

ISSN 2409-546X

# ЮНЫЙ УЧЁНЫЙ

МЕЖДУНАРОДНЫЙ НАУЧНЫЙ ЖУРНАЛ



6+

2  
Часть 1  
2021

# Юный ученый

## Международный научный журнал

№ 2 (43) / 2021

Издается с февраля 2015 г.

*Главный редактор:* Ахметов Ильдар Геннадьевич, кандидат технических наук

*Редакционная коллегия:*

Жураев Хусниддин Олтинбоевич, доктор педагогических наук (Узбекистан)

Иванова Юлия Валентиновна, доктор философских наук

Каленский Александр Васильевич, доктор физико-математических наук

Кошербаева Айгерим Нуралиевна, доктор педагогических наук, профессор (Казахстан)

Куташов Вячеслав Анатольевич, доктор медицинских наук

Лактионов Константин Станиславович, доктор биологических наук

Сараева Надежда Михайловна, доктор психологических наук

Абдрасилов Турганбай Курманбаевич, доктор философии (PhD) по философским наукам (Казахстан)

Авдеюк Оксана Алексеевна, кандидат технических наук

Айдаров Оразхан Турсункожаевич, кандидат географических наук (Казахстан)

Алиева Тарана Ибрагим кызы, кандидат химических наук (Азербайджан)

Ахметова Валерия Валерьевна, кандидат медицинских наук

Бердиев Эргаш Абдуллаевич, кандидат медицинских наук (Узбекистан)

Брезгин Вячеслав Сергеевич, кандидат экономических наук

Данилов Олег Евгеньевич, кандидат педагогических наук

Дёмин Александр Викторович, кандидат биологических наук

Дядюна Кристина Владимировна, кандидат юридических наук

Желнова Кристина Владимировна, кандидат экономических наук

Жуйкова Тамара Павловна, кандидат педагогических наук

Игнатова Мария Александровна, кандидат искусствоведения

Искаков Руслан Маратбекович, кандидат технических наук (Казахстан)

Кайгородов Иван Борисович, кандидат физико-математических наук (Бразилия)

Калдыбай Кайнар Калдыбайулы, доктор философии (PhD) по философским наукам (Казахстан)

Кенесов Асхат Алмасович, кандидат политических наук

Коварда Владимир Васильевич, кандидат физико-математических наук

Комогорцев Максим Геннадьевич, кандидат технических наук

Котляров Алексей Васильевич, кандидат геолого-минералогических наук

Кузьмина Виолетта Михайловна, кандидат исторических наук, кандидат психологических наук

Курпаяниди Константин Иванович, доктор философии (PhD) по экономическим наукам (Узбекистан)

Кучерявенко Светлана Алексеевна, кандидат экономических наук

Лескова Екатерина Викторовна, кандидат физико-математических наук

Макеева Ирина Александровна, кандидат педагогических наук

Матвиенко Евгений Владимирович, кандидат биологических наук

Матроскина Татьяна Викторовна, кандидат экономических наук

Матусевич Марина Степановна, кандидат педагогических наук

Мусаева Ума Алиевна, кандидат технических наук

Насимов Мурат Орленбаевич, кандидат политических наук (Казахстан)

Паридинова Ботагоз Жаппаровна, магистр философии (Казахстан)

Прончев Геннадий Борисович, кандидат физико-математических наук

Рахмонов Азиз Боситович, доктор философии (PhD) по педагогическим наукам (Узбекистан)

Семахин Андрей Михайлович, кандидат технических наук

Сенцов Аркадий Эдуардович, кандидат политических наук

Сенюшкин Николай Сергеевич, кандидат технических наук

Султанова Дилшоода Намозовна, доктор архитектуры (Узбекистан)

Титова Елена Ивановна, кандидат педагогических наук

Ткаченко Ирина Георгиевна, кандидат филологических наук

Федорова Мария Сергеевна, кандидат архитектуры

Фозилов Садриддин Файзуллаевич, кандидат химических наук (Узбекистан)

