

О ШЕСТОЙ МЕЖДУНАРОДНОЙ ОЛИМПИАДЕ
ПО КРИПТОГРАФИИ NSUCRYPTO

*А. А. Городилова^{1, a}, Н. Н. Токарева^{1, 2}, С. В. Агиевич³,
К. Карле⁴, Е. В. Горкунов^{1, 5}, В. А. Идрисова¹, Н. А. Коломеец¹,
А. В. Куценко^{1, 5}, Р. К. Лебедев⁵, С. Никова⁶, А. К. Облаухов¹,
И. А. Панкратова⁷, М. А. Пудовкина⁸, В. Реймен⁶, А. Н. Удовенко⁹*

¹ Институт математики им. С. Л. Соболева СО РАН,
пр. Ак. Коптюга, 4, 630090 Новосибирск, Россия

² Лаборатория криптографии JetBrains Research,
ул. Пирогова, 1, 630090 Новосибирск, Россия

³ Белорусский гос. университет,
пр. Независимости, 4, 220030 Минск, Беларусь

⁴ University of Paris 8,
Rue de la Liberte, 2, 93200 Saint-Denis, France

⁵ Новосибирский гос. университет,
ул. Пирогова, 2, 630090 Новосибирск, Россия

⁶ ESAT-COSIC, KU Leuven,
Kasteelpark Arenberg, 10, B-3001 Leuven, Belgium

⁷ Томский гос. университет,
пр. Ленина, 36, 634050 Томск, Россия

⁸ Московский гос. технический университет им. Н. Э. Баумана,
ул. 2-я Бауманская, 5/1, 105005 Москва, Россия

⁹ SnT, University of Luxembourg,
Avenue de l'Universite, 2, L-4365 Esch-sur-Alzette, Luxembourg

E-mail: ^ansucrypto@nsu.ru

Работа первого, второго и шестого авторов выполнена при поддержке Математического центра в Академгородке (соглашение № 075–15–2019–1613 с Министерством науки и высшего образования РФ) и лаборатории криптографии JetBrains Research; пятого автора — в рамках гос. задания ИМ СО РАН (проект № 0314–2019–0016); седьмого, восьмого и одиннадцатого авторов — при поддержке Российского фонда фундаментальных исследований (проекты № 20–31–70043, 18–07–01394, 19–31–90093).

© А. А. Городилова, Н. Н. Токарева, С. В. Агиевич, К. Карле, Е. В. Горкунов, В. А. Идрисова, Н. А. Коломеец, А. В. Куценко, Р. К. Лебедев, С. Никова, А. К. Облаухов, И. А. Панкратова, М. А. Пудовкина, В. Реймен, А. Н. Удовенко, 2020

Аннотация. Представлены задачи Шестой международной олимпиады по криптографии NSUCRYPTO'2019 вместе с их решениями. Рассмотренные задачи связаны с атаками на шифры и хэш-функции, протоколами, булевыми функциями, полиномами Диксона, простыми числами, роторными машинами и т. д. Обсуждаются несколько открытых проблем по математическим мерам противодействия атакам по сторонним каналам, APN-инволюциям, S-блокам и т. д. Задача о поиске коллизии для хэш-функции *Cur127* была частично решена во время олимпиады. Табл. 11, ил. 7, библиогр. 21.

Ключевые слова: криптография, шифр, хэш-функция, код Хэмминга, слайдовая атака, пороговая реализация, полином Диксона, APN-функция, олимпиада, NSUCRYPTO.

Введение

NSUCRYPTO (Non-Stop University Crypto) — международная олимпиада по криптографии, которая в 2019 г. проводилась в шестой раз. В программный комитет олимпиады входят специалисты из Бельгии, Франции, Нидерландов, США, Норвегии, Индии, Люксембурга, Беларуси, Казахстана и России.

Кратко опишем формат олимпиады. Принять участие может любой желающий. Регистрируясь на сайте олимпиады [1], участник выбирает свою категорию: «школьник» (для старшеклассников), «студент» (для учащихся университетов) и «профессионал» (категория без ограничений). Награждение победителей проводится в каждой категории отдельно.

В олимпиаде два независимых интернет-тура: индивидуальный (длится 4,5 часа) и командный (1 неделя). Первый тур проводится в двух секциях: А — для «школьников», В — для «студентов» и «профессионалов». Второй тур общий для всех участников. Участники получают задания и загружают свои решения на сайте олимпиады. Язык олимпиады — английский.

На олимпиаде предлагаются не только интересные задачи с известными решениями, но и нерешённые проблемы в этой области. В этом году одна из таких задач, «Cur127» (см. п. 2.14), была частично решена во втором раунде. Полный список открытых проблем, представленных за всю историю олимпиады, можно найти на сайте олимпиады [2], где также отмечено текущее состояние каждой задачи. Например, помимо «Cur127», три команды решили задачу «Матрицы Сильвестра» в 2018 г., а задача «Алгебраическая иммунность» была полностью решена в 2016 г. Некоторые участники продолжают поиск решений и после олимпиады.

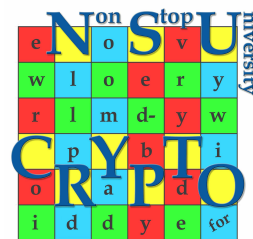


Рис. 1. Логотип

Например, в [3] было предложено частичное решение задачи «Разделение секрета» (2014 г.).

Структура статьи следующая: в разд. 1 приводится список задач олимпиады, далее в разд. 2 даются все формулировки задач и приводятся их подробные решения, наконец, в разд. 3 приводится список победителей олимпиады NSUCRYPTO'2019.

Математические задачи с их решениями всех предыдущих международных олимпиад по криптографии NSUCRYPTO с 2014 по 2018 гг. можно найти в [4–8].

1. Список задач олимпиады

В общей сложности на олимпиаде было предложено 16 задач, некоторые из которых вошли в оба раунда (табл. 1, 2). Секция А первого раунда состояла из 6 задач, секция В включала 7 задач. Три задачи были общими для обеих секций. Второй раунд содержал 11 задач, 5 из которых включали нерешённые вопросы (решения таких задач награждаются специальными призами от программного комитета).

Таблица 1

Задачи первого раунда

Секция А			Секция В		
№	Название задачи	Баллы	№	Название задачи	Баллы
1	1024-битный ключ	4	1	Осенние листья	4
2	Магнитная буря	4	2	Магнитная буря	4
3	Осенние листья	4	3	Роторная машина	4
4	Роторная машина	4	4	16QAM	8
5	Сломанный Calculator	4	5	Обещание и деньги	6
6	Обещание	6	6	Calculator	6
			7	APN + инволюции	7

Таблица 2

Задачи второго раунда

№	Название задачи	Баллы
1	1024-битный ключ	4
2	Разбиение	6 + баллы за откр. вопросы
3	Факторизция в 2019	8
4	TwinPeaks-3	8
5	Curl127	10 + баллы за откр. вопросы

6	8-битный S-блок	неограниченно (откр. вопрос)
7	Роторная машина	4
8	16QAM	8
9	Calculator	6
10	APN + инволюции (расшир.)	12 + баллы за откр. вопросы
11	Гипотеза	неограниченно (откр. вопрос)

2. Задачи и решения

В этом разделе мы сформулируем все задачи NSUCRYPTO'2019 и приведём их подробные решения, уделяя внимание решениям участников.

2.1. Задача «1024-битный ключ».

ФОРМУЛИРОВКА. У Алисы есть 1024-битный ключ для симметричного шифра, состоящий из нулей и единиц. Алиса боится злоумышленников, поэтому она изменяет ключ каждый день следующим образом.

1. Алиса выбирает подпоследовательность b битов ключа так, что первый и последний биты в ней равны нулю; также может быть $b = 0$.
2. Алиса инвертирует все биты в b ($0 \leftrightarrow 1$); биты ключа вне b остаются прежними.

Докажите, что данный процесс остановится. Определите ключ, который в итоге получит Алиса.

Пример операции: ключ 1100101101110011... преобразуется в ключ 1100110010001011....

РЕШЕНИЕ. Закодируем двоичный вектор ключа соответствующим десятичным числом. Очевидно, что каждый следующий день это число будет увеличиваться. Действительно, биты слева от выбранной подпоследовательности не изменяются, а первый её бит меняется с 0 на 1. Заметим, что это число не может увеличиваться бесконечно, поскольку размер ключа ограничен 1024 битами. Следовательно, ключ будет максимально возможным, т. е. будет состоять из всех 1.

Почти все участники успешно решили задачу.

2.2. Задача «Магнитная буря».

ФОРМУЛИРОВКА. Аппаратный генератор случайных чисел — устройство, генерирующее случайные последовательности, состоящие из нулей и единиц. К несчастью, на генератор повлияло возмущение, вызванное магнитной бурей. В результате устройство сгенерировало последовательность нулей длины k (где k — положительное целое число), а затем начало генерировать бесконечную последовательность единиц.

Докажите, что в определённый момент генератор выдаст число вида $1 \dots 10 \dots 0$, которое делится на 2019.

РЕШЕНИЕ. Докажем, что найдётся число вида $1 \dots 10 \dots 0$, которое делится на 2019. Рассмотрим все числа, состоящие только из единиц. Поскольку таких чисел бесконечно много, среди них найдутся A и B , которые имеют одинаковый остаток от деления на 2019. Тогда число $C = A - B = 1 \dots 10 \dots 0$, содержащее m единиц для некоторого натурального m , делится на 2019. Так как 2019 не делится на 2 и 5, то $C^* = C \times 10 \dots 0 = 1 \dots 10 \dots 0$ делится на 2019 для любого числа нулей.


Было получено большое число правильных решений от участников.

2.3. Задача «Осенние листья».

ФОРМУЛИРОВКА. Прочитайте скрытое сообщение (рис. 2).



Рис. 2. Осенние листья

РЕШЕНИЕ. На рис. 2 видим различные листья и пробелы между ними. Это подсказывает, что здесь использовался простой подстановочный шифр и листья соответствуют различным английским буквам. Согласно английской грамматике предположим, что второе и третье слова — это *is a*. Тогда первое слово начинается с *a* и по своей структуре может быть *autumn* (что весьма вероятно, поскольку изображён осенний пейзаж). Кроме того, лист  является наиболее частой буквой в тексте; можно предположить, что это *e*. Далее видим **ea** в третьей строке, что напоминает *leaf*. В результате последнее слово становится *fl**e**, т. е. *flower*. Наконец, получаем *Autumn is a second spring when every leaf is a flower*, что является известной цитатой Альбера Камю.

Практически все участники олимпиады прочитали сообщение.

2.4. Задача «Роторная машина».

ФОРМУЛИРОВКА. В одной стране для шифрования информации были полезны роторные машины (рис. 3). Ева знает, что для некоторой секретной коммуникации использовалась простая роторная машина. Она работает только с буквами O, P, R, S, T, Y и состоит из входного колеса с лампами (start), одного ротора и рефлектора (рис. 4).

Входное колесо и рефлектор зафиксированы, в то время как ротор может быть в одной из 6 возможных позиций. После нажатия клавиши на клавиатуре электрический сигнал, соответствующий букве, проходит через машину, приходит обратно на входное колесо, и соответствующая лампа показывает результат зашифрования. После того как очередная буква зашифрована, ротор поворачивается вправо (т. е. по часовой стрелке) на 60° . Точки разного цвета (пронумерованы) на сторонах ротора указывают на различные непересекающиеся сигнальные линии внутри ротора.

Например, если ротор зафиксирован в положении как на рис. 4, то при нажатии клавиши O, буква будет зашифрована как T (сигнал попадёт в ротор через красную точку (цвет 1), отразится через рефлектор и затем вернётся в ротор через фиолетовую точку (цвет 5)). Если вы снова нажмёте O, то она будет зашифрована как R. Затем при нажатии T, вы получите S и т. д.



Рис. 3. Примеры роторных машин

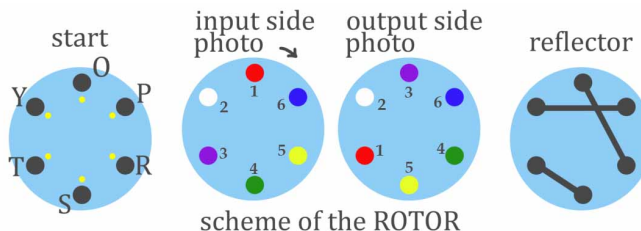


Рис. 4. Схема ротора

Ева перехватила секретное сообщение **TRRYSSPRYRYROYTOPTOPTSPSPRS**. Помогите ей расшифровать его, помня, что исходного положения ротора Ева не знает.

