задача дискретного логарифмирования

Задача дискретного логарифмирования формулируется следующим образом:

$$A^X \equiv B \bmod P, \qquad (5)$$

где P – большое простое число, A такое, что $(A,P)=1$. Требуется найти число X, удовлетворяющее сравнению (5).

Идея процедуры сведения задачи дискретного логарифмирования основана на следующем равенстве:

$$A^X \left(\bmod P\right) = \left(\ldots\left(A_1^{x_1} \times A_2^{x_2}\right) \times A_3^{x_3} \ldots\right) \times A_N^{x_N}, \qquad (6)$$

где $A_i = A^{2^{i-1}} \left(\bmod P\right)$, а операция $\times$ означает умножение в поле *GF(P)*:

$$UV \equiv R\left(\bmod P\right). \qquad (7)$$

При кодировании выражения (7) в виде КНФ значения A_i вычисляются численно и соответствующие биты подставляются в КНФ. Таким образом, требуется закодировать N-1 (N – разрядность числа P) операций умножения в поле *GF(P)*.

Для представления в виде КНФ левой части выражения (6) требуется реализовать кодирование в КНФ следующих операций:

1. возведения числа в степень, показатель которой может принимать только значения 0 или 1 (возведение в «однобитовую» степень);
2. умножения в поле *GF(P)* (7).

Кодирование в виде КНФ первой операции осуществляется следующим алгоритмом (на входе число A, содержащее N двоичных разрядов, и литерал b, отвечающий показателю экспоненты, вектор c_i (i=1,…,N) литералов, отвечающих битам результата):

1. Если младший бит A равен 0, то добавить в КНФ два дизъюнкта $\left(\bar{c}_1 \vee b\right) \wedge \left(c_1 \vee \bar{b}\right)$, иначе, добавить в КНФ дизъюнкт $\left(c_1\right)$.
2. Для i=2,…,N, если i-ый бит A (нумерация бит с единицы, начиная с младшего) равен 1, то положить $c_i=b$, иначе, добавить в КНФ дизъюнкт $\left(\bar{c}_i\right)$.

Вторая операция, которую необходимо закодировать в виде КНФ – умножение в поле *GF(P)*, представима в виде следующей системы:

$$\begin{cases} UV = QP + R, \\ P \geq R, \end{cases} \quad (8)$$

где U и V – сомножители, P – модуль поля $GF(P)$, R – результат умножения, Q – целая часть от деления UV/P. Без ограничения общности используется нестрогое неравенство, т.к., с одной стороны его проще закодировать в виде КНФ, с другой – в рассматриваемой задаче $UV \neq 0 \pmod P$.

Перепишем систему (8) так, чтобы в каждом уравнении содержалось по одной бинарной операции:

$$\begin{cases} UV = T, \\ QP = Y, \\ T = Y + R, \\ P \geq R, \end{cases} \quad (9)$$

где T и Y – дополнительные переменные связи между уравнениями.

Алгоритм кодирования в КНФ операции умножения в поле $GF(P)$ строится на основе системы (9) с использованием генераторов КНФ для операций умножения, сложения и отношения «больше или равно».

Алгоритм сведения задачи дискретного логарифмирования к задаче «ВЫПОЛНИМОСТЬ» принимает на вход числа A, B, P, имеющие N разрядов в двоичной записи и связанные соотношением (5), а также вектор литералов $\{x_i\}$ i=1,…,N, отвечающих битам искомой экспоненты. Основные шаги алгоритма:

1. Вычислить $A_i = A^{2^{i-1}} \pmod P$, при i=1,…,N.
2. При i=1,…,N-1 и $S_1 = A_1^{x_1}$ представить в виде КНФ следующее выражение:

$$S_{i+1} = S_i \times A_i^{x_i} \pmod P. \quad (10)$$

3. В полученную КНФ подставить биты P в качестве значений литералов p_i и биты B в качестве значений S_N.