Яхина Асия Сергеевна, кандидат технических наук

Ячинова Светлана Николаевна, кандидат педагогических наук

## **Международный редакционный совет:**

Айрян Заруи Геворковна, кандидат филологических наук, доцент (Армения)  
Арошидзе Паата Леонидович, доктор экономических наук, ассоциированный профессор (Грузия)  
Атаев Загир Вагитович, кандидат географических наук, профессор (Россия)  
Ахмеденов Кажмурат Максutowич, кандидат географических наук, ассоциированный профессор (Казахстан)  
Бидова Бэла Бертовна, доктор юридических наук, доцент (Россия)  
Борисов Вячеслав Викторович, доктор педагогических наук, профессор (Украина)  
Буриев Хасан Чутбаевич, доктор биологических наук, профессор (Узбекистан)  
Велковска Гена Цветкова, доктор экономических наук, доцент (Болгария)  
Гайич Тамара, доктор экономических наук (Сербия)  
Данатаров Агахан, кандидат технических наук (Туркменистан)  
Данилов Александр Максимович, доктор технических наук, профессор (Россия)  
Демидов Алексей Александрович, доктор медицинских наук, профессор (Россия)  
Досманбетова Зейнегуль Рамазановна, доктор философии (PhD) по филологическим наукам (Казахстан)  
Ешиев Абдыракман Молдоалиевич, доктор медицинских наук, доцент, зав. отделением (Кыргызстан)  
Жолдошев Сапарбай Тезекбаевич, доктор медицинских наук, профессор (Кыргызстан)  
Игисинов Нурбек Сагинбекович, доктор медицинских наук, профессор (Казахстан)  
Кадыров Кутлуг-Бек Бекмурадович, кандидат педагогических наук, декан (Узбекистан)  
Кайгородов Иван Борисович, кандидат физико-математических наук (Бразилия)  
Каленский Александр Васильевич, доктор физико-математических наук, профессор (Россия)  
Козырева Ольга Анатольевна, кандидат педагогических наук, доцент (Россия)  
Колпак Евгений Петрович, доктор физико-математических наук, профессор (Россия)  
Кощербаяева Айгерим Нуралиевна, доктор педагогических наук, профессор (Казахстан)  
Курпаяниди Константин Иванович, доктор философии (PhD) по экономическим наукам (Узбекистан)  
Куташов Вячеслав Анатольевич, доктор медицинских наук, профессор (Россия)  
Кыят Эмине Лейла, доктор экономических наук (Турция)  
Лю Цзюань, доктор филологических наук, профессор (Китай)  
Малес Людмила Владимировна, доктор социологических наук, доцент (Украина)  
Нагервадзе Марина Алиевна, доктор биологических наук, профессор (Грузия)  
Нурмамедли Фазиль Алигусейн оглы, кандидат геолого-минералогических наук (Азербайджан)  
Прокопьев Николай Яковлевич, доктор медицинских наук, профессор (Россия)  
Прокофьева Марина Анатольевна, кандидат педагогических наук, доцент (Казахстан)  
Рахматуллин Рафаэль Юсупович, доктор философских наук, профессор (Россия)  
Ребзов Максим Борисович, доктор сельскохозяйственных наук, профессор (Россия)  
Сорока Юлия Георгиевна, доктор социологических наук, доцент (Украина)  
Султанова Дилшода Намозовна, доктор архитектурных наук (Узбекистан)  
Узаков Гулом Норбоевич, доктор технических наук, доцент (Узбекистан)  
Федорова Мария Сергеевна, кандидат архитектуры (Россия)  
Хоналиев Назарали Хоналиевич, доктор экономических наук, старший научный сотрудник (Таджикистан)  
Хоссейни Амир, доктор филологических наук (Иран)  
Шарипов Аскар Калиевич, доктор экономических наук, доцент (Казахстан)  
Шуклина Зинаида Николаевна, доктор экономических наук (Россия)

# СОДЕРЖАНИЕ

## РУССКИЙ ЯЗЫК

*Калугин Т. А.*

Диалектные особенности речи жителей Южного Урала . . . . . 1

## ЛИТЕРАТУРА

*Комогорова А. С.*

«Радость красок» в стихотворениях И. А. Бунина . . . . . 3

*Моринец О. В.*

«Я была тогда с моим народом...» — поэтическое многоголосье в стихах о Великой Отечественной войне  
Е. Благиной . . . . . 6

*Пекарь С. А.*

Симеон Верхотурский как художественный образ в поэме Л. Кулешовой . . . . . 13

*Положенко Е. В.*

Воспитание и образование дворянских детей в первой половине XIX века на основе анализа  
произведений А. С. Пушкина . . . . . 16

*Санько П. А.*

Черты викторианского романа в американской литературе 2-й половины XIX века (на примере  
произведения Л. М. Олкотт «Маленькие женщины») . . . . . 19

## ИСТОРИЯ

*Князев М. А.*

Фриц Габер — гений злодейства . . . . . 22

*Сабуркин Д. А.*

Монеты как отражение истории . . . . . 23

## ОБЩЕСТВОЗНАНИЕ

*Ерохина А. И.*

Социальные сети как основа современной социальной структуры. . . . . 26

## МАТЕМАТИКА: АЛГЕБРА И НАЧАЛА АНАЛИЗА, ГЕОМЕТРИЯ

*Андрющенко Б. Е.*

Использование моделей многогранников для изучения возможностей реставрации и сохранения памятников  
архитектуры . . . . . 28

*Ким Де Хан*

Числа и их секреты . . . . . 31

*Толкачев В. А., Щербина В. В.*

Численный анализ прямых задач массопереноса, возникающих в результате применения биологического,  
химического и ядерного оружия . . . . . 33

## ИНФОРМАТИКА

*Андреев В. О.*

Робот «Газовый анализатор» с нейрофизиологической системой управления для проведения работ в ограниченных  
пространствах. . . . . 39

*Оловянишников А. Р.*

Разработка алгоритма и программного обеспечения для шифрования данных ..... 46

**ФИЗИКА**

*Шалаев А. Н.*

Альтернативные источники электроэнергии для дома ..... 52

# Разработка алгоритма и программного обеспечения для шифрования данных

Оловянишников Александр Романович, учащийся 11-го класса

Научный руководитель: Симаков Егор Евгеньевич, учитель математики, информатики и ИКТ  
МБОУ Лицей № 1 г. Южно-Сахалинска

Человечество за всю свою историю никогда не развивалось так быстро, как в новейшем времени. Если раньше компьютер могла купить только крупная компания, то сейчас его может позволить себе каждый. Вместе с компьютером в жизнь людей прочно вошел интернет, с помощью которого стало возможным передавать информацию по всему земному шару в считанные секунды. Однако такие возможности имеют свои минусы, и самым главным из них является проблема информационной безопасности.