РЕШЕНИЕ. Для того чтобы решить задачу и дешифровать сообщение, требуется корректно понять схему работы. Ключ шифра — исходное положение ротора. Обозначим его цветом (номером) точки, соответствующей букве **O**, на входной стороне ротора. Таблица шифрования в зависимости от ключа представлена в табл. 3.

Таблица 3

Таблица шифрования

№	Цвет	O	P	R	S	T	Y	№	Цвет	O	P	R	S	T	Y
1	красный	T	Y	S	R	O	P	4	зелёный	S	R	P	O	Y	T
2	белый	R	S	O	P	Y	T	5	жёлтый	S	T	Y	O	P	R
3	фиолетовый	Y	R	P	T	S	O	6	синий	R	T	O	Y	P	S

Опробовав все шесть ключей, находим единственное осмысленное сообщение **POST TO TOP OOPS SORRY STOP ROTOR** для «жёлтого» ключа.

Почти все участники решили задачу. Наиболее интересные решения включали в себя реальные модели роторной машины, сделанные участниками, например, школьницей Варварой Лебединской (Россия, СУНЦ НГУ), командой Кристины Геут, Сергея Титова и Дмитрия Ананичева (Россия, УрГУПС, УрФУ).

2.5. Задача «Сломанный Calculator».

ФОРМУЛИРОВКА. Алиса и Боб практикуются в разработке простых криптографических приложений для смартфонов. В этом году они создали приложение **Calculator**, которое позволяет выполнять следующие операции по модулю 2019:

- вводить не более чем 4-значное десятичное целое число (≥ 0);
- выполнять сложение, вычитание и умножение двух чисел;
- сохранять результаты вычислений и считывать их из памяти.

Предположим, что Алиса хочет отправить Бобу шифртекст y (4-значное целое положительное число). Для этого она отправляет y с её смартфона в память **Calculator** на смартфоне Боба. Чтобы расшифровать y , Бобу требуется вычислить открытый текст x (используя его **Calculator**) по правилу: x равен остатку от деления $f(y) = y^5 + 1909y^3 + 401y$ на 2019.

К сожалению, Боб уронил смартфон и разбил его экран (рис. 5). Кнопка **+** и все кнопки цифр кроме **1** и **5** перестали работать.

Помогите Бобу в сложившейся ситуации придумать эффективный алгоритм расшифрования любого шифртекста y , используя **Calculator**.

Более точно, предложите короткий список команд, где каждая команда имеет один из следующих типов ($1 \leq j, k < i$):

$$S_i = y, \quad S_i = a, \quad S_i = S_j - S_k, \quad S_i = S_j * S_k.$$

Здесь a — не более чем 4-значное целое число, состоящее только из цифр 1 и 5: например, $a = 1$, $a = 15$, $a = 551$, $a = 5115$ и т. д.

Первая команда обязательно должна быть $S_1 = y$. В результате последней команды должен вычисляться открытый текст x . Напомним, что все вычисления идут по модулю 2019. В частности, число 2500 становится равным 481, а -1000 становится равным 1019 сразу после ввода или вычислений. Чем короче будет ваш список команд, тем больше баллов вы сможете получить за эту задачу.

Например, список команд в табл. 4 позволяет вычислить $x = y^2 - 55$.

Таблица 4

Команда	Результат
$S_1 = y$	y
$S_2 = S_1 * S_1$	y^2
$S_3 = 11$	11
$S_4 = 5$	5
$S_5 = S_3 * S_4$	55
$S_6 = S_2 - S_5$	$y^2 - 55$



Рис. 5. Сломанный Calculator

РЕШЕНИЕ. Приведём решение авторов задачи, содержащее 9 шагов. Запись $a \equiv_m b$ означает, что a и b сравнимы по модулю m . Справедливы следующие сравнения:

$$\begin{aligned}
 f(y) &\equiv_{2019} y^5 + 1909y^3 + 401y \equiv_{2019} y(y^4 - 110y^2 + 401) \\
 &\equiv_{2019} y(y^4 - 2 * 55y^2 + 55^2 - 55^2 + 401) \\
 &\equiv_{2019} y((y^2 - 55)^2 - 55^2 + 5 * 22^2) \\
 &\equiv_{2019} y((y^2 - 55)^2 - 11^2 * (5^2 - 5 * 2^2)) \\
 &\equiv_{2019} y((y^2 - 55)^2 - 11^2 * 5) \equiv_{2019} y((y^2 - 55)^2 - 11 * 55).
 \end{aligned}$$

Таким образом, остаток от деления $f(y)$ на 2019 можно вычислить для любого y с помощью списка команд из табл. 5.

Такое же решение было найдено Бориславом Кириловым (Болгария, Первая частная математическая гимназия).

Замечание. Многочлен $f(y) = y^5 + 1909y^3 + 401y$ представляет собой многочлен Диксона $D_5(y, a) = y^5 - 5y^3a + 5ya^2$ при $a = 22$ с коэффициентами, взятыми по модулю 2019.

Таблица 5

Список команд для решения от авторов задачи

Команда	Результат	Команда	Результат
$S_1 = y$	y	$S_6 = 11$	11
$S_2 = S_1 * S_1$	y^2	$S_7 = S_3 * S_6$	$11 * 55$
$S_3 = 55$	55	$S_8 = S_5 - S_7$	$(y^2 - 55)^2 - 11 * 55$
$S_4 = S_2 - S_3$	$y^2 - 55$	$S_9 = S_1 * S_8$	$y((y^2 - 55)^2 - 11 * 55)$
$S_5 = S_4 * S_4$	$(y^2 - 55)^2$		

2.6. Задача «Calculator».

ФОРМУЛИРОВКА. Алиса и Боб практикуются в разработке простых криптографических приложений для смартфонов. В этом году они создали приложение **Calculator**, которое позволяет выполнять следующие операции по модулю 2019:

- вводить не более чем 4-значное десятичное целое число (≥ 0);
- выполнять сложение, вычитание и умножение двух чисел;
- сохранять результаты вычислений и считывать их из памяти.

Предположим, что Алиса хочет отправить Бобу шифртекст y (4-значное целое положительное число). Для этого она отправляет y с её смартфона в память **Calculator** на смартфоне Боба. Чтобы расшифровать y , Бобу требуется вычислить открытый текст x (используя его **Calculator**) по правилу: $x = f(y) \bmod 2019$, где f — секретный многочлен, известный только Алисе и Бобу.

В самый неподходящий момент Боб уронил смартфон и разбил его экран (рис. 6). Кнопка $\boxed{+}$ и все кнопки цифр кроме $\boxed{2}$ перестали работать.

Помогите Бобу в сложившейся ситуации придумать эффективный алгоритм расшифрования любого шифртекста y , используя **Calculator**, если текущий секретный многочлен равен: $f(y) = y^5 + 1909y^3 + 401y$. Более точно, предложите короткий список команд, где каждая команда имеет один из следующих типов ($1 \leq j, k < i$):

$$S_i = y, \quad S_i = 2, \quad S_i = 222, \quad S_i = S_j - S_k, \\ S_i = 22, \quad S_i = 2222, \quad S_i = S_j * S_k.$$

Первая команда обязательно должна быть $S_1 = y$. В результате последней команды должен вычисляться открытый текст x . Напомним, что все вычисления идут по модулю 2019. В частности, число 2222 становится равным 203 сразу после ввода. Чем короче будет ваш список команд, тем больше баллов вы сможете получить за эту задачу.

Например, список команд в табл. 6 позволяет вычислить $x = y^2 - 4$.

Таблица 6

Команда	Результат
$S_1 = y$	y
$S_2 = S_1 * S_1$	y^2
$S_3 = 2$	2
$S_4 = S_3 * S_3$	4
$S_5 = S_2 - S_4$	$y^2 - 4$



Рис. 6. Сломанный Calculator

РЕШЕНИЕ. Многочлен $f(y) = y^5 + 1909y^3 + 401y$ представляет собой многочлен Диксона $D_5(y, a) = y^5 - 5y^3a + 5ya^2$ при $a = 22$ с коэффициентами, взятыми по модулю 2019. Справедливы следующие равенства:

$$\begin{aligned} D_5(y, a) &= yD_4(y, a) - aD_3(y, a) = yD_2(D_2(y, a), a^2) - aD_3(y, a) \\ &= y((y^2 - 2a)^2 - 2a^2) - ay(y^2 - 2a - a). \end{aligned}$$

При $a = 22$ значение $f(y)$ можно вычислить для любого y с помощью списка команд из табл. 7.

Таблица 7

Список команд для решения от авторов задачи

Команда	Результат	Команда	Результат
$S_1 = y$	y	$S_8 = S_7 * S_7$	$(y^2 - 2a)^2$
$S_2 = 2$	2	$S_9 = S_8 - S_5$	$(y^2 - 2a)^2 - 2a^2$
$S_3 = 22$	a	$S_{10} = S_1 * S_9$	$y((y^2 - 2a)^2 - 2a^2)$
$S_4 = S_2 * S_3$	$2a$	$S_{11} = S_7 - S_2$	$y^2 - 2a - a$
$S_5 = S_3 * S_4$	$2a^2$	$S_{12} = S_1 * S_{11}$	$y(y^2 - 2a - a)$
$S_6 = S_1 * S_1$	y^2	$S_{13} = S_3 * S_{12}$	$ay(y^2 - 2a - a)$
$S_7 = S_6 - S_4$	$y^2 - 2a$	$S_{14} = S_{10} - S_{13}$	$f(y)$

Таблица 8

Список команд для решения в 11 шагов

Команда	Результат	Команда	Результат
$S_1 = y$	y	$S_7 = S_6 - S_4$	$y^2 - 44 - 22$
$S_2 = S_1 * S_1$	y^2	$S_8 = S_6 * S_7$	$(y^2 - 44) * (y^2 - 44 - 22)$
$S_3 = 2$	2	$S_9 = S_4 * S_4$	22^2
$S_4 = 22$	22	$S_{10} = S_8 - S_9$	$(y^2 - 44) * (y^2 - 44 - 22) - 22^2$
$S_5 = S_3 * S_4$	44	$S_{11} = S_1 * S_{10}$	$f(y)$
$S_6 = S_2 - S_5$	$y^2 - 44$		

Отметим, что среди прочих участники представили решения, содержащие 11 и 13 шагов. Оба этих решения были отмечены дополнительными баллами. Мадалина Болбочеану (Румыния, Bitdefender) нашла решение в 11 шагов во время первого раунда (табл. 8). Команда Хеннинга Зайдлера и Кати Штумпп (Германия, Берлинский технический университет) представила решение в 13 шагов во втором раунде. Эти решения основаны на представлении $f(y) = y((y^2 - 44)(y^2 - 66) - 22^2)$.

2.7. Задача «Обещание».

ФОРМУЛИРОВКА. Юные криптографы Алиса, Боб и Кэрл заинтересованы в квантовых вычислениях и очень хотят купить квантовый компьютер. Один миллионер подарил им некоторое количество денег (скажем, X_A для Алисы, X_B для Боба и X_C для Кэрл). Кроме того, миллионер взял с ребят обещание, что они никому, включая друг друга, не расскажут, сколько денег каждый из них получил.

- Сможете ли вы помочь криптографам придумать алгоритм, позволяющий определить, хватит ли их общей суммы денег $X_A + X_B + X_C$ для покупки квантового компьютера, не нарушая данного ими обещания?

- Какие слабости у вашего алгоритма (если кто-то нарушит обещание)? Всегда ли он защищает секрет честных участников от нечестных?

РЕШЕНИЕ. Эта задача представляет собой частный случай задачи «Обещание и деньги» для трёх участников (см. п. 2.8).

2.8. Задача «Обещание и деньги».

ФОРМУЛИРОВКА. Несколько юных криптографов интересуются квантовыми вычислениями и очень хотят купить квантовый компьютер. Миллионер подарил им некоторое количество денег. Скажем, для каждого из n криптографов количество денег равно X_i , $i = 1, \dots, n$. Кроме того, миллионер взял с ребят обещание, что они не расскажут никому, включая друг друга, сколько денег каждый из них получил.

- Сможете ли вы помочь криптографам придумать алгоритм, позволяющий определить, хватит ли их общей суммы денег $\sum_{i=1}^n X_i$ для покупки квантового компьютера, не нарушая данного ими обещания?