Представленный алгоритм является консервативным, полиномиальным сведением задачи дискретного логарифмирования в простом поле по модулю, содержащему N двоичных разрядов, к задаче «ВЫПОЛНИМОСТЬ» и обладает следующими свойствами:

1. время работы алгоритма есть $O(N^3)$;
2. количество литералов в КНФ есть $O(N^3)$;
3. количество дизъюнктов в КНФ есть $O(N^3)$.

Задача логарифмирования в группе точек эллиптической кривой

Задача логарифмирования на эллиптической кривой формулируется следующим образом: дано две точки R, Q некоторой эллиптической кривой $E(A, B, P)$, необходимо найти число k такое, что при умножении на него точки Q результатом получим точку R:

$$R = kQ, \quad R, Q \in E(A, B, P). \quad (11)$$

При этом в теории эллиптических кривых имеются следующие правила вычисления суммы двух точек [10,11]:

1. $(x, y)+O=(x, y)$;
2. $(x, y)+(x, -y)=O$;
3. $(x_1, y_1)+(x_2, y_2)=(x_3, y_3)$,

где

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \pmod P \\ y_3 = \lambda(x_1 - x_3) - y_1 \pmod P \end{cases}. \quad (12)$$

Параметр λ вычисляется следующим образом:

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} \pmod P, \text{ если } (x_1, y_1) \neq (x_2, y_2), \\ \dfrac{y_3 = 3x_1^2 - A}{2y_1} \pmod P, \text{ иначе.} \end{cases} \quad (13)$$

Точка O представляет нулевой элемент группы точек эллиптической кривой.

Для сведения данной задачи к задаче «ВЫПОЛНИМОСТЬ» воспользуемся следующим разложением, которое во многом аналогично выражению (6):

$$kQ = (\ldots(k_1 Q + k_2\{2Q\}) + \ldots) + k_N\{2^{N-1}Q\}, \quad (14)$$

где k_i (i=1,…,N) – биты искомой экспоненты.

Выражения, заключённые в фигурные скобки, не зависят от неизвестных величин, поэтому нет необходимости кодировать алгоритм их вычисления в виде КНФ.

Без ограничения общности можно считать, что точки $2^i Q$ отличны от точки O. В противном случае, начиная с некоторого номера, все слагаемые в правой части (14) равнялись бы O, что существенно упрощало бы решаемую задачу.

Однако в сумме (14) возможно появление O за счёт того, что отдельные k_i обращаются в нуль.

Итак, для представления правой части (14) в виде КНФ необходимо закодировать следующие операции:

1. сложение двух точек эллиптической группы;
2. деление в поле $GF(P)$ (необходимо в (13) для представления λ);
3. линейную комбинацию точек кривой:

$$aU + bV = cR, \quad (15)$$

где a, b и c могут принимать значение 0 и 1, U, V и R точки эллиптической кривой.

Для представления в виде КНФ суммы двух точек эллиптической кривой от уравнений (12), (13) переходим к следующей системе уравнений:

$$\begin{cases} W = X_V - X_U \pmod P, \\ T = Y_V - Y_U \pmod P, \\ L = T / W \pmod P, \\ S = X_V + X_U \pmod P, \\ K = LL \pmod P, \\ X_R = K - S \pmod P, \\ D = X_U - X_R \pmod P, \\ M = DL \pmod P, \\ Y_R = M - Y_U \pmod P, \end{cases} \quad (16)$$

где X_U, Y_U, X_V, Y_V, X_R, Y_R – координаты точек, W, T, L, S, K, D, M – вспомогательные числовые переменные.

Для кодирования в виде КНФ операции деления предлагается следующий подход: строится КНФ представление операции умножения в поле *GF(P)* (7).

Далее в построенную КНФ подставляются значения битов U, R и P. Полученная КНФ, очевидно, является КНФ представлением для операции деления в поле *GF(P)*.