**Ключевые слова:** шифрование, информационная безопасность, защита информации, алгоритм AOCRYPT.

## I. Решение проблемы защиты данных. Алгоритмы шифрования

Данные в сети могут быть украдены злоумышленниками и использованы в неблагоприятных целях. Некоторые вредоносные программы могут просто занимать место на диске и замедлять ОС, а другие перезаписывать MBR (загрузочная область диска) своим кодом и при перезапуске компьютера стирать данные с дисков (например, VineMEMZ). Во избежание запуска трояна — вредоносной программы, маскирующейся под любую другую, используется криптографическая функция SHA-256, MD5 и др. Результат функции является уникальным для каждого файла, поэтому, зная проверочную сумму оригинала, можно сразу понять о принадлежности данного файла к вредоносному ПО. Кроме того, многие криптографические алгоритмы были реализованы на компьютере, что позволило зашифровывать информацию с помощью сгенерированного ключа, который невозможно подобрать, ведь на это могут уйти десятки лет (например, RSA-2048).

Все алгоритмы шифрования делятся на две группы: симметричные и асимметричные. Для симметричных алгоритмов нужен один и тот же ключ для кодирования и декодирования информации, а для асимметричных — разные. Если алгоритм не поддается взлому в течение пяти лет — значит он может использоваться для защиты секретной информации. Ниже приведены одни из надежных и распространенных криптографических алгоритмов:

- **AES** в настоящее время является федеральным стандартом шифрования США. Используется вариант шифра с размером блока 128 бит.
- **Blowfish** — сложная схема выработки ключа существенно затрудняет атаку на алгоритм методом перебора, однако делает его непригодным для использования в системах, где ключ часто меняется, и на каждом ключе шифруются небольшие по объему данные. Алгоритм подходит для систем, в которых на одном и том же ключе шифруются большие массивы данных.
- **RSA** — алгоритм с открытым ключом, основывающийся на вычислительной сложности зада-

чи факторизации больших целых чисел. Стала первой системой, пригодной и для шифрования, и для цифровой подписи. Алгоритм используется в большом числе криптографических приложений, включая PGP, S/MIME, TLS/SSL, IPSEC/IKE и других.

Принцип работы всех криптографических алгоритмов одинаков: генерируется ключ, в соответствии с которым по установленному алгоритму изменяются данные и записываются в зашифрованный файл. Однако применение криптографических алгоритмов разное. Автором разработан собственный симметричный алгоритм (AOCRYPT) и программа-шифровальщик (AOCRYPT0R) на языке программирования C++.

Рассмотрим более подробно принципы работы двух наиболее популярных алгоритмов — BlowFish и RSA.

**Алгоритм шифрования BLOWFISH.** Blowfish — криптографический алгоритм, реализующий блочное симметричное шифрование с переменной длиной ключа. Разработан Брюсом Шнайером в 1993 году. Выполнен на простых и быстрых операциях: XOR, подстановка, сложение. Является незапатентованным и свободно распространяемым. По заявлению автора, критериями проектирования Blowfish были: скорость (шифрование на 32-битных процессорах происходит за 26 тактов); простота (за счёт использования простых операций, уменьшающих вероятность ошибки реализации алгоритма); компактность (возможность работать в менее, чем 5 Кбайт памяти); настраиваемая безопасность (изменяемая длина ключа).

Алгоритм разделён на этапы.

1. Инициализация массивов P и S при помощи секретного ключа K
  - Инициализация фиксированной строкой, состоящей из шестнадцатеричных цифр мантиссы числа пи.
  - Производится операция XOR над P<sub>1</sub> с первыми 32 битами ключа K, над P<sub>2</sub> со вторыми 32-битами и так далее. Если ключ K короче, то он накладывается циклически.
2. Шифрование ключей и таблиц замен
  - Алгоритм шифрования 64-битного блока, используя инициализированные ключи P<sub>1</sub> — P<sub>18</sub>

и таблицу замен  $S_1 — S_4$ , шифрует 64 битную нулевую (0x0000000000000000) строку. Результат записывается в  $P_1, P_2$ .

- $P_1$  и  $P_2$  шифруются изменёнными значениями ключей и таблиц замен. Результат записывается в  $P_3$  и  $P_4$ .
- Шифрование продолжается до изменения всех ключей  $P_1 — P_{18}$  и таблиц замен  $S_1 — S_4$ .
- 3. Шифрование текста полученными ключами и  $F(x)$ , с предварительным разбиением на блоки по 64 бита. Если невозможно разбить начальный

текст точно на блоки по 64 бита, используются различные режимы шифрования для построения сообщения, состоящего из целого числа блоков. Суммарная требуемая память 4168 байт.

Дешифрование происходит аналогично, только  $P_1 — P_{18}$  применяются в обратном порядке.