- Как вы считаете, существуют ли такие алгоритмы, которые защищают секрет честных участников от нечестных?

- Какие слабости у вашего алгоритма (если кто-то нарушит обещание)? Всегда ли он защищает секрет честных участников от нечестных?

РЕШЕНИЕ. Опишем идею решения, представленного Михаилом Кудиновым (Россия, МГТУ им. Н. Э. Баумана).

Для начала предполагается, что ни один участник не может купить квантовый компьютер без других участников. Допустим, что N' — это то количество денег, которое требуется каждому для покупки компьютера, и $N = nN'$, где n — число участников. Миллионер дал каждому из участников денег в количестве X_i , $i \in \{1, \dots, n\}$.

Каждый участник выбирает случайно и равновероятно секреты $s_{i,j}$ так, что

$$\sum_{j=1}^n s_{i,j} \equiv X_i \pmod{N}.$$

Далее каждый из них передаёт по секретному каналу долю $s_{i,j}$ владельцу суммы X_j . После этого, владелец X_i имеет доли $s_{k,i}$ для всех $k \in \{1, \dots, n\}$. Очевидно, что

$$\sum_{j=1}^n \sum_{i=1}^n s_{i,j} \equiv \sum_{i=1}^n X_i \pmod{N}.$$

При первом предположении все участники смогут вместе посчитать общую имеющуюся сумму денег.

Основной недостаток алгоритма (в дополнение к предположению) заключается в том, что требуется большое число личных обменов сообщениями (хотя число ключей может быть n для асимметричных схем).

Многие участники также предлагали алгоритмы, похожие на алгоритм расчёта средней зарплаты Шнайера [9]. В общем случае все такие алгоритмы уязвимы, если нечестны $n - 1$ участников. Некоторые участники пробовали использовать криптосистему, гомоморфную по сложению и сохраняющую отношение $<$.

Несмотря на большое число поступивших решений каждое из них имело те или иные недостатки. Это не позволило выбрать «лучшее из лучших» решение на 6 баллов, так что максимальная оценка составила 5 баллов. Всего оказалось девять таких решений.

В школьной секции первого раунда было предложено решить частный случай этой задачи для $n = 3$ (оценивание было менее строгим).

2.9. Задача «16QAM».

ФОРМУЛИРОВКА. Алиса и Боб пользуются оптоволоконной линией связи с применением технологии 16QAM. Сообщения кодируются алфавитом из 16 символов, каждый из которых — двоичное слово кода Грэя длины 4. В результате помех на линии связи возможны одиночные ошибки в словах кода Грэя. Как-то раз Алиса дочитала интересную книгу и решила поделиться с Бобом восторгом от прочитанного. Для большей убедительности Алиса отправила Бобу отрывок из книги (на английском

Таблица 9

Полученные Бобом последовательности

Часть 1	Часть 2
66674C36666F43D3C199900AA1AA325992A	66CA61967319CCD2CE76998CE6433332D19
67A59D9B4A8B69330D1BC000153367A5E33	B46784C65334E999A402ADA0265A99A6633
D30E6692D0F349D3321FFFF0ED706667A7F	33319B32D3299698CCC96986619967134CC
670D999679F4AA67561BA679B4AA54F34D5	B4CE2333334CC6730CE90170CCCD2CE669
AB0F4AACCF000055CE633670D9DA54CE37F	996A61999EA63332CCA4C3332D4CD3334CC
660DE19CD995335495523CCAA8F1E03325	D3319994730CCCD3A6669D96A66999699B3
86CF48A98CD9B387FD9D546A99E9D200033	98640CC86CE619676AD4CD3308999866D33
3201513FE5B4AA00CCCE9667554CD2CCCB3	79321C33210B4C6732199B53218019A404C
330F32A666553CD756AC3E0674E9D369E1D	D2DE65A986663398CCCCB5319CC6665997
C6A9999780007F00961E66465519FEA8B25	B96A63398CD9CCD2CD9A399A66339866619
14CCCB332AA63332CCCE6D2A99AACCC004	98CD9CC325A6339CCE619998C04C66CE633
	996A61998CF66967334CC66CA6199865E(0) ₂

языке). Зная особенности линии связи, Алиса разделила текст на две части и отправила их по отдельности. В первую она поместила 16 согласных букв, встречающихся в тексте, а во вторую — гласные (в т. ч. букву «у»), пробел, дефис и знаки препинания. После этого Алиса закодировала символы кодом Хэмминга длины 7 с проверочной матрицей в лексикографическом порядке.

В письме Боб получил две последовательности, записанные в 16-ричной форме (табл. 9). Также Боб получил ряд чисел 22, 19, 3, 3, 36, 53, 3, 33, 20, 28. Каждое число определяет, сколько согласных содержится между знаками пунктуации.

Восстановите текст и определите главного героя книги, прочитанной Алисой.

РЕШЕНИЕ. Некоторые детали задачи несущественны. А именно, можно опустить шаг с использованием кода Грея и считать, что Алиса заменяет каждый символ 7-битным кодовым словом кода Хэмминга в каждой части открытого текста.

Основная идея взлома шифра, используемого Алисой и Бобом, заключается в проведении частотного анализа для каждой части шифртекста. Это помогает восстановить вероятные значения часто встречающихся символов и форм слов. Тщательный поиск комбинаций согласных и гласных, встречающихся в словах английского алфавита, расширяет частичное решение. Частотный анализ биграмм также помогает расшифровать часть текста. Наконец, можно поискать в сети Интернет фрагмент книги, отправленный Алисой Бобу.

Рассмотрим одно из возможных решений. Алиса использует код Хэмминга с проверочной матрицей H и порождающей матрицей G :

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Сначала запишем шифртекст в двоичном виде. Разделим его на 7-битные слова и исправим ошибки с помощью проверочной матрицы H . Декодировем слова кода Хэмминга в 4-битные слова кода Грея (хотя это необязательный шаг решения). Посчитаем частоты встречаемости кодовых слов в каждой части данного шифртекста (табл. 10).

Таблица 10

Частоты встречаемости кодовых слов кода Хэмминга в тексте

Часть 1			Часть 2		
Код Грея	Код Хэмминга	Частота	Код Грея	Код Хэмминга	Частота
1011	0110011	46	0100	1001100	85
0010	0101010	30	1011	0110011	50
1001	0011001	24	1001	0011001	33
0001	1101001	24	0001	1101001	26
0011	1000011	19	1010	1011010	17
0000	0000000	15	0011	1000011	9
0110	1100110	13	0000	0000000	8
1100	0111100	8	1110	0010110	7
1111	1111111	8	1100	0111100	2
1101	1010101	7	0010	0101010	1
0100	1001100	6	1000	1110000	1
1110	0010110	5	0111	0001111	0
1010	1011010	5	0101	0100101	0
0101	0100101	4	1101	1010101	0
1000	1110000	4	0110	1100110	0
0111	0001111	2	1111	1111111	0

Сравним полученные частоты с частотами символов английского алфавита. Подходящее частотное распределение можно найти в [10], на которое ссылаются в [11]. Согласно [10], список букв от наиболее к наименее часто встречаемым выглядит следующим образом:

e t a o i n s h r d l c u m w f g y p b v k j x q z. (*)

Начнём с части 2. Предположим, что пробел — наиболее часто встречающийся символ. Заметим, что за большинством знаков препинания следует пробел, в отличие от дефиса, который обычно окружён буквами. Используя частоты букв, определяем вероятные пробелы, гласные и дефис и строим следующее частичное решение для этой части открытого текста (знак # заменяет знаки препинания):

ее ae e oe o e ua iaia# e oo oy-oy i o ea ee# u# ea# auae o
 ie ea o e aoy a oe o i a i eae# a i o o o eae a oo o i o iee
 ay ue aeii o aa aie# uuay# e uai uy oy oe i a e ea i e eae#
 i e ee oeee o e a a ee a# e e a uy ee e i a e oe o ee a a#

Обратимся к части 1. Для упрощения записи переобозначим 7-битные кодовые слова кода Хэмминга 16-ричными числами от 0 до F в порядке, указанном в табл. 10 (часть 1). Получаем шифртекст из 220 символов, разделённый на 10 частей согласно цепочке чисел из условия:

023402C43E0251412B0103 02C1B32407551003703 4A3 B46
 33A4884CE02E804020631094106311739943
 1675510A0040C1068047266101D10619FF56D4031A00048090103 355
 025108B315023021A3020246102173994 E2333C72410275585D46
 021281BD102021A0202631016055

Сопоставим частоты символов в части 1 с частотами согласных букв в английском алфавите. Первые пять пар вероятно будут 0 — t, 1 — n, 2 — s/h, 3 — s/h, 4 — r. Биграмма th наиболее часто встречающаяся в английских текстах, поэтому допустим, что 2 означает h, а 3 — s.

Совместим полученные частичные решения и попробуем восстановить некоторые части открытого текста (табл. 11). Несложно увидеть слова **these are the** в начале (1), а также, что **the** — первое слово в (2) и (8).

Таблица 11

Частичный открытый текст

№	Текст
(1)	thsrtthCrSEth5nrnhBtnts ее ae e oe o e ua iaia#
(2)	thCnBshrt755ntts7ts e oo oy-oy i o ea ee#
(3)	rAs u#
(4)	Br6 ea#
(5)	ssAr88rCEthE8trtht6snt9rnt6snn7s99rs auae o ie ea o e aoy a oe o i a i eae#
(6)	n6755ntAttrtCnt68tr7h66ntnDnt6n9FF56DrtsnAttr8t9tnts a i o o o eae a oo o i o iee ay ue aeii o aa aie#
(7)	s55 uuay#
(8)	th5nt8Bsn5thsthAsththr6nthn7s99r e uai uy oy oe i a e ea i e eae#
(9)	EhsssC7hrnth75585Dr6 i e ee oeee o e a a ee a#
(10)	thnh8nBDnththnAthth6sntn6t55 e e a uy ee e i a e oe o ee a a#

Лучшая идея для следующего шага — поискать в словаре английского языка слова, которые содержат гласные в установленном порядке. Можно использовать один из инструментов для распознавания шаблонов, доступных в Интернете, например, [12]. Некоторые участники олимпиады сами реализовали подобные программы.

Так, в (7) согласные `s55` и гласные `uaa` приводят к подходящему слову `usually`, поэтому допускаем, что `5` означает букву `l`. Гласные `uaae` и двойная `s` в (5) дают два варианта: `assuage` и `sausage`. В любом случае, вероятно `A` заменяет `g`, так что в (3) находим `rugs`. Шаблон `uai` и согласные `5nt8` дают `lunatic` в (8), поэтому запишем, что `8` есть `s`.

Далее вновь обратимся к частотному распределению букв [10]. В части 1 шифртекста `t n h s r l 6 7/c` — наиболее часто встречающиеся восемь символов. Согласно (*) наиболее вероятно, что `6` означает `d`. Тогда в (4) имеем `Brd` и `ea`, что даёт возможные слова `beard` и `bread`. Так что вероятно `B` есть `b`.

Тщательный анализ оставшегося зашифрованного текста и поиск слов по шаблонам и количеству букв в конечном итоге приводят к открытому тексту (с заменой знаков препинания на `#`):

```
these are the mores of the lunar inhabitants# the moon
boy-shorty will not eat sweets# rugs# bread# sausage or ice
cream of the factory that does not print ads in newspapers#
and will not go to treatment a doctor who did not invented
any puzzle advertising to attract patients# usually#
the lunatic buys only those things that he read in the newspaper#
if he sees somewhere on the wall a clever ad#
then he can buy even the thing that he does not need at all#
```

Это отрывок из сказочного романа русского писателя Николая Носова «Незнайка на Луне». Главный герой романа — коротышка Незнайка.

Задача была полностью решена 13 командами во втором раунде и Самуэлем Тангом (Гонконг, Black Bauhinia) в первом раунде. Лучшие решения предложили команда Ирины Слонкиной, Михаила Сорокина и Владимира Боброва (Россия, МГТУ им. Н. Э. Баумана), а также команда Владимира Папроцкого, Дмитрия Зарембо и Карины Круглик (Беларусь, Белорусский государственный университет).

2.10. Задача «APN + инволюции (расширенная)». Первые три вопроса Q1, Q2, Q3 были даны как задача «APN + инволюции» в первом раунде. Расширенная версия задачи для второго раунда включала также вопрос Q4 с открытыми проблемами.