Для кодирования в виде КНФ линейной комбинации двух точек используется следующая логическая функция:

$$r = (t \wedge c) \vee (f \wedge \bar{c}). \quad (17)$$

Данная функция по построению обладает следующими свойствами:

1. если литерал c равен «истина», то значение литерала r есть значение t;
2. если литерал c равен «ложь», то значение литерала r есть значение f.

Воспользуемся следующей эквивалентностью: $x = y \Leftrightarrow (x \vee \bar{y}) \wedge (\bar{x} \vee y)$, а также законом дистрибутивности булевой алгебры для преобразования выражения (17) к следующему виду:

$$\left(r \vee (\bar{t}\bar{f})\right) \vee (\bar{t}c) \vee (\bar{f}\bar{c})) \wedge (\bar{r} \vee (tc) \vee (f\bar{c})). \quad (18)$$

Полученное выражение преобразуется в КНФ аналогично операциям сложения по модулю 2 и переноса с использованием (2) и правила де-Моргана.

Алгоритм сведения задачи логарифмирования в группе точек эллиптической кривой к задаче «ВЫПОЛНИМОСТЬ» принимает на вход: числа A B, P, имеющие N разрядов в двоичной записи и являющиеся параметрами эллиптической кривой; точки соответствующей эллиптической кривой Q и R, а также литералы $k_1,k_2,\ldots,k_N$, отвечающие битам искомой экспоненты. Основные шаги алгоритма:

1. вычислить точки $2^{i-1}Q$, при i=1,…,N;
2. при i=1,…,N-1 и S_1=Q и c_1=k_1 генерировать КНФ эквивалентную следующему выражению:

$$\begin{cases} c_{i+1}S_{i+1} = c_i S_i + k_i \{2^{i-1}Q\}, \\ c_{i+1} = c_i \vee k_i; \end{cases} \quad (19)$$

3. в полученную КНФ подставить биты P в качестве значений литералов p_i и биты R в качестве значений S_N, а также значения всех $2^{i-1}Q$, при i=1,…,N.

Представленный алгоритм является консервативным, полиномиальным сведением задачи логарифмирования на эллиптической кривой к задаче «ВЫПОЛНИМОСТЬ», обладающим следующими свойствами:

1. время работы алгоритма есть $O(N^4)$, при этом основной вклад вносит вычисление точек $2^{i-1}Q$, сама генерация КНФ за счёт эффективного кодирования деления в поле *GF(P)* имеет трудоемкость $O(N^3)$;
2. количество литералов в КНФ есть $O(N^3)$;
3. количество скобок в КНФ есть $O(N^3)$.

2. Результаты вычислительных экспериментов, выводы

Для оценки практических свойств полученных алгоритмов были проведены следующие виды вычислительных экспериментов (использовалась ПЭВМ со следующими характеристиками CPU: 1,7ГГц; RAM: 2ГБ):

1. Генерация КНФ для задач практически значимых размерностей (т.е. размерностей реально используемых в различных криптосистемах). Полученные результаты представлены в таблице 1. Таким образом, можно утверждать, что построенные алгоритмы действительно пригодны для сведения рассматриваемых задач актуальных размерностей к задаче «ВЫПОЛНИМОСТЬ».

Таблица 1.Сведение рассматриваемых задач к задаче «ВЫПОЛНИМОСТЬ»

Размерность задачи	**Количество литералов**	**Количество дизъюнктов**	**Время генерации** (ч:мин:сек)
Задача факторизации			
2048	6,5E5	1,6E7	0:08:05
8192	1,0E7	2,6E8	5:59:50
Задача дискретного логарифмирования			
128	1,0E7	1,8E8	2:18:08
152	1,7E7	3,0E8	7:12:06
Задача логарифмирования на эллиптической кривой			
70	5,3E6	8,9E7	03:06:16
100	1,5E7	2,6E8	10:24:00

2. Решение полученных экземпляров задачи «ВЫПОЛНИМОСТЬ» с помощью различных современных решателей (победителей конкурса SAT-Competition 2007 года [12]). Полученные результаты представлены в таблице 2. Из приведенных данных следует, что с помощью современных SAT – решателей общего назначения возможно решение задач весьма скромных размерностей, несопоставимых с размерностями, рекомендованными к использованию в криптографических приложениях.