Алгоритм шифрования RSA (аббревиатура от фамилий Rivest, Shamir и Adleman) — криптографический алгоритм с открытым ключом. Принцип работы алгоритма представлен на схеме ниже.

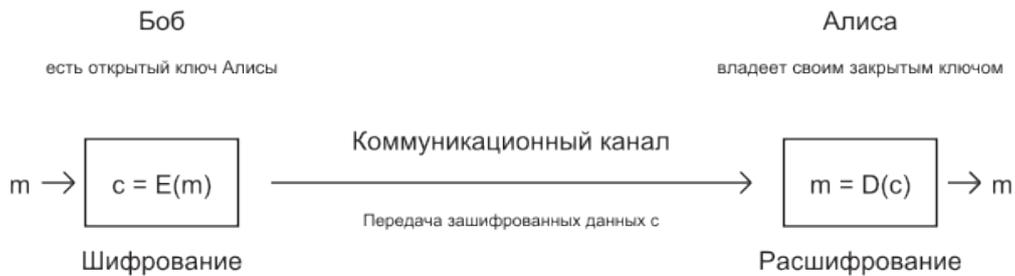


Рис. 1. Схема работы RSA (шифрование и дешифрование)

Предположим, Боб хочет послать Алисе сообщение  $m$ . Сообщениями являются целые числа в интервале от 0 до  $n-1$ , т. е.  $m$  принадлежит  $Z_n$ .

1. Алгоритм шифрования:
  - Взять открытый ключ  $(e, n)$  Алисы
  - Взять открытый текст  $m$
  - Зашифровать сообщение  $c$  использованием открытого ключа Алисы:
    - $C = E(m) = me \text{ mod } n$
2. Алгоритм дешифрования:
  - Принять зашифрованное сообщение  $c$
  - Взять свой закрытый ключ  $(d, n)$

- Применить закрытый ключ для расшифрования сообщения:

$$m = D(c) = cd \text{ mod } n$$

Данная схема на практике не используется по причине того, что она не является надёжной. Функция  $E(m)$  является детерминированной — при одних и тех же значениях входных параметров выдаёт одинаковый результат. В настоящее время используется смешанный алгоритм шифрования, в котором сначала шифруется сеансовый ключ, а потом уже с его помощью участники шифруют свои сообщения симметричными системами. После завершения сеанса сеансовый ключ, как правило, уничтожается.



Рис. 2. Алгоритм шифрования сеансового ключа

## II. Разработка алгоритма AOCRYPT

Создание программного обеспечения на основе разрабатываемого алгоритма AOCRYPT произведено на языке C++ в среде Code::Blocks — это свободная кросс-платформенная среда разработки. Code::Blocks использует библиотеку wxWidgets. Имея открытую архитектуру,

может масштабироваться за счёт подключаемых модулей. Code::Blocks разрабатывается для Windows, Linux и Mac OS X. Среду можно собрать из исходников практически под любую Unix-подобную систему, например FreeBSD, PC-BSD.

### Принцип работы AOCRYPT

Самый простой способ зашифровать данные — это просто инвертировать каждый байт файла, то есть: *новое значение* = 255 — *оригинальное значение*. Данный алгоритм является самым простым и небезопасным способом шифрования. Для надежного шифрования нужно сгенерировать ключ, но создавать его с помощью функции `rand()` крайне небезопасно в криптографии, потому что данная функция создает предсказуемые значения

ключа. Безопаснее будет использовать специальные библиотеки, но самым безопасным вариантом является генерация ключа, связанная с человеческим фактором. Человек может вводить данные с помощью клавиатуры и мыши. Создавать ключ с помощью ввода с клавиатуры займет много времени, а с мышью проще и «неповторимей», т. к. сдвиг на 1–2 пикселя повлияет на результат. Ниже приведен алгоритм генерации ключа (max = 512 байт — размер ключа):

```
for (int i = 0; i < max; i++)
{
    again:
    if (GetCursorPos(&p))
    {
        if ((p.x == oldx) && (p.y == oldy)) goto again;
        oldx = p.x;
        oldy = p.y;
        key[i] = (p.x*p.y) / 256;
        if (key[i] == 0) goto again;
    }
    delay(10);
}
```

Рис. 3. Алгоритм генерации ключа

Итак, ключ готов! Пусть файл шифруется по данному алгоритму:

*новое значение* = *ключ [i]* + *оригинальное значение*

Такой способ обеспечивает более высокую безопасность, но зная тип зашифрованного файла, можно сразу определить первые значения ключа. Например, каждое выполняемое приложение ОС Windows и некоторые для

MSDOS начинаются с MZ — инициалов Марка Збиковски. Подставив значения в формулу выше и решив простое уравнение, можно определить два значения ключа, а так как каждые 512 байт будут шифроваться одними и теми же числами, то можно расшифровать часть значений после инициалов.

4D 5A 80 00 01 00 00 00 04 00 10 00 FF FF 00 00	MZЪ.....яя..
40 01 00 00 00 00 00 00 40 00 00 00 00 00 00	@.....@.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00	.....Ъ...
0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68	..e..r.H!ë.LH!Th
69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F	is program canno
74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20	t be run in DOS
6D 6F 64 65 2E 0D 0A 24 00 00 00 00 00 00 00 00	mode...\$......