ФОРМУЛИРОВКА. Алиса хочет построить блочный шифр с использованием инволюций в качестве подкомпонент; это сводит к минимуму разницу между алгоритмами шифрования и расшифрования. Она знает, что

APN-перестановки — лучший выбор подкомпонент для защиты от атак, основанных на дифференциальной технике. Она хочет найти набор APN-перестановок, являющихся инволюциями для каждого $n \geq 2$.

Алиса знает, что любую инволюцию можно выразить как произведение непересекающихся транспозиций, поэтому она решила рассматривать инволюцию в следующем виде:

$$g = \prod_{i=1}^d (\alpha_i, \alpha'_i),$$

где $\{\alpha_i, \alpha'_i\} \cap \{\alpha_j, \alpha'_j\} = \emptyset$ для всех $i, j \in \{1, \dots, d\}$, $i \neq j$, $1 \leq d \leq 2^{n-1}$.

Помогите Алисе найти APN-перестановки среди таких инволюций g . Найдите ответы на следующие вопросы.

Q1. Пусть g — APN-перестановка,

$$\Lambda(g) = \{\alpha_i \oplus \alpha'_i \mid i = 1, \dots, d\}, \quad \widehat{\Lambda}(g) = [\alpha_i \oplus \alpha'_i \mid i = 1, \dots, d],$$

$$B(g) = \{x \oplus y \mid \{x, y\} \subseteq \text{FixP}(g), x \neq y\},$$

$$\widehat{B}(g) = [x \oplus y \mid \{x, y\} \subseteq \text{FixP}(g), x \neq y],$$

где $\text{FixP}(g)$ — множество всех неподвижных точек g , т. е.

$$\text{FixP}(g) = \{x \in \mathbb{F}_2^n \mid g(x) = x\}.$$

Найдите необходимые условия для множеств $\Lambda(g)$, $B(g)$ и мультимножеств $\widehat{\Lambda}(g)$, $\widehat{B}(g)$. Докажите, что если найденные условия не выполняются, то g не является APN-перестановкой.

Q2. Пусть g — инволюция и APN-функция,

$$d_{a,b}(g) = |\{x \in \mathbb{F}_2^n \mid g(x \oplus a) \oplus g(x) = b\}|, \quad a, b \in \mathbb{F}_2^n.$$

Найдите $d_{a,a}(g)$ для любого ненулевого $a \in \mathbb{F}_2^n$.

Q3. Найдите нетривиальную верхнюю границу на $|\text{FixP}(g)|$.

Q4. Пусть M_n — множество всех n -битных APN-инволюций.

(a) Найдите мощность множества M_n для $n = 2, 3, 4$.

(b) Найдите мощность множества M_n для $n = 5$.

(c) Бонусная задача (доп. баллы, спец. приз). Пусть $n \geq 6$. Найдите нижние и верхние границы на мощность M_n . Опишите инволюции из M_n . Предложите конструкции инволюций из M_n .

Заметим, что отображение $x \mapsto x^{-1}$ в поле Галуа $GF(2^n)$ принадлежит M_n для всех нечётных $n \geq 3$.

Замечание. Напомним основные определения.

• $\mathbb{F}_2^n = \{x = (x_1, \dots, x_n) \mid x_i \in \mathbb{F}_2\}$ — векторное пространство размерности n над $\mathbb{F}_2 = \{0, 1\}$.

- Для $x, y \in \mathbb{F}_2^n$ сумма определяется как $x \oplus y = (x_1 \oplus y_1, \dots, x_n \oplus y_n)$, где \oplus означает сложение по модулю 2.
- Пусть $\widehat{X} = [x_1, \dots, x_d]$ — мультимножество над базовым множеством \mathbb{F}_2^n , т. е. среди элементов $x_1, \dots, x_d \in \mathbb{F}_2^n$ могут быть одинаковые.
- *Перестановка* s — это отображение из \mathbb{F}_2^n в \mathbb{F}_2^n такое, что $s(x) \neq s(y)$ для всех $x, y \in \mathbb{F}_2^n$, $x \neq y$.
- *Инволюция* s — это перестановка, обратная к которой совпадает с s , т. е. $s^2(x) = s(s(x)) = x$ для любого $x \in \mathbb{F}_2^n$.
- Для любых различных векторов $\alpha, \beta \in \mathbb{F}_2^n$ перестановка s называется *транспозицией*, если $s(\alpha) = \beta$, $s(\beta) = \alpha$ и $s(x) = x$ для всех $x \in \mathbb{F}_2^n \setminus \{\alpha, \beta\}$; обозначается $s = (\alpha, \beta)$.
- Перестановка s называется *APN-перестановкой* (almost perfect non-linear), если уравнение $s(x \oplus a) \oplus s(x) = b$ имеет не более двух решений для любого ненулевого вектора $a \in \mathbb{F}_2^n$ и любого вектора $b \in \mathbb{F}_2^n$.

РЕШЕНИЕ. Q1. Пусть $a \in \Lambda(g)$. Следовательно, $a = x \oplus y$, где $y = g(x)$ и $(x, y) = (\alpha_i, \alpha'_i)$ для некоторого i . Тогда

$$g(x \oplus a) = g(y) = x = y \oplus a = g(x) \oplus a.$$

Пусть $a \in B(g)$. Следовательно, $a = x \oplus y$, где $x, y \in \text{FixP}(g)$. Тогда

$$g(x \oplus a) = g(y) = y = x \oplus a = g(x) \oplus a.$$

Таким образом, $d_{a,a}(g) \geq 2$ для любого вектора $a \in \Lambda(g) \cup B(g)$.

Пусть g — APN-перестановка. Тогда $d_{a,a}(g) = 2$. Следовательно, кратность каждого элемента из $\Lambda(g)$ и $B(g)$ равна 1. Таким образом, $\Lambda(g) = \widehat{\Lambda}(g)$ и $B(g) = \widehat{B}(g)$. Заметим, что $\Lambda(g) \cap B(g) = \emptyset$.

Q2. Так как g — APN-перестановка, то $d_{a,a}(g) \leq 2$. В Q1 получили, что $d_{a,a}(g) = 2$ для любого вектора $a \in \Lambda(g) \cup B(g)$. Докажем, что $d_{a,a}(g) = 0$ для $a \notin \Lambda(g) \cup B(g)$.

Пусть a — ненулевой вектор и x — решение $g(x \oplus a) \oplus g(x) = a$. Так как g — перестановка, либо $x \in \text{FixP}(g)$, либо $x = \alpha_i$ ($x = \alpha'_i$) для некоторого i . Рассмотрим два случая.

1. Пусть $x \in \text{FixP}(g)$. Тогда $g(x \oplus a) \oplus g(x) = a$ влечёт $g(x \oplus a) = x \oplus a$. Значит, $x \oplus a \in \text{FixP}(g)$. Как следствие, $a \in B(g)$.

2. Без ограничения общности полагаем, что $x = \alpha_i$ для некоторого i и $y = x \oplus a$. Если $y \in \text{FixP}(g)$, то $g(x \oplus a) \oplus g(x) = a$ влечёт $g(x) = x$; противоречие. Следовательно, без ограничения общности $y = \alpha'_j$ для некоторого j (поэтому $\alpha_i \oplus \alpha'_j = a$). Тогда

$$g(\alpha_i \oplus a) \oplus g(\alpha_i) = a \Rightarrow g(\alpha'_j) \oplus \alpha'_i = a \Rightarrow \alpha_j \oplus \alpha'_i = a.$$

Покажем, что α'_i и α_j также являются решениями. Действительно,

$$\begin{aligned} g(\alpha'_i \oplus a) \oplus g(\alpha'_i) &= g(\alpha_j) \oplus \alpha_i = \alpha'_j \oplus \alpha_i = a, \\ g(\alpha_j \oplus a) \oplus g(\alpha_j) &= g(\alpha'_i) \oplus \alpha'_j = \alpha_i \oplus \alpha'_j = a. \end{aligned}$$

Таким образом, если $i \neq j$, то получаем по крайней мере три решения, что противоречит APN-свойству перестановки g . Следовательно, $j = i$ и $a \in \Lambda(g)$.

Q3. Докажем, что $|\text{FixP}(g)| \leq 1 + (2^{n-1} - 1)^{1/2}$.

Инволюция g является APN-перестановкой. Согласно Q1 имеем

$$B(g) \cap \Lambda(g) = \emptyset. \quad (1)$$

Пусть $q = |\text{FixP}(g)|$. Так как g — инволюция, получаем, что q чётно. Из уравнения (1) и $\Lambda(g) \cup B(g) \subseteq \mathbb{F}_2^n \setminus \{\mathbf{0}\}$ следует, что

$$|\Lambda(g)| + |B(g)| \leq 2^n - 1. \quad (2)$$

Так как $|B(g)| = \binom{q}{2}$, $|\Lambda(g)| = 2^{n-1} - q/2$, имеем

$$q(q-1)/2 + 2^{n-1} - q/2 \leq 2^n - 1,$$

откуда

$$q \leq 1 + (2^{n-1} - 1)^{1/2}.$$

Q4. (а) Можно вычислить, что $M_2 = \emptyset$ и $|M_3| = 224$. Далее, известно [13], что нет APN-перестановок при $n = 4$. Следовательно, $M_4 = \emptyset$.

(б) Напомним, что функция $A: \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ *аффинная*, если $A(x \oplus y) = A(x) \oplus A(y) \oplus A(\mathbf{0})$ для любых $x, y \in \mathbb{F}_2^n$. Две функции $F, G: \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ *аффинно эквивалентны*, если существуют аффинные перестановки A_1, A_2 такие, что $G = A_1 \circ F \circ A_2$. Свойство быть APN-перестановкой инвариантно относительно аффинной эквивалентности. Существуют [13] пять классов аффинной эквивалентности APN-перестановок при $n = 5$. Более того, согласно [13, теорема 3] только один класс содержит функции вместе с их обратными функциями. Следовательно, только этот класс APN-перестановок может содержать инволюции. В частности, он содержит функцию обращения в конечном поле: $F(x) = x^{-1}$ для $x \neq 0$, $F(0) = 0$ (здесь пространство \mathbb{F}_2^n рассматривается как конечное поле порядка 2^n). Таким образом, при $n = 5$ все APN-инволюции аффинно эквивалентны функции обращения.

(с) К сожалению, участниками не было предложено интересных идей по данным открытым вопросам.

Полное решение в первом раунде предложил только Хеннинг Зайдлер (Германия, Берлинский технический университет). Во втором раунде лучшее решение на 11 баллов было найдено командой Кристины Геут, Сергея Титова и Дмитрия Ананичева (Россия, УрГУПС, УрФУ).

2.11. Задача «Разбиение».

ФОРМУЛИРОВКА. Боб интересуется математическими средствами защиты от атак по сторонним каналам на блочные шифры. Ему стало известно о методе специальных разложений функций. Теперь он размышляет над следующей математической задачей данного подхода.

Пусть \mathcal{F} обозначает множество обратимых функций (перестановок) из \mathbb{F}_2^4 в \mathbb{F}_2^4 и \mathcal{F}^n обозначает множество обратимых функций из $(\mathbb{F}_2^4)^n$ в $(\mathbb{F}_2^4)^n$. Пусть $F \in \mathcal{F}^n$ записывается в виде

$$F(x_1, \dots, x_n) = (F_1(x_1, \dots, x_n), \dots, F_n(x_1, \dots, x_n)),$$

здесь $F_i: (\mathbb{F}_2^4)^n \rightarrow \mathbb{F}_2^4$, $i = 1, \dots, n$, — компонентные функции.

Для произвольной $f \in \mathcal{F}$ функция $F \in \mathcal{F}^n$ называется *разбиением* f , если для любого $(x_1, x_2, \dots, x_n) \in (\mathbb{F}_2^4)^n$

$$\sum_{i=1}^n F_i(x_1, x_2, \dots, x_n) = f\left(\sum_{i=1}^n x_i\right).$$

Более того, F — *неполное* разбиение f , если F — разбиение f с условием, что каждая компонентная функция F_i не зависит от x_i .

Бобу требуется ваша помощь в изучении функций, для которых существуют неполные разбиения. Найдите ответы на следующие вопросы.

Q1. Пусть \mathcal{A} — множество аффинных функций из \mathbb{F}_2^4 в \mathbb{F}_2^4 . Две функции $f, g \in \mathcal{F}$ *аффинно эквивалентны*, если существуют $a, b \in \mathcal{A}$ такие, что $g = b \circ f \circ a$. Пусть $f, g \in \mathcal{F}$ — две функции из одного класса аффинной эквивалентности и $F \in \mathcal{F}^n$ — неполное разбиение f . Получите с помощью F неполное разбиение для g .