Таблица 2. Решение полученных КНФ с помощью SAT-решателей, победителей конкурса SAT-Competition 2007

Алгоритм решателя	**Время решения** (ч:мин:сек)		
КНФ для задачи факторизации			
Размерность задачи	*80*	*96*	*112*
march_ks	0:01:45	0:03:26	0:36:20
picosat	0:01:40	0:10:53	-
3-КНФ для задачи факторизации			
Размерность задачи	*88*	*96*	*104*
vallst	0:57:28	1:51:50	4:26:34
picosat	0:02:18	2:21:39	5:26:48
КНФ для задачи дискретного логарифмирования			
Размерность задачи	*10*	*12*	*14*
SatElite	0:00:14	0:09:00	1:16:46
rsat	0:00:33	0:07:56	1:23:23
КНФ для задачи логарифмирования на эллиптической кривой			
Размерность задачи	*8*	*10*	*12*
rsat	0:06:51	0:49:42	>5:00:00
minisat	0:02:50	1:00:41	>5:00:00

3. Исследование стойкости рассматриваемых задач к восстановлению полного ключа по его известным фрагментам. Было установлено, что для задачи факторизации сложность решения соответствующей задачи «ВЫПОЛНИМОСТЬ» существенно уменьшается при подстановке фрагментов ключа. Например, выполняющий набор для КНФ представления задачи факторизации числа размером 512 бит находится за 8 мин. после подстановки в неё значений 47% случайно выбранных битов ключа. Для задач дискретного логарифмирования и логарифмирования на эллиптической кривой такой закономерности обнаружено не было. Например, КНФ представление задачи дискретного логарифмирования размерности 24 бита после подстановки 50% случайно выбранных бит ключа ни один из использованных SAT – решателей не смог решить за отведённые 4 часа, при том, что задача размерности 12 бит решается менее чем за 10 минут (см. табл. 2).

Литература

1. **Cook S. A., Mitchel D. G.** Finding hard instances for the satisfiability problem // A survey. DIMACS series in discrete mathematics and theoretical computer science. – 1997. V. 5. – P. 151.
2. **Massacci F., Marraro L.** Towards the formal verification of ciphers: Logical cryptanalysis of DES // Proc. Third LICS Workshop on Formal Methods and Security Protocols, Federated Logic Conferences, 1999.
3. **Šušem M., Janičič P.** Uniform reduction of cryptographic problems to SAT // Faculty of Mathematics, University of Belgrade, Serbia, 2009. URL: http://argo.matf.bg.ac.yu/events/2009/slides/ (дата обращения: 12.10.2009).
4. **Семенов А. А.** Логико-эвристический подход в криптоанализе генераторов двоичных последовательностей // Труды международной научной конференции ПАВТ‘07. – 2007. - Т. 1. – С. 170–180.
5. **Беспалов Д. В., Семёнов А. А.** О логических выражениях для задачи 2-ФАКТОРИЗАЦИЯ // Вычислительные технологии. – 2002. – Т. 7.
6. **Srebrny M.** Factorization with sat – classical propositional calculus as a programming environment // Faculty of Mathematics Informatics and Mechanics at the University of Warsaw. 2004. URL: http://www.mimuw.edu.pl/~mati/fsat-20040420.pdf (дата обращения: 06.07.2009).
7. **Дулькейт В. И., Файзуллин Р. Т., Хныкин И. Г.** Непрерывные аппроксимации решения задачи ВЫПОЛНИМОСТЬ применительно к криптографическому анализу асимметричных шифров // Компьютерная оптика. – 2009. – Т. 33, № 1. – С. 86–91.
8. **Дулькейт В. И., Файзуллин Р. Т., Хныкин И. Г.** Алгоритм минимизации функционала, ассоциированного с задачей 3-sat и его практические применения // Компьютерная оптика. – 2008. – Т. 32, № 1. – С. 68–73.
9. **Салаев Е. В., Файзуллин Р. Т.** Применение метода последовательных приближений с инерцией к решению задачи Выполнимость // Вестник Томского Государственного Университета. – 2006. – Т. 17.
10. **Столингс В.** Криптография и защита сетей: принципы и практика. - М.: Вильямс, 2001.
11. **Jurisic A., Menezes A.** Elliptic curves and cryptography. // Dr. Dobb’s Journal. – April, 1997.
12. Sat competition [Сайт]. URL: http://www.satcompetition.org (дата обращения: 10.08.2009).