Рис. 4. Заголовок.exe файла (обведен в синий прямоугольник)

Было решено разбить ключ на две части по 256 байт. Пусть первая часть будет отвечать за числа, с которыми производят действия, а вторая — за сами действия, причем первые 4 бита — номер действия, а вторые — номер дополнительного числа в ряду. Теперь с каждым байтом файла будут производиться действия, в которых будут принимать участие 2 байта ключа.

Первое число = ключ [i]

Второе число = ключ [номер доп. числа в строке из 16 байт]

*новое значение* = *оригинальное значение* ± *результат действия*

Данный алгоритм является еще более надежным, чем предыдущие версии AOCRYPT, но если файл имеет данные, которые повторяются каждые 256 байт, то злоумышленник может сразу догадаться, что файл имеет повторяющиеся фрагменты. Заключительный этап разработки AOCRYPT — добавление сдвига действий на 4 бита каждые 256 байт данных. Конечный алгоритм шифрования данных приведен ниже. В программе используются следующие переменные:

— *ас* — массив номеров доп. чисел и действий над ними

- $f$  — номер элемента ключа;  $d$  — номер действия или доп. числа
- $oddnum$  — доп. число
- $sec$  — номер действия;  $shft$  — сдвиг (в 4 бит)
- $fdata$  — данные файла;  $fsize$  — размер файла

```

int ac[max];
int f = 0;
for (int i = 256; i < max; i++)
{
    ac[f] = key[i] / 16;
    ac[f + 1] = key[i] - ac[f] * 16;
    f += 2;
}

f = 0;
int d = 0, n = 0, sec = 0, oddnum = 0, chk = 0, shft = 0;

for (long i = 0; i < fsize; i++)
{
    n = 0;
    sec = ac[d + shft];
    repeat:
    d++;
    if (d >= max - shft) d = 0;
    if (n == 0)
    {
        n = 1;
        chk = fdata[i];
        oddnum = key[ac[d + shft] + 16 * (int)(f / 16)];
        if (sec == 0) fdata[i] += key[f] + oddnum;
        if (sec == 1) fdata[i] += key[f] - oddnum;
        if (sec == 2) fdata[i] += key[f] * oddnum;
        if (sec == 3) fdata[i] -= key[f] * oddnum;
        if (sec == 4) fdata[i] -= key[f] + oddnum;
        if (sec == 5) fdata[i] -= key[f] - oddnum;
        if (sec == 6) fdata[i] += key[f] * key[f] + oddnum;
        if (sec == 7) fdata[i] += key[f] * key[f] - oddnum;
        if (sec == 8) fdata[i] += key[f] + oddnum * oddnum;
        if (sec == 9) fdata[i] += key[f] - oddnum * oddnum;
        if (sec == 10) fdata[i] += key[f] * key[f] + oddnum * oddnum;
        if (sec == 11) fdata[i] += key[f] * key[f] - oddnum * oddnum;
        if (sec == 12) fdata[i] += key[f] * oddnum + key[f];
        if (sec == 13) fdata[i] += key[f] * oddnum - key[f];
        if (sec == 14) fdata[i] += oddnum * (key[f] + oddnum);
        if (sec == 15) fdata[i] += oddnum * (key[f] - oddnum);
        if (fdata[i] == chk) fdata[i] += 32;
        goto repeat;
    }
    f++;
    if (f >= max / 2) f = 0, shft++;
    if (shft >= max) shft = 0;
}

```

Рис. 5. Конечный вариант алгоритма шифрования

### III, Разработка программного обеспечения АООСРУТОР

Для создания графического интерфейса использован Windows API — набор базовых функций интерфейсов программирования приложений операционных систем семейств Microsoft Windows, который предоставляет прямой способ взаимодействия приложений пользователя с операционной системой Windows. Windows API представляет собой множество функций, структур данных и числовых констант, следующих соглашениям языка Си. В то же время конвенция вызова функций отличается от cdecl, принятой для языка C: Windows API использует stdcall (winapi). Для облегчения переноса на другие платформы программ, написанных с опорой на Windows API, разработана библиотека Wine.

Wine — свободное программное обеспечение, позволяющее пользователям UNIX-подобных операционных систем исполнять 16-, 32- и 64- битные приложения без наличия при этом установленной Microsoft Windows.

Wine также предоставляет программистам библиотеку программ Winelib, при помощи которой они могут компилировать Windows-приложения для портирования их в UNIX-подобные операционные системы. Название W. I. N.E. — рекурсивный акроним и расшифровывается «Wine Is Not an Emulator» — «W. I. N.E. — это не эмулятор». Имеется в виду, что Wine это не эмулятор компьютера или виртуальная машина, как, например, qemu, VirtualBox и другие подобные им. С помощью Wine данная программа может свободно работать на ОС Linux и MacOS.

В интерфейсе программы использованы следующие элементы: кнопки, поля для отображения локации файла на диске. Соответственно, при нажатии кнопки *Browse* создается диалог выбора файла, кнопки *Process* файл шифруется или расшифровывается, кнопки *Quit* программа завершает работу. Также в интерфейс добавлен логотип разработчика, рамки, музыкальное сопровождение. Все это было сжато в 50.0Кб, что обеспечивает быстрое скачивание и эксплуатацию данного ПО.