Все функции в одном классе аффинной эквивалентности имеют одинаковую степень. Известно [14], что данное отношение эквивалентности делит \mathcal{F} на 302 класса: 1 класс соответствует \mathcal{A} , 6 классов содержат квадратичные функции, 295 содержат кубические функции.

Кроме того, Боб знает, что при $n \geq 5$ неполное разбиение существует для любой $f \in \mathcal{F}$; при $n = 2$ — только для функций из \mathcal{A} ; при $n = 3$ — для \mathcal{A} и для 5 из 6 классов эквивалентности, содержащих квадратичные функции; при $n = 4$ — для \mathcal{A} , для всех 6 квадратичных классов эквивалентности, а также для 5 кубических классов.

Q2. Бонусная задача (доп. баллы, спец. приз). Найдите точное математическое свойство, которым должна обладать функция $f \in \mathcal{F}$, чтобы для неё существовало неполное разбиение F при $n = 3, 4$.

Q3. Бонусная задача (доп. баллы, спец. приз). Обобщите на случай функций над $\mathbb{F}_2^5, \mathbb{F}_2^6$.

РЕШЕНИЕ. Q1. Пусть $f, g \in \mathcal{F}$ — аффинно эквивалентные функции, при этом $g = b \circ f \circ a$ для некоторых $a, b \in \mathcal{A}$, и пусть $F \in \mathcal{F}^n$ — неполное разбиение f . Для начала заметим, что функции f, g обратимы, а значит, отображения a, b также обратимы. Обозначим

$$a(x) = Ax + a', \quad b(x) = Bx + b', \quad x \in \mathbb{F}_2^4,$$

где A, B — невырожденные двоичные (4×4) -матрицы и $a', b' \in \mathbb{F}_2^4$. По компонентным функциям $\{F_i\}_{i=1}^n$ функции F определим обратимую функцию $G \in \mathcal{F}^n$ с компонентными функциями (здесь $j = 1, 2, \dots, n$)

$$G_j(x_1, x_2, \dots, x_n) = \begin{cases} BF_1(Ax_1 + a', Ax_2, \dots, Ax_n) + b', & \text{если } j = 1, \\ BF_j(Ax_1 + a', Ax_2, \dots, Ax_n), & \text{если } j \neq 1. \end{cases}$$

Тогда для любого $(x_1, x_2, \dots, x_n) \in (\mathbb{F}_2^4)^n$ выполнено

$$\begin{aligned} \sum_{j=1}^n G_j(x_1, x_2, \dots, x_n) &= BF_1(Ax_1 + a', Ax_2, \dots, Ax_n) + b' \\ &\quad + \sum_{j=2}^n BF_j(Ax_1 + a', Ax_2, \dots, Ax_n) \\ &= B \left(\sum_{j=1}^n F_j(Ax_1 + a', Ax_2, \dots, Ax_n) \right) + b' \\ &= Bf(Ax_1 + a' + Ax_2 + \dots + Ax_n) + b' \\ &= Bf \left[A \left(\sum_{i=1}^n x_i \right) + a' \right] + b' = b \circ f \circ a \left(\sum_{i=1}^n x_i \right) = g \left(\sum_{i=1}^n x_i \right). \end{aligned}$$

Следовательно, функция $G \in \mathcal{F}^n$ является разбиением g .

Из неполноты F следует, что G_j , которая по сути является аффинным преобразованием F_j , не зависит от x_j . Значит, G — неполное разбиение g .

Q2, Q3. Данные открытые задачи не были решены в течение олимпиады. Тем не менее, было предложено одно перспективное решение. Команда Виктории Власовой, Михаила Полякова и Алексея Чиликова (Россия, МГТУ им. Н. Э. Баумана) нашла достаточное условие существования неполного разбиения при $n = 3$. Кратко опишем его.

Пусть $\text{wt}(y)$ — вес Хэмминга двоичного вектора y . Для $\sigma \in \mathbb{F}_2$ введём

$$\delta_\sigma(y) = \begin{cases} y, & \text{если } \sigma = 1, \\ \mathbf{0}, & \text{если } \sigma = 0, \end{cases}$$

где $\mathbf{0}$ — нулевой вектор той же размерности, что и y .

Пусть V — векторное пространство над полем K . Предположим, что

$$\sum_{\sigma \in \mathbb{F}_2^n} (-1)^{\text{wt}(\sigma)} f\left(\sum_{i=1}^n \delta_{\sigma_i}(x_i)\right) = 0 \quad (3)$$

для обратимой функции $f: V \rightarrow V$. Тогда существует неполное разбиение для f . Далее рассмотрим случай $n = 3$. Для любого $(x_1, x_2, x_3) \in V^3$ положим

$$\begin{aligned} F_1(x_1, x_2, x_3) &= f(x_2) - f(x_2 + x_3), \\ F_2(x_1, x_2, x_3) &= f(x_3) - f(x_1 + x_3), \\ F_3(x_1, x_2, x_3) &= f(x_1) - f(x_1 + x_2). \end{aligned}$$

Ясно, что функция $F_i: V^3 \rightarrow V$ не зависит от x_i , $i = 1, 2, 3$. Рассмотрим равенства

$$\begin{aligned} &F_1(x_1, x_2, x_3) + F_2(x_1, x_2, x_3) + F_3(x_1, x_2, x_3) \\ &= f(x_2) - f(x_2 + x_3) + f(x_3) - f(x_3 + x_1) + f(x_1) - f(x_1 + x_2) \\ &= \sum_{\sigma \in \mathbb{F}_2^3} (-1)^{\text{wt}(\sigma)} f\left(\sum_{i=1}^3 \delta_{\sigma_i}(x_i)\right) + f(x_1 + x_2 + x_3) - f(0) \\ &= f(x_1 + x_2 + x_3) - f(0). \end{aligned}$$

Без ограничения общности полагаем, что $f(0) = 0$. Иначе можем рассматривать исходную задачу для функции $g(x) = f(x) - f(0)$ с $g(0) = 0$. Согласно Q1 функция g имеет неполное разбиение тогда и только тогда, когда его имеет f . Таким образом, получаем требуемое:

$$F_1(x_1, x_2, x_3) + F_2(x_1, x_2, x_3) + F_3(x_1, x_2, x_3) = f(x_1 + x_2 + x_3).$$

Авторами также было показано, что условие (3) необходимо для существования неполного разбиения f для любого n . Положив $V = \mathbb{F}_2^m$ с $m = 4, 5, 6$ и $K = \mathbb{F}_2$, можно получить решение Q2, Q3 для $n = 3$.

2.12. Задача «Факторизация в 2019».

ФОРМУЛИРОВКА. Николь изучает криптосистему RSA. Она выбрала случайные 500-битные простые числа p и q , $2^{499} \leq p, q < 2^{500}$, и вычислила $n = p \cdot q$. Как любопытный и творческий человек, она также забавным образом соединила эти три числа. Её любимое число — целое число h такое, что

$$h \equiv 3^{2019} p^2 + 5^{2019} q^2 \pmod{n^2 + 8 \cdot 2019}.$$

К сожалению, она потеряла листок, где записаны её простые числа. К счастью, она помнит n и h :

$n = 4076361302550483684524984004483156158356462640553515813866703$
 $7187916726709053088608443040552850196515077288316636771660924$
 $7516155419756121537288444995708421977847213953345126368990185$
 $2711025976018935658830540651908064758287421268759621419191593$
 $672520947172224181322892513146475004919963234000020193827,$
 $h = 7830799927833657758696152811024002692382891492752691194950119$
 $6645494977563735699853935546611327171983687170931118125666490$
 $3117342818449633588647098544612151278035131454234786653136500$
 $8870883047099654288891241821353207362290372720539680784860373$
 $5835726536308836859069167015873622366491268957196566632938255$
 $0122397088799629252601249428062432254738935764304610281613264$
 $2256417499027286468001256009599212578383223023458925765092934$
 $8364268481174940654635292018596007475218929572581040331954410$
 $1402343236581529201392185327635674923459290749241831590661903$
 $9651325142154451518308886658505820006667836934411881.$

Помогите Николь восстановить p и q .

РЕШЕНИЕ. Задача основана на (упрощённом) варианте метода Коперсмита. Пусть $m = n^2 + 8 \cdot 2019$. Это составное число с неизвестными множителями. Идея заключается в том, чтобы найти целое a такое, что

$$a_1 = a \cdot 3^{2019} \bmod m, \quad a_2 = a \cdot 5^{2019} \bmod m$$

достаточно малы, а число $a_1 p^2 + a_2 q^2$ ненамного превышает модуль m и может быть восстановлено из $a \cdot h \bmod m$. Это можно сделать с помощью алгоритма Лагранжа — Гаусса (который является частным случаем и строительным блоком алгоритма LLL).

Пусть Λ — решётка, натянутая на векторы

$$v_1 = (1, 5^{2019} \cdot (3^{2019})^{-1} \bmod m), \quad v_2 = (0, m).$$

Рассмотрим случайный вектор $v = (a_1, a_2) \in \Lambda$. Легко проверить, что

$$a_1 p^2 + a_2 q^2 \equiv a_1 \cdot h \cdot (3^{2019})^{-1} \pmod{m}.$$

Редукция решётки гарантирует нахождение такого вектора v с нормой

$$\|v\| = \sqrt{a_1^2 + a_2^2} \leq 2^{(d-1)/4} (\det \Lambda)^{1/d} = \sqrt{m} / \sqrt[4]{2},$$

где $d = 2$ — размерность решётки. В частности,

$$|a_1 p^2 + a_2 q^2| \leq n(p^2 + q^2) < n(p + q)^2 < 10n^2,$$

где два последних неравенства следуют из сбалансированности простых чисел: $\max(p, q) \leq 2 \min(p, q)$.

Из этого следует, что существует целое число z , $|z| < 10$, такое, что

$$a_1 \cdot h \cdot (3^{2019})^{-1} \bmod m + zm = a_1 p^2 + a_2 q^2.$$

В результате имеем уравнение от p^2 и q^2 . Заменяя $p = n/q$, получаем биквадратное уравнение от q , которое легко решить и разложить n . В итоге

$$\begin{aligned} p &= 2019000075878154181681129810414477022346818209175194524879 \\ &\quad 2088909215011445470480079537222712856903502641160815792411 \\ &\quad 89587393202602664199899594021414383, \\ q &= 2019000073973494194521339805682093959182265746083995594826 \\ &\quad 3937536316692891758278516666680141671194393865432898509407 \\ &\quad 34885806826120718179729242641026893. \end{aligned}$$

Лучшее решение предложила команда Алексея Зеленецкого, Михаила Кудинова и Дениса Набокова (Россия, МГТУ им. Н. Э. Баумана).

2.13. Задача «TwinPeaks-3».

ФОРМУЛИРОВКА. Шифр Боба TwinPeaks-2 (с олимпиады NSUCRYPTO'2018) вновь был взломан, поэтому он, наконец, решил прочитать книги по криптографии. Новый шифр основан на практических шифрах, а количество раундов уменьшено для повышения производительности. Боб не только использовал лучшие методы, но и решил усилить шифр за счёт скрытности, поэтому раундовые функции теперь неизвестны. Известно только, что они одинаковы для нечётных и чётных раундов.

Новый шифр Боба работает следующим образом. Сообщение X представляется двоичным словом длины 128 и делится на четыре 32-битных слова a, b, c, d . Затем 32 раза применяется раундовое преобразование

$$(a, b, c, d) \leftarrow (b, c, d, a \oplus (F_i(b, c, d))),$$

где $F_i = F_1$ для нечётных раундов и $F_i = F_2$ для чётных раундов — секретные функции, принимающие на вход три 32-битных слова и возвращающие одно слово, а \oplus — побитовая операция XOR. Шифртекст Y для сообщения X представляет собой конкатенацию финальных a, b, c, d .

Агент Купер вновь хочет читать сообщения Боба. Он перехватил шифртекст

$$Y = \text{e473f19a247429ab33b66268d57dd241}$$

(текст дан в шестнадцатеричной системе счисления, первый байт — e4).