References

1. **Cook S.A., Mitchel D.G.** Finding hard instances for the satisfiability problem // A survey. DIMACS series in discrete mathematics and theoretical computer science. – 1997. – V. 5. – P. 151.
2. **Massacci F., Marraro L.** Towards the formal verification of ciphers: Logical cryptanalysis of DES // Proc. Third LICS Workshop on Formal Methods and Security Protocols, Federated Logic Conferences. – 1999.
3. **Šušem M., Janičič P.** Uniform reduction of cryptographic problems to SAT // Faculty of Mathematics, University of Belgrade, Serbia; 2009. URL: http://argo.matf.bg.ac.yu/events/2009/slides/
4. **Semenov A.A.** Logical and heuristically approach in cryptanalysis of binary sequences generators // Proc. international scientific conference PAVT‘07. – 2007. – V. 1. – P. 170–180. – (in Russian).
5. **Bespalov D.V., Semjonov A.A.** About logical statements for 2-FACTORIZATION problem // Calculation technologies. – 2002. – V. 7. – (in Russian).
6. **Srebrny M.** Factorization with sat – classical propositional calculus as a programming environment // Faculty of Mathematics Informatics and Mechanics at the University of Warsaw. 2004. URL: http://www.mimuw.edu.pl/~mati/fsat-20040420.pdf

7. **Dulkeyt V.I., Faizullin R.T., Khnykin I.G.** Continuous approximation of SAT decision as applied to cryptographic analysis of asymmetric ciphers // Computer Optics. – 2009. – V. 33, N 1. – P. 86–91. – (in Russian).
8. **Dulkeyt V.I., Faizullin R.T., Khnykin I.G.** Algorithm for minimization of functional associated with 3-SAT problem and its practical usage // Computer Optics. – 2008. – V. 32, N 1. – P. 68–73. – (in Russian).
9. **Salaev E.V., Faizullin R.T.** Using the method of sequential approximations with inertia for solving SAT problems // Herald of Tomsk State University. – 2006. - V. 17. – (in Russian).
10. **Stallings W.** Cryptography and network Security. - Moscow: Williams, 2001 – (in Russian).
11. **Jurisic A., Menezes A.** Elliptic curves and cryptography. // Dr. Dobb's Journal. – April, 1997.
12. Sat competition. URL: http://www.satcompetition.org.

REDUCTION OF FACTORIZATION, DISCREET LOGARITHM AND ELLIPTIC CURVE LOGARITHM PROBLEMS TO SOLVING ASSOCIATED SATISFIABILITY PROBLEMS

Vladimir Igorevitch Dulkeyt[1] (post-graduate, e-mail: vidulkeyt@mail.ru)
[1] F.M. Dostoevsky Omsk State University

Abstract

The algorithms of conservative reduction of factorization, discreet logarithm and elliptic curve logarithm problems to satisfiability problem (SAT) were proposed. The capabilities of modern SAT-solvers for solving produced SAT instances were investigated. The resistance to whole key repairing by its fragments was investigated for considered problems.

Key words: CNF, factorization, discreet logarithm, elliptic curve logarithm, SAT.

Поступила в редакцию 28.12.2009 г.