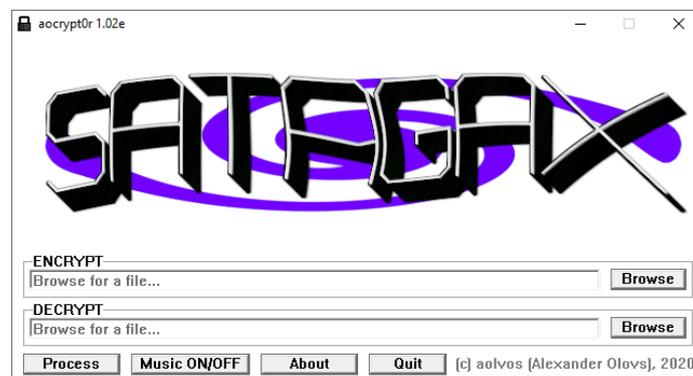


Рис. 6. Графический интерфейс программы Аосcrypt0r

При разработке программного обеспечения были решены следующие проблемы. Во-первых, во время работы программы было замечено сильное мерцание всех элементов окна. Это было обусловлено тем, что видеокарта отрисовывала большее число кадров в секунду, чем мог отрисовать монитор. Окно программы вырисовывается следующим образом: сначала происходит полное заполнение всего белым цветом, а потом начинают отображаться элементы. Так, мерцание происходило из-за того, что на экран выводились моменты, когда все окно было закрашено белым цветом. Для решения проблемы использован алгоритм вертикальной синхронизации — двойная буферизация. Для этого изображение, отрисованное видеокартой, подаётся не сразу на экран, а в специальный буфер (область памяти). Как только монитор отрисует следующий кадр, содержимое этого буфера передаётся на монитор, а в буфер пишется следующий кадр. С технической точки зрения это реализовано как смена указателя на первичный буфер (тот, из которого изображение поступает на монитор): два буфера меняются местами, и, пока изображение из нового буфера выводится на экран, следующий кадр пишется в старый буфер. Затем эта операция повторяется.

Логотип SATAGAX был сжат с помощью **собственного алгоритма imagescom-256**. Формат поддерживает до 255 цветов. Первые два слова — размеры изображения в пикселях, затем идет байт, отвечающий за количество цветов в палитре, затем данные самой палитры в формате RGB, потом номера цветов в соответствии с палитрой. Если есть повторяющиеся подряд цвета, то ставится сигнальный байт 0, затем количество повторений цвета и номер цвета в палитре. Это очень эффективно сжимает простые изображения, в отличие от BMP. Так в формате BMP логотип весит 126.0Кб, PNG: 25.2Кб, imagescom-256: 53.0Кб. Но, если PNG фактически больше напощаётся сжатию, то изображения imagescom-256 очень хорошо сжимаются, и после сжатия, например, в zip архив, размер становится равным 25.3Кб. Программа распаковывает данные в память в формате BGRA, а затем создает в памяти 24-х битное BMP изображение и выводит его на экран с помощью команды: **StretchDIBits(hdcmem, 0, 0, xres, yres, 0, 0, xres, yres, logodata, &info, DIB\_RGB\_COLORS, SRCCOPY);** xres, yres — размеры изображения, logodata — данные, info — заголовок BMP изображения в памяти.

Алгоритм работает следующим образом. Пусть существует некоторое изображение. В начале оно конвертируется в изображение с палитрой на 255 цветов, затем imagescom-256 создает и записывает в файл палитру изображения. Далее imagescom-256 считывает цвет каждого пикселя изображения и сравнивает его с цветом в палитре. Если цвета совпадают, то в файл записывается номер соответствующего цвета в палитре.

Одним из критериев при создании программного обеспечения был минимальный размер (до 50 кб). На

свободное после разработки основной части программы интерфейса место была добавлена фоновая трекерная музыка. Размер аудио файла — 12.0Кб. Для сравнения: WAV файл всего в 2 секунды весит целых 1.5Мб! Трекерная (модульная) музыка — музыка, созданная на компьютере при помощи программы «трекера» (в данном случае Fasttracker II), представляющей собой специализированный музыкальный редактор. Музыка формата XM занимает очень маленькие объемы из-за принципа хранения данных в виде номеров нот, команд и 8-ми битных WAV семплов. Для проигрывания XM формата использована библиотека UFMOD, разработанную на ассемблере.

#### *Заключение*

Таким образом, и алгоритм, и программное обеспечение для шифрования были успешно созданы, но это далеко не предел возможностей криптографии. Программа-шифровальщик имеет общий размер, равный 50.0Кб, что делает ее портативной и удобной в использовании. Кроме того, с помощью ПО Wine стал возможен запуск данной программы не только на ОС Windows, но и на Linux и MacOS! Ключ алгоритма имеет размер в 512 байт, а также зависимость выполняемых операций от ключа, что делает взлом перебором практически невозможным в настоящее время. В целом, любая программа-шифровальщик представляет собой несложное, но мощное приложение, превращающее файлы в массив данных, который можно декодировать только при наличии ключа. Данный алгоритм может применяться как для электронных подписей, так и для шифрования файлов или архивов. Способен работать с большими объемами информации, для чего и был предназначен.