Кроме того, у него есть доступ к тестовому серверу Боба, где реализовано шифрование и расшифрование на секретном ключе. Сервер доступен по ссылке [15]. К сожалению, версия доступного программного обеспечения на сервере не финальная, поэтому процедура расшифрования неполная: в ней только используются ключи в обратном порядке,

чего недостаточно для расшифрования:

$$(a, b, c, d) \leftarrow (b, c, d, a \oplus (F_i(b, c, d))),$$

где $F_i = F_2$ для нечётных раундов, $F_i = F_1$ для чётных раундов.

Сервер также может обрабатывать несколько блоков текста одновременно: они будут обрабатываться один за другим, а затем объединяться, как в обычном режиме шифрования ECB. Сообщения передаются и обрабатываются сервером в шестнадцатеричной системе счисления.

Помогите Куперу дешифровать Y .

РЕШЕНИЕ. Пусть f_i — раундовое преобразование раунда i :

$$f_i: (a, b, c, d) \leftarrow (b, c, d, a \oplus (F_{k(i)}(b, c, d))),$$

где $k(i) = 1$ для нечётных i и $k(i) = 2$ для остальных. Тогда функцию зашифрования E можно представить в виде $E = (f_1 f_2)^{16}$.

Пусть I — неполная функция расшифрования, описанная в условии задачи. Зашифрование и неполное расшифрование различаются только в порядке ключей, поэтому $I = (f_2 f_1)^{16}$. Вместе с тем функция расшифрования имеет вид $E^{-1} = (f_2^{-1} f_1^{-1})^{16}$, где f_i^{-1} — обращение f_i , заданное следующим преобразованием:

$$f_i^{-1}: (a, b, c, d) \leftarrow (d \oplus (F_{k(i)}(a, b, c)), a, b, c).$$

Таким образом, для того чтобы применить E^{-1} к шифртексту, необходимо вычислить $F_1(x, y, z)$ и $F_2(x, y, z)$, которые секретны. Можно использовать слайдовую атаку, чтобы восстановить данные функции.

Основная идея — найти слова $x = (x_1, x_2, x_3, x_4)$ и $y = (y_1, y_2, y_3, y_4)$ такие, что $f_i(x) = y$. Как только такая пара будет найдена, F_i можно получить как

$$F_i(x_2, x_3, x_4) = y_4 \oplus x_1.$$

Будем использовать следующую идею, чтобы найти необходимую пару: если $E f_i(x) = E(y)$, то $f_i(x) = y$. Начнём с F_1 . Найдём пару x и y такую, что $E f_1(x) = E(y)$. Перепишем это равенство:

$$E f_1(x) = (f_1 f_2)^{16} f_1(x) = f_1 (f_2 f_1)^{16}(x) = f_1 I(x) = E(y).$$

Приходим к выводу, что если $f_1 I(x) = E(y)$, то $f_1(x) = y$. Условие $f_1 I(x) = E(y)$ можно проверить, используя определение f_1 : если

$$(I(x))_2 = (E(y))_1, \quad (I(x))_3 = (E(y))_2, \quad (I(x))_4 = (E(y))_3,$$

то вероятно, что $f_1 I(x) = E(y)$. Вероятность ложных срабатываний приближённо равна 2^{-96} для случайной функции F_i , поэтому её можно считать незначительной. И $I(x)$, и $E(y)$ доступны в оракуле шифрования для произвольных x и y в качестве процедур неполного расшифрования и зашифрования соответственно.

Чтобы найти $F_i(a, b, c)$, подберём x и y в виде $x = (X, a, b, c)$ и $y = (a, b, c, X')$. Согласно парадоксу дней рождения желаемая пара может быть найдена в среднем за $2 \cdot 2^{16}$ операций (вместо 2^{32} , если мы зафиксируем X или X' некоторым постоянным значением).

Как только найдём такую пару x и y , сможем вычислить $F_1(a, b, c)$, применить f_1^{-1} к шифртексту и расшифровать последний раунд. Затем аналогично находим F_2 , меняя I и E местами в силу симметрии. Раунд за раундом расшифровываем шифртекст и получаем исходное сообщение (в шестнадцатеричной системе счисления):

acherryplease.

Эталонная реализация этой атаки требует шифрования 2^{22} блоков текста и 10 минут в среднем. Важно использовать возможность сервера обрабатывать несколько блоков текста одновременно, чтобы минимизировать количество HTTP-запросов.

Четыре команды успешно решили задачу аналогичным методом.

2.14. Задача «Curl27».

ФОРМУЛИРОВКА. Боб занимается развитием инфраструктуры ЗОТА и разработал для неё функцию хэширования Curl27. Необычным свойством инфраструктуры является использование троичной логики: вместо битов используются триты — элементы множества $T = \{-1, 0, 1\}$, а вместо двоичных слов — троичные. Функция Curl27 определяется ниже. Её реализацию на языке Java можно найти по ссылке [16].

Найдите коллизию для Curl27 — различные троичные слова X и X' такие, что $\text{Curl27}(X) = \text{Curl27}(X')$. Коллизионные сообщения представьте в виде двух последовательностей тритов, разделённых запятыми. Пример записи решения (неверного):

$$X = (-1, 1, 0, 1, 1, 0), \quad X' = (-1, -1, 1, 0, 1, 1, -1, 0).$$

ОПИСАНИЕ Curl27. Функция Curl27 ставит в соответствие троичному слову X произвольной длины хэш-значение из T^{243} . При хэшировании применяется вспомогательная sponge-функция Curl27-f: $T^{729} \rightarrow T^{729}$. Алгоритм хэширования следующий.

1) Расширить X нулями до слова, длина которого кратна 243. Разбить полученное слово на блоки $X_1, X_2, \dots, X_d \in T^{243}$.

2) Подготовить слово-состояние $W = W_0 W_1 W_2 \in T^{729}$, $W_i \in T^{243}$, $i = 1, 2, 3$. Инициализировать слово-состояние, заполняя слова W_0 и W_2 нулями и записывая в W_1 длину входного слова X (до расширения).

Длина представляется троичным словом с порядком тритов от младшего к старшему. Например, длина $25 = 3^3 - 3^1 + 1$ задаётся словом $\underbrace{1\bar{1}01000 \dots 0}_{243}$. Здесь $\bar{1}$ означает -1 .

- 3) Для $i = 1, 2, \dots, d$ выполнить: $W_0 \leftarrow X_i, W \leftarrow \text{Curl27-f}(W)$.
- 4) Возвратить W_0 .

ОПИСАНИЕ Curl27-f. В Curl27-f используется S -блок

$$S: \mathbb{T}^3 \rightarrow \mathbb{T}^3, \quad (a, b, c) \mapsto (F(a, b, c), F(b, c, a), F(c, a, b)),$$

$$F(a, b, c) = a^2 b^2 c + a^2 b c^2 - a b^2 c^2 + a^2 b^2 - a^2 b c + a^2 c^2 + a b^2 c - a^2 c + a b^2 - a c^2 + b^2 c + b c^2 - a^2 - b^2 + b c - c^2 - c + 1,$$

где вычисления ведутся по модулю 3 и 2 представляется тритом -1 .

По преобразованию состояния W выполняется 27 итераций. Итерация состоит из 6 шагов. На каждом шаге группируются определённые тройки тритов W . Затем каждая тройка (a, b, c) заменяется на $S(a, b, c)$.

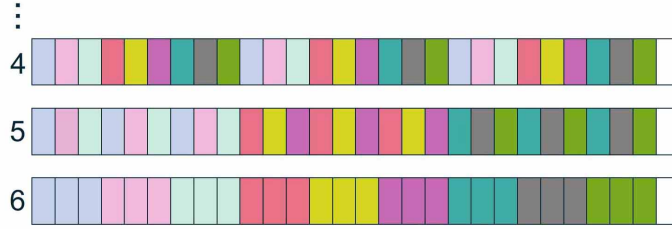


Рис. 7. Группировки (3 последних шага, сгруппированные триты окрашены одинаково)

Группировки организованы следующим образом (рис. 7). На первом шаге состояние разбивается на 3 слова по 243 трита. Группируются символы этих слов с одинаковыми номерами. На втором шаге состояние разбивается на 9 слов по 81 триту. Группируются символы 1, 2 и 3-го слов с одинаковыми номерами, затем символы 4, 5 и 6-го слов, и т. д. После этого выполняется разбиение на слова длины 27, затем длины 9, затем длины 3 с сохранением логики группировки. На последнем, шестом, шаге группируются последовательные тройки тритов.

Бонусная задача (доп. баллы, спец. приз). Найдите коллизию при изменении правил формирования начального слова-состояния W : теперь при инициализации $W_0 = W_2 = (01\bar{1})^{81} \in \mathbb{T}^{243}$.

РЕШЕНИЕ. Для слова $u \in \mathbb{T}^n$ через u^m обозначим m повторений слова u . Пусть $u = u_0 u_1 \dots u_{n-1}$ и $u^{[m]} = u_0^m u_1^m \dots u_{n-1}^m$. Слово вида $u^{[m]}$ назовём m -фрагментированным.

Теорема 1. Пусть m — степень числа 3, $m \leq 729$. Тогда sponge-функция Curl27-f сохраняет m -фрагментацию.

ДОКАЗАТЕЛЬСТВО. На i -м шаге раундовой функции Curl27-f состояние W делится на слова длины $n = 3^{6-i}$, $i = 1, 2, \dots, 6$.

Для $n \leq m$ шаговое преобразование сохраняет равенство тритов внутри фрагментов. Это следует из того, что $S(a, a, a) = (b, b, b)$.

Для $n > m$ равенство также сохраняется, поскольку в каждом фрагменте триты на различных позициях обрабатываются одинаково. Теорема 1 доказана.

Пусть m — небольшая степень числа 3 (интересны случаи $m = 3, 9, 27$). Рассмотрим троичную строку X длины

$$1 + 3 + 3^2 + \dots + 3^{m-1} = (3^m - 1)/2.$$

Длина задаётся словом из m единиц. Следовательно, начальное состояние Curl27 при обработке X будет m -фрагментированным (один фрагмент из единиц, остальные фрагменты нулевые).

Будем выбирать триты X так, чтобы сохранять m -фрагментацию состояния при хэшировании. Это легко сделать с учётом теоремы 1: все полные m -фрагменты X должны иметь вид α^m , $\alpha \in \mathbb{T}$, а триты последнего (неполного) фрагмента должны быть нулевыми, чтобы соответствовать дополненным тритам. Добившись m -фрагментации состояния, получаем m -фрагментацию хэш-значения. Фактически, хэш-значение задаётся $243/m$ тритами, каждый из которых повторяется m раз.

Найти коллизию для Curl27 можно после обработки порядка $\sqrt{3^{243/m}}$ строк X описанной структуры, т. е. за время порядка

$$3^m \cdot \sqrt{3^{243/m}} = 3^{m+121,5/m}.$$

Минимум функции, описанной выше, достигается при $m = 9$. Атака при $m = 9$ требует обработки приблизительно $\sqrt{3^{13,5}}$ строк по $9841 = 243 \cdot 40 + 121$ тритов каждая.

Джереми Джин (Франция, Национальное агентство кибербезопасности Франции) нашёл коллизию

$$\begin{aligned} X &= 0^{243-39}(101100110101111100101100000)^{[9]}0^{121}, \\ X' &= 0^{243-39}(000011110100111111001000000)^{[9]}0^{121} \end{aligned}$$

и стал единственным участником, решившим эту задачу.

Фрагментация является инвариантом Curl27-f. Это позволяет понизить размерность и быстро решить основную задачу. Для решения бонусной задачи Дж. Джин предложил использовать другой инвариант для Curl27-f: назовём *3-расширенным* слово вида $(abc)^{81} \in \mathbb{T}^{243}$. Тогда

если каждая часть W_0, W_1, W_2 состояния W является 3-расширенной, то и слово $\text{Curl27-f}(W)$ обладает тем же свойством.

В начальном состоянии W_0 и W_2 действительно 3-расширенные. В соответствии с инвариантом слово W_1 , представляющее длину хэшируемого сообщения X , должно иметь вид $(ab1)^{81}$, $(a10)^{81}$ или $(100)^{81}$ (длина ненулевая и положительная). В результате, X содержит по крайней мере $1 + 27 + \dots + 27^{80} > 3^{240}$ тритов.

При хэшировании легко сохранять инвариант: полные 243-фрагменты из X должны быть 3-расширенными, а последний неполный фрагмент (если он существует) — заполнен нулями. Результирующие хэш-значения 3-расширенные, и для них существует всего 27 вариантов, так что коллизия гарантированно найдётся после обработки 28 строк X .