Сравнение кодировок приведено ниже (1-исходный файл, 2-инвертирование, 3-аосгурт + 4-ключ)

В стадии разработки находится отдельная операционная система, загружающееся с внешнего устройства и работающая в 16-ти битном незащищенном режиме, что позволяет использовать драйвера BIOS и неограниченно изменять любые данные на жестких дисках. Данная ОС позволит зашифровывать целые диски с помощью алгоритма AOCRYPT, который будет совершенствоваться, если будут обнаружены серьезные уязвимости. Шифрование дисков целиком не только не даст загрузить систему и просмотреть файлы, но и просмотреть файлы на диске, подключив его к другому устройству. Так, работа над проектом еще не завершена, и в будущем есть множество перспектив его развития.

Разработанное программное обеспечение Aocrypt0r 1.02e находится в свободном доступе по ссылке: <http://satagax.com/data/aocrypt0r1.02e.zip> Альтернативная ссылка: <https://drive.google.com/uc?export=download&id=146XmbIRZwHj6gF7mGPg-xa8nSt1aIvjl>.

Исходный код алгоритма сжатия изображений imagescom-256 доступен по ссылке: <https://github.com/aolvos/imagescom-256>

6F 72 65 6D 20 69 70 73 75 6D 20 64 6F 6C 6F	lorem ipsum dolo	8D 90 8D 9A 92 DF 96 8F 8C 8A 92 DF 9B 90 93 90	р/м/я-ШБ/я>б"б
72 20 73 69 74 20 61 6D 65 74 2C 20 63 6F 6E 73	r sit amet, cons	8D DF 8C 96 8B DF 9E 92 9A 8B D3 DF 9C 90 91 8C	КЯБ<-Я'я' ак УЯЯБ'Б
65 63 74 65 74 75 72 20 61 64 69 70 69 73 63 69	ectetur adipisci	9A 9C 8B 9A 8B 8A 8D DF 9E 9B 96 8F 96 8C 9C 96	авк ак ЛКЯБ'>-Ц-ЯБ
6E 67 20 65 6C 69 74 2C 20 73 65 64 20 64 6F 20	ng elit, sed do	91 98 DF 9A 93 96 8B D3 DF 8C 9A 9B DF 9B 90 DF	'ЯБ'<-УЯБ>Я'ЯБ
65 69 75 73 6D 6F 64 20 74 65 6D 70 6F 72 20 69	eiusmod tempor i	9A 96 8A 8C 92 90 9B DF 8B 9A 92 8F 90 8D DF 96	а-ББ'б'Я'ак Ц'б'КЯ-
6E 63 69 64 69 64 75 6E 74 20 75 74 20 6C 61 62	ncidunt ut lab	91 9C 96 9B 96 9B 8A 91 8B DF 8A 8B DF 93 9E 9D	'в'>->б'Я'ак Я'б'к
6F 72 65 20 65 74 20 64 6F 6C 6F 72 65 20 6D 61	ore et dolore ma	90 8D 9A DF 9A 8B DF 9B 90 93 90 8D 9A DF 92 9E	б'к'Я'ак>б'б'к'Я'б'б
67 6E 61 20 61 6C 69 71 75 61 2E 20 55 74 20 65	gna aliqua. Ut e	98 91 9E DF 9E 93 96 8E 8A 9E D1 DF AA 8B DF 9A	'Я'ЯБ'>-Б'б'СЯЕ<Я'а
6E 69 6D 20 61 64 20 6D 69 6E 69 6D 20 76 65 6E	nim ad minim ven	91 96 92 DF 9E 9B DF 92 96 91 96 92 DF 89 9A 91	'-Я'ЯБ'Я'-'-'Я'ЯБ'
69 61 6D 2C 20 71 75 69 73 20 6E 6F 73 74 72 75	iam, quis nostru	96 9E 92 D3 DF 8E 8A 96 8C DF 91 90 8C 8B 8D 8A	-Я'УЯБ'Б-Я'б'к'КБ
64 20 65 78 65 72 63 69 74 61 74 69 6F 6E 20 75	d exercitatio u	9B DF 9A 87 9A 8D 9C 96 8B 9E 8B 96 90 91 DF 8A	Я'а'а'к'б'<к'<-б'Я'Б
6C 6C 61 6D 63 6F 20 6C 61 62 6F 72 69 73 20 6E	llamco laboris n	93 93 9E 92 9C 90 DF 93 9E 9D 90 8D 96 8C DF 91	"'б'б'Я'ак'б'к'к'Б-Я'а
69 73 69 20 75 74 20 61 6C 69 71 75 69 70 20 65	isi ut aliquip e	96 8C 96 DF 8A 8B DF 9E 93 96 8E 8A 96 8F DF 9A	-Б-Я'Б<Я'Б'>-Б'Б-Ц'Я'а
78 20 65 61 20 63 6F 6D 6D 6F 64 6F 20 63 6F 6E	x ea commodo con	87 DF 9A 9E DF 9C 90 92 92 90 9B 90 DF 9C 90 91	Я'а'Я'Я'Б'б' / б'>Я'Я'Б'б'
73 65 71 75 61 74 2E 20 44 75 69 73 20 61 75 74	sequat. Duis aut	8C 9A 8E 8A 9E 8B D1 DF BB 8A 96 8C DF 9E 8A 8B	Б'а'Б'Б'б'<С'Я'Б'Б-Б'Я'Б'Б
65 20 69 72 75 72 65 20 64 6F 6C 6F 72 20 69 6E	e irure dolor in	9A DF 96 8D 8A 8D 9A DF 9B 90 90 8D DF 96 91	а'Я'-К'Б'К'Я'б'б'<К'Б'<-
20 72 65 70 72 65 68 65 6E 64 65 72 69 74 20 69	reprehenderit i	DF 8D 9A 8F 8D 9A 97 9A 91 9B 9A 8D 96 8B DF 96	Я'к'а'Ц'К'а'>'а'к'<-Я'-
6E 20 76 6F 6C 75 70 74 61 74 65 20 76 65 6C 69	n voluptate veli	91 DF 89 90 93 8A 8F 8B 9E 8B 9A DF 89 9A 93 96	'Я'б'б'Б'б'<к'а'Я'а'>-
74 20 65 73 73 65 20 63 69 6C 75 6D 20 64 6F	t esse cillum do	8B DF 9A 8C 8C 9A DF 9C 96 93 93 8A 92 DF 9B 90	<Я'Б'Б'а'Б'Б'>""'Б'Я'б'>
6C 6F 72 65 20 65 70 20 66 75 67 69 61 74 20 6E	lore eu fugiat n	93 90 8D 9A DF 9A 8A DF 99 8A 8A 96 8F DF 91	"'б'к'Я'а'а'Я'Б'б'>-Б'Я'Я'
75 6C 6C 61 20 70 61 72 69 61 74 75 72 74 20 45	ulla pariatu. E	8A 93 93 9E DF 8F 9E 8D 96 9E 8B 8A 8D D1 DF BA	Б'""'Я'Ц'К'Б'<к'Б'К'Я'е
78 63 65 70 74 65 75 72 63 73 69 6E 74 20 6F 63	xcepteur sint oc	87 9C 9A 8F 8B 9A 8A 8D DF 8C 96 91 8B DF 90 8C	Я'а'Б'<а'Б'К'Б'>'-'Я'Б'б'
63 61 65 63 61 74 20 63 75 70 69 64 61 74 61 74	caecat cupidatad	9C 9E 9A 9C 9E 8B DF 9C 8A 8F 96 9B 9E 8B 9E 8B	Б'а'а'а'Б'<к'Б'Б'>'-'Я'Б'б'
20 6E 6F 6E 20 70 72 6F 69 64 65 6E 74 2C 20 73	non proident, s	DF 91 90 91 DF 8F 8D 90 96 9B 9A 91 8B DF 8C 8C	Я'б'Я'Ц'К'Б'>'>'<У'Я'Б'
75 6E 74 20 69 6E 20 63 75 6C 70 61 20 71 75 69	unt in culpa qui	8A 91 8B DF 96 91 DF 9C 8A 93 8F 9E DF 8E 8A 96	Б'<Я'-Я'Б'Б'>'Ц'Я'Б'Б'>
20 6F 66 66 69 63 69 61 20 64 65 73 65 72 75 6E	officia deserun	DF 90 99 99 96 9C 96 9E DF 9B 9A 8C 9A 8D 8A 91	Я'Б'Б'Б'>-Б'Я'Б'а'Б'Б'Б'>
74 20 6D 6F 74 2C 6C 69 74 20 61 6E 69 6D 20 69 64	t mollit anim id	8B DF 92 90 93 93 96 8B DF 9E 91 96 92 DF 96 9B	<Я'б'>""<-Я'Б'Я'-Я'
20 65 73 64 20 6C 61 62 6F 72 75 6D 2E	est laborum.	DF 9A 8C 8B DF 93 9E 9D 90 8D 8A 92 D1	Я'а'Б'<Я'Б'к'Б'К'Б'>'С

Рис. 7. Сравнение кодировок

ЛИТЕРАТУРА:

1. Баричев, С. Г. Основы современной криптографии. — М.: СИНТЕГ, 2011.
2. Фергюсон, Н.. Практическая криптография. — М.: Диалектика, 2004.
3. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы. — М.: Триумф, 2002.
4. Интернет-энциклопедия «Википедия». RSA. MZ. Blowfish [Электронный ресурс]. URL: <https://ru.Wikipedia.org/wiki/RSA>; <https://ru.wikipedia.org/wiki/MZ>; <https://ru.wikipedia.org/wiki/Blowfish>
5. Официальный сайт UFMOD [Электронный ресурс]. URL: [https://ufmod.sourceforge.io/ind\\_ru.htm](https://ufmod.sourceforge.io/ind_ru.htm)
6. Портал stackoverflow.com [Электронный ресурс]. URL: <https://ru.stackoverflow.com/>
7. Портал rohos.ru [Электронный ресурс]. URL: [http://www.rohos.ru/help/crypto\\_algorithms](http://www.rohos.ru/help/crypto_algorithms)