На самом деле атака практически не реализуема: время порядка 3^{240} , требуемое только для записи сообщений, недопустимо большое даже по сравнению со временем $3^{243/2}$ классической атаки дней рождения.

2.15. Задача «8-битный S-блок».

ФОРМУЛИРОВКА. Перестановку S множества $\{0, 1\}^n$, или \mathbb{F}_2^n , обычно называют n -битным S-блоком. Будем акцентировать внимание на следующих криптографических свойствах S-блоков.

1) *Алгебраическая степень* (минимальная) $\deg(S)$ — минимальная алгебраическая степень среди компонентных функций перестановки S .

2) *Нелинейность* $\text{nl}(S)$ — расстояние Хэмминга между множеством всех компонентных функций S и множеством всех аффинных функций.

3) *Дифференциальная равномерность* $\text{du}(S)$ — максимальное число решений уравнения $S(x) \oplus S(x \oplus \alpha) = \beta$ при $\alpha, \beta \in \mathbb{F}_2^n$, $\alpha \neq 0$.

4) *Алгебраическая иммунность* (графовая) $\text{ai}(S)$ — минимальная алгебраическая степень булевых функций $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ таких, что $f \not\equiv 0$ и $f(x, y) = 0$ для любого $x \in \mathbb{F}_2^n$ и $y = S(x)$.

В современной симметричной криптографии большое распространение получили S-блоки размерности $n = 8$. Например, такой S-блок используется в блочном шифре AES. Характеристики S_{AES} следующие:

$$(\deg, \text{nl}, \text{du}, \text{ai})(S_{\text{AES}}) = (7, 112, 4, 2).$$

Равенство $\text{ai}(S_{\text{AES}}) = 2$ означает, что S_{AES} (и весь AES) может быть компактно описан с помощью квадратичных уравнений. Это может быть слабостью в контексте алгебраических атак.

Задача на спец. приз. При ограничениях $(\deg, \text{ai})(S) = (7, 3)$ (оптимальные значения) максимизируйте $\text{nl}(S)$ и/или минимизируйте $\text{du}(S)$. Текущий рекорд [17, 18] имеет вид $(\deg, \text{nl}, \text{du}, \text{ai})(S) = (7, 108, 6, 3)$.

Замечание. Напомним необходимые определения.

- Булева функция $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ однозначно представляется в алгебраической нормальной форме (АНФ): $f(x) = \bigoplus_{I \in \mathcal{P}(N)} a_I \prod_{i \in I} x_i$, где $\mathcal{P}(N)$ — множество всех подмножеств множества $N = \{1, \dots, n\}$, $a_I \in \mathbb{F}_2$.

- Алгебраическая степень f — степень её АНФ:

$$\deg(f) = \max\{|I| \mid a_I \neq 0, I \in \mathcal{P}(N)\}.$$

- Булева функция степени не выше 1 называется аффинной.
- Расстояние Хэмминга равно $d(f, g) = |\{x \in \mathbb{F}_2^n \mid f(x) \neq g(x)\}|$.
- Функцию $S: \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ можно записать в виде $S = (s_1, \dots, s_n)$, где s_i — булева функция, $i = 1, \dots, n$. Нетривиальная линейная комбинация функций s_1, \dots, s_n называется компонентной функцией S .

РЕШЕНИЕ. К сожалению, ценных идей по этой задаче не поступило. Для заданного набора криптографических свойств проблема остаётся открытой. Существует несколько конструкций, основанных на так называемой butterfly-структуре, которые дают текущий рекорд $(7, 108, 6, 3)$ (см. [17, 18]). Если кандидаты на улучшение существуют, по-видимому их надо искать за пределами известных структур и конструкций криптографических перестановок.

2.16. Задача «Гипотеза».

ФОРМУЛИРОВКА. Пусть \mathbb{F}_2 — поле из двух элементов, $n \geq 3$ — целое число и $f(X)$ — неприводимый многочлен над \mathbb{F}_2 степени n . Известно, что множество классов эквивалентности многочленов над \mathbb{F}_2 по модулю $f(X)$ образует конечное поле порядка 2^n . Обозначим его через \mathbb{F}_{2^n} . Различные неприводимые многочлены порождают изоморфные поля, которые обладают одинаковыми алгебраическими свойствами.

Задача на спец. приз. Докажите или опровергните гипотезу ниже.

Гипотеза 1. Пусть k и n взаимно просты, $F(\beta) = \beta^{4^k - 2^k + 1}$, $\beta \in \mathbb{F}_{2^n}$, и $\Delta = \{F(\beta) + F(\beta + 1) + 1 \mid \beta \in \mathbb{F}_{2^n}\}$. Тогда для любых различных v_1, v_2 из \mathbb{F}_{2^n} выполнено $|\{(x, y, z) \in \Delta^3 \mid v_1x + v_2y + (v_1 + v_2)z = 0\}| = 2^{2n-3}$.

Пример для $n = 3$. Возьмём $f(X) = X^3 + X + 1$. Каждый элемент $\beta \in \mathbb{F}_{2^3}$ может быть записан в виде многочлена $a_0 + a_1X + a_2X^2 \in \mathbb{F}_2[X]$. Многочлены 0 и 1 играют роль нуля и единицы поля. Выполняя операции в построенном поле \mathbb{F}_{2^3} можно проверить требуемое равенство.

РЕШЕНИЕ. Эта задача, впервые описанная в [19] и рассмотренная также в [20], представляет собой открытую проблему. Гипотеза проверена для $n \leq 9$ и $n = 11$. Участники олимпиады предложили некоторые идеи.

К сожалению, ни одна из них не даёт значительного вклада в доказательство гипотезы или поиск контрпримера. Команда Кристины Геут, Сергея Титова и Дмитрия Ананичева (Россия, УрГУПС, УрФУ) и команда Алексея Зеленецкого, Михаила Кудинова и Дениса Набокова (Россия, МГТУ им. Н. Э. Баумана) доказали гипотезу для частного случая $k = 1$. Однако этот случай непоказательный, поскольку тогда функция квадратичная, а проверяемое свойство хорошо известно для квадратичных функций. При этом доказательство нельзя расширить на общий случай.

3. Победители олимпиады

По итогам олимпиады были награждены призами и почётными дипломами 42 участника в первом раунде и 21 команда во втором раунде из 26 стран мира. Ниже представлена информация о победителях Олимпиады NSUCRYPTO'2019. Полную информацию о призёрах можно найти на официальном сайте [21].

3.1. Победители первого тура, секция А («Школьники»).

1 место. *Борислав Кирилов*, 16 баллов (Первая частная математическая гимназия, София, Болгария).

1 место. *Алексей Львов*, 16 баллов (Гимназия № 6, Новосибирск, Россия).

2 место. *Ленарт Букар*, 15 баллов (Бежиградская гимназия, Любляна, Словения).

3 место. *Варвара Лебединская*, 14 баллов (СУНЦ НГУ, Новосибирск, Россия).

3 место. *Габриель Эрикссон*, 14 баллов (Tullangsskolan, Эребру, Швеция).

3.2. Победители первого тура, секция В («Студенты»).

1 место. *Максим Плюшкин*, 22 балла (МГУ им. М. В. Ломоносова, Москва, Россия).

1 место. *Михаил Кудинов*, 21 балл (МГТУ им. Н. Э. Баумана, Москва, Россия).

2 место. *Нарендра Пател*, 19 баллов (Индийский институт технологий, Рурки, Индия).

2 место. *Владимир Щавелев*, 19 баллов (СПбГУ, Санкт-Петербург, Россия).

3 место. *Тхань Нгуен Ван*, 16 баллов (Технологический университет Хошимина, Хошимин, Вьетнам).

3 место. *Роман Гибадулин*, 16 баллов (ЯрГУ им. П. Г. Демидова, Ярославль, Россия).

3 место. *Дарья Гребенчук*, 16 баллов (ЯрГУ им. П. Г. Демидова, Ярославль, Россия).

3 место. *Тюонг Нгуен*, 15 баллов (Технологический университет Хошимина, Хошимин, Вьетнам).

3.3. Победители первого тура, секция В («Профессионалы»).

1 место. *Хеннинг Зайдлер*, 26 баллов (Берлинский технический университет, Берлин, Германия).

2 место. *Мадалина Болбочану*, 20 баллов (Bitdefender, Бухарест, Румыния).

2 место. *Самуэль Танг*, 20 баллов (Black Bauhinia, Гонконг, КНР).

3 место. *Ирина Слопкина*, 16 баллов (НИЯУ МИФИ, Москва, Россия).

3.4. Победители второго тура («Студенты»).

1 место. *Алексей Зеленецкий, Михаил Кудинов, Денис Набоков*, 51 балл (МГТУ им. Н. Э. Баумана, Москва, Россия).

2 место. *Нгок Ки Нгуен, Дунг Чыонг, Фуок Нгуен*, 43 балла (Технологический университет Хошимина, Хошимин, Вьетнам; Высшая нормальная школа, Париж, Франция).

2 место. *Тхань Нгуен Ван, Куок Бао Нгуен, Нган Нгуен*, 40 баллов (Технологический университет Хошимина, Хошимин, Вьетнам).

3 место. *Максим Плюшкин*, 34 балла (МГУ им. М. В. Ломоносова, Москва, Россия).

3 место. *Илья Трусевич, Максим Бибик, Александр Шульга*, 31 балл (Белорусский гос. университет, Минск, Беларусь).

3.5. Победители второго тура («Профессионалы»).

1 место. *Ирина Слопкина, Михаил Сорокин, Владимир Бобров*, 48 баллов (МГТУ им. Н. Э. Баумана, Москва, Россия).

1 место. *Кристина Геут, Сергей Титов, Дмитрий Ананичев*, 46 баллов (УрГУПС, УрФУ, Екатеринбург, Россия).

2 место. *Хеннинг Зайдлер, Катя Штумпп*, 42 балла (Берлинский технический университет, Берлин, Германия).

3 место. *Виктория Власова, Михаил Поляков, Алексей Чиликов*, 37 баллов (МГТУ им. Н. Э. Баумана, Москва, Россия).

3 место. *Дюк Три Нгуен, Куан Доан, Тюонг Нгуен*, 36 баллов (Cryptographic Engineering Research Group, pwnphofun, Технологический университет Хошимина, Хошимин, Вьетнам).

3 место. *Мадалина Болбочану, Андрей Мозаге, Раду Титиу*, 34 балла (Bitdefender, Ясский университет им. А. И. Кузы, Бухарест, Румыния).

Спец. приз. *Джеремии Джин*, 20 баллов (Национальное агентство кибербезопасности Франции, Париж, Франция).

ЛИТЕРАТУРА

1. The official website of NSUCRYPTO. Novosibirsk: Novosibirsk State Univ., 2020. Available at nsucrypto.nsu.ru (accessed Sept. 24, 2020).
2. Unsolved problems of NSUCRYPTO. Available at nsucrypto.nsu.ru/unsolved-problems (accessed Sept. 24, 2020).
3. Геут К. Л., Кириенко К. А., Садков П. О., Таскин Р. И., Титов С. С. О явных конструкциях для решения задачи “A secret sharing” // Прикл. дискрет. математика. Прил. 2017. № 10. С. 68–70.
4. Agievich S. V., Gorodilova A. A., Idrisova V. A., Kolomeec N. A., Shushuev G. I., Tokareva N. N. Mathematical problems of the Second International Students’ Olympiad in Cryptography // Cryptologia. 2017. Vol. 41, No. 6. P. 534–565.
5. Agievich S. V., Gorodilova A. A., Kolomeec N. A., Nikova S., Preneel B., Rijmen V., Shushuev G. I., Tokareva N. N., Vitkup V. A. Problems, solutions and experience of the First International Students’ Olympiad in Cryptography // Прикл. дискрет. математика. 2015. № 3. С. 41–62.
6. Gorodilova A. A., Agievich S. V., Carlet C., Gorkunov E. V., Idrisova V. A., Kolomeec N. A., Kutsenko A. V., Nikova S., Oblaukhov A. K., Picek S., Preneel B., Rijmen V., Tokareva N. N. Problems and solutions from the Fourth International Students’ Olympiad in Cryptography (NSUCRYPTO) // Cryptologia. 2019. Vol. 43, No. 2. P. 138–174.
7. Gorodilova A. A., Agievich S. V., Carlet C., Hou X., Idrisova V. A., Kolomeec N. A., Kutsenko A. V., Mariot L., Oblaukhov A. K., Picek S., Preneel B., Rosie R., Tokareva N. N. The Fifth International Students’ Olympiad in Cryptography – NSUCRYPTO: Problems and their solutions // Cryptologia. 2020. Vol. 44, No. 3. P. 223–256.
8. Tokareva N. N., Gorodilova A. A., Agievich S. V., Idrisova V. A., Kolomeec N. A., Kutsenko A. V., Oblaukhov A. K., Shushuev G. I. Mathematical methods in solutions of the problems presented at the Third International Students’ Olympiad in Cryptography // Прикл. дискрет. математика. 2018. № 40. С. 34–58.
9. Schneier B. Applied cryptography: Protocols, algorithms and source code in C. Hoboken, NJ: Wiley, 1996.
10. Lewand R. E. Cryptological mathematics. Washington, DC: MAA Press, 2000.
11. Letter frequency // Wikipedia. San Francisco: Wikimedia Foundation, 2020. Available at en.wikipedia.org/wiki/Letter_frequency (accessed Sept. 24, 2020).
12. Find words using pattern matching // Litscape.com. Calgary: The Bitmill, 2018. Available at www.litscape.com/word_tools/pattern_match.php (accessed Sept. 24, 2020).
13. Brinkmann M., Leander G. On the classification of APN functions up to dimension five // Des. Codes Cryptogr. 2008. Vol. 49, No. 1–3. P. 273–288.

14. **De Cannière C.** Analysis and design of symmetric encryption algorithms: PhD thesis. Katholieke Univ. Leuven, Heverlee, 2007.
15. Test server for the problem **TwinPeaks3**. Available at nsucrypto.nsu.ru/archive/2019/round/2/task/4 (accessed Sept. 24, 2020).
16. An implementation for the function **Cur127** in Java. Available at nsucrypto.nsu.ru/media/Olympiads/2019/Round_2/Tasks/cur127.java (accessed Sept. 24, 2020).
17. **De la Cruz Jiménez R. A.** Generation of 8-bit S-boxes having almost optimal cryptographic properties using smaller 4-bit S-boxes and finite field multiplication // Progress in Cryptology – LATINCRYPT 2017. Rev. Sel. Pap. 5th Int. Conf. Cryptol. Inform. Secur. Latin America (Havana, Cuba, Sept. 20–22, 2017). Cham: Springer, 2019. P. 191–206. (Lect. Notes Comput. Sci.; Vol. 11368).
18. **Fomin D. B.** New classes of 8-bit permutations based on a butterfly structure // Мат. вопросы криптографии. 2019. Т. 10, вып. 2. С. 169–180.
19. **Carlet C.** Componentwise APNness, Walsh uniformity of APN functions, and cyclic-additive difference sets // Finite Fields Appl. 2018. Vol. 53. P. 226–253.
20. **Carlet C.** On APN exponents, characterizations of differentially uniform functions by the Walsh transform, and related cyclic-difference-set-like structures // Des. Codes Cryptogr. 2019. Vol. 87, No. 2–3. P. 203–224.
21. Total results of NSUCRYPTO'2019. Available at nsucrypto.nsu.ru/archive/2019/total_results/#data (accessed Sept. 24, 2020).

Городилова Анастасия Александровна
 Токарева Наталья Николаевна
 Агиевич Сергей Валерьевич
 Карле Клод
 Горкунов Евгений Владимирович
 Идрисова Валерия Александровна
 Коломеец Николай Александрович
 Куценко Александр Владимирович
 Лебедев Роман Константинович
 Никова Светла
 Облаухов Алексей Константинович
 Панкратова Ирина Анатольевна
 Пудовкина Марина Александровна
 Реймен Винсент
 Удовенко Алексей Николаевич

Статья поступила
 20 мая 2020 г.
 После доработки —
 18 августа 2020 г.
 Принята к публикации
 21 августа 2020 г.

ON THE SIXTH INTERNATIONAL OLYMPIAD
IN CRYPTOGRAPHY NSUCRYPTO

*A. A. Gorodilova^{1, a}, N. N. Tokareva^{1, 2}, S. V. Agievich³, C. Carlet⁴,
E. V. Gorkunov^{1, 5}, V. A. Idrisova¹, N. A. Kolomeec¹,
A. V. Kutsenko^{1, 5}, R. K. Lebedev⁵, S. Nikova⁶, A. K. Oblaukhov¹,
I. A. Pankratova⁷, M. A. Pudovkina⁸,
V. Rijmen⁶, and A. N. Udovenko⁹*

¹ Sobolev Institute of Mathematics,

4 Akad. Koptyug Avenue, 630090 Novosibirsk, Russia

² Laboratory of Cryptography JetBrains Research,

1 Pirogov Street, 630090 Novosibirsk, Russia

³ Belarusian State University,

4 Nezavisimost Avenue, 220030 Minsk, Belarus

⁴ University of Paris 8,

2 Rue de la Liberte, 93200 Saint-Denis, France

⁵ Novosibirsk State University,

2 Pirogov Street, 630090 Novosibirsk, Russia

⁶ ESAT-COSIC, KU Leuven,

10 Kasteelpark Arenberg, B-3001 Leuven, Belgium

⁷ Tomsk State University,

36 Lenin Avenue, 634050 Tomsk, Russia

⁸ Bauman Moscow State Technical University,

5/1 Vtoraya Baumanskaya Street, 105005 Moscow, Russia

⁹ SnT, University of Luxembourg,

2 Avenue de l'Universite, L-4365 Esch-sur-Alzette, Luxembourg

E-mail: ^ansucrypto@nsu.ru

The work of the first, second, and sixth authors is supported by Mathematical Center in Akademgorodok (Agreement 075–15–2019–1613 with the Ministry of Science and Higher Education of the Russian Federation) and Laboratory of Cryptography JetBrains Research; the work of the fifth author is carried out under the state contract of the Sobolev Institute of Mathematics (Project 0314–2019–0016); the work of the seventh, eighth, and eleventh authors is supported by the Russian Foundation for Basic Research (Projects 20–31–70043, 18–07–01394, 19–31–90093).

English version: Journal of Applied and Industrial Mathematics **14** (4), 623–647 (2020), DOI 10.1134/S1990478920040031.

Abstract. We present problems of the Sixth International Olympiad in cryptography NSUCRYPTO'2019 along with their solutions. The problems are related to attacks on ciphers and hash functions, protocols, Boolean functions, Dickson polynomials, prime numbers, rotor machines, etc. We discuss several open problems on mathematical countermeasures to side-channel attacks, APN involutions, S-boxes, etc. The problem of finding a collision for the hash function `Cur127` was partially solved during the Olympiad. Tab. 11, illustr. 7, bibliogr. 21.

Keywords: cryptography, cipher, hash function, Hamming code, slide attack, threshold implementation, Dickson polynomial, APN function, olympiad, NSUCRYPTO.

REFERENCES

1. The official website of NSUCRYPTO (Novosibirsk State Univ., Novosibirsk, 2020). Available at nsucrypto.nsu.ru (accessed Sept. 24, 2020).
2. Unsolved problems of NSUCRYPTO. Available at nsucrypto.nsu.ru/unsolved-problems (accessed Sept. 24, 2020).
3. K. L. Geut, K. A. Kirienko, P. O. Sadkov, R. I. Taskin, and S. S. Titov, On explicit constructions for solving the problem “A secret sharing”, *Prikl. Diskretn. Mat., Prilozh.*, No. 10, 68–70 (2017) [Russian].
4. S. V. Agievich, A. A. Gorodilova, V. A. Idrisova, N. A. Kolomeec, G. I. Shushuev, and N. N. Tokareva, Mathematical problems of the Second International Students' Olympiad in Cryptography, *Cryptologia* **41** (6), 534–565 (2017).
5. S. V. Agievich, A. A. Gorodilova, N. A. Kolomeec, S. Nikova, B. Preneel, V. Rijmen, G. I. Shushuev, N. N. Tokareva, and V. A. Vitkup, Problems, solutions and experience of the First International Students' Olympiad in Cryptography, *Prikl. Diskretn. Mat.*, No 3, 41–62 (2015).
6. A. A. Gorodilova, S. V. Agievich, C. Carlet, E. V. Gorkunov, V. A. Idrisova, N. A. Kolomeec, A. V. Kutsenko, S. Nikova, A. K. Oblaukhov, S. Picek, B. Preneel, V. Rijmen, and N. N. Tokareva, Problems and solutions from the Fourth International Students' Olympiad in Cryptography (NSUCRYPTO), *Cryptologia* **43** (2), 138–174 (2019).
7. A. A. Gorodilova, S. V. Agievich, C. Carlet, X. Hou, V. A. Idrisova, N. A. Kolomeec, A. V. Kutsenko, L. Mariot, A. K. Oblaukhov, S. Picek, B. Preneel, R. Rosie, and N. N. Tokareva, The Fifth International Students' Olympiad in Cryptography – NSUCRYPTO: Problems and their solutions, *Cryptologia* **44** (3), 223–256 (2020).
8. N. N. Tokareva, A. A. Gorodilova, S. V. Agievich, V. A. Idrisova, N. A. Kolomeec, A. V. Kutsenko, A. K. Oblaukhov, and G. I. Shushuev, Mathematical methods in solutions of the problems presented at the Third International Students' Olympiad in Cryptography, *Prikl. Diskretn. Mat.*, No. 40, 34–58 (2018).

9. **B. Schneier**, *Applied cryptography: Protocols, algorithms and source code in C* (Wiley, Hoboken, NJ, 1996).
10. **R. E. Lewand**, *Cryptological mathematics* (MAA Press, Washington, DC, 2000).
11. Letter frequency, in *Wikipedia* (Wikimedia Foundation, San Francisco, 2020). Available at en.wikipedia.org/wiki/Letter_frequency (accessed Sept. 24, 2020).
12. Find words using pattern matching, in *Litscape.com* (The Bitmill, Calgary, 2018). Available at www.litscape.com/word_tools/pattern_match.php (accessed Sept. 24, 2020).
13. **M. Brinkmann** and **G. Leander**, On the classification of APN functions up to dimension five, *Des. Codes Cryptogr.* **49** (1–3), 273–288 (2008).
14. **C. De Cannière**, Analysis and design of symmetric encryption algorithms, *PhD thesis* (Katholieke Univ. Leuven, Heverlee, 2007).
15. Test server for the problem TwinPeaks3. Available at nsucrypto.nsu.ru/archive/2019/round/2/task/4 (accessed Sept. 24, 2020).
16. An implementation for the function Curl127 in Java. Available at nsucrypto.nsu.ru/media/Olympiads/2019/Round_2/Tasks/curl127.java (accessed Aug. 24, 2020).
17. **R. A. De la Cruz Jiménez**, Generation of 8-bit S-boxes having almost optimal cryptographic properties using smaller 4-bit S-boxes and finite field multiplication, in *Progress in Cryptology – LATINCRYPT 2017* (Rev. Sel. Pap. 5th Int. Conf. Cryptol. Inform. Secur. Latin America, Havana, Cuba, Sept. 20–22, 2017) (Springer, Cham, 2019), pp. 191–206 (Lect. Notes Comput. Sci., Vol. 11368).
18. **D. B. Fomin**, New classes of 8-bit permutations based on a butterfly structure, *Mat. Vopr. Kriptogr.* **10** (2), 169–180 (2019).
19. **C. Carlet**, Componentwise APNness, Walsh uniformity of APN functions, and cyclic-additive difference sets, *Finite Fields Appl.* **53**, 226–253 (2018).
20. **C. Carlet**, On APN exponents, characterizations of differentially uniform functions by the Walsh transform, and related cyclic-difference-set-like structures, *Des. Codes Cryptogr.* **87** (2–3), 203–224 (2019).
21. Total results of NSUCRYPTO'2019. Available at nsucrypto.nsu.ru/archive/2019/total_results/#data (accessed Sept. 24, 2020).

Anastasiya A. Gorodilova
 Natalia N. Tokareva
 Sergey V. Agievich
 Claude Carlet
 Evgeny V. Gorkunov
 Valeria A. Idrisova
 Nikolay A. Kolomeec
 Aleksandr V. Kutsenko

Roman K. Lebedev
 Svetla Nikova
 Aleksey K. Oblaukhov
 Irina A. Pankratova
 Marina A. Pudovkina
 Vincent Rijmen
 Aleksey N. Udovenko

Received May 20, 2020

Revised August 18, 2020

Accepted August 21, 2020