## Условия задач заключительного этапа

## Задача 1. Гербы

Аналитику удалось обнаружить папку с графическими изображениями и текстовым файлом. Известно, что в изображениях скрыто кодовое слово. Помогите определить кодовое слово, если известно, что для его сокрытия изменили содержимое всех файлов.

*К задаче прилагается:*

*1) 6 файлов-изображений (*.jpg),*

*2) текстовый файл bytes.txt.*

## Задача 2. Аутентификация

Система аутентификации шифрует пароли особым образом, показанном в виде функции `scrambler` на языке C++. Зная алгоритм шифрования, вычислите пароль. Фрагмент кода указан ниже.

**C++**

```
int swap(int value, int start, int len)
{
 len = len % (sizeof(value) * 8);
 start = start % (sizeof(value) * 8);

 int bits = (INT_MAX >> ((sizeof(value) * 8) - len - 1));
 bits = bits << start;

 int buf = (value >> len) & bits;
    int res = value & bits;
    res = res << len;
    res |= buf;

    int rest = value & ~bits;
    rest = rest & ~(bits << len);

    return rest | res;
 }

 ////////////////////////////////////////////////////
 void scrambler(int keyword)
 {
    int res = keyword;
    for (int i = 1; i < 16; i = i * 2)
    {
      for (int j = 0; j < 32 - i; j = j + i * 2)
        res = swap(res, j, i);
    }

    if (res == 1268560121)
      std::cout << "Password is correct\n";
    else
      std::cout << "Password is wrong\n";
```

## Задача 3. Сеть LOR

Один студент решил создать свою анонимную сеть с шифрованием и виртуальными тоннелями и назвал её LOR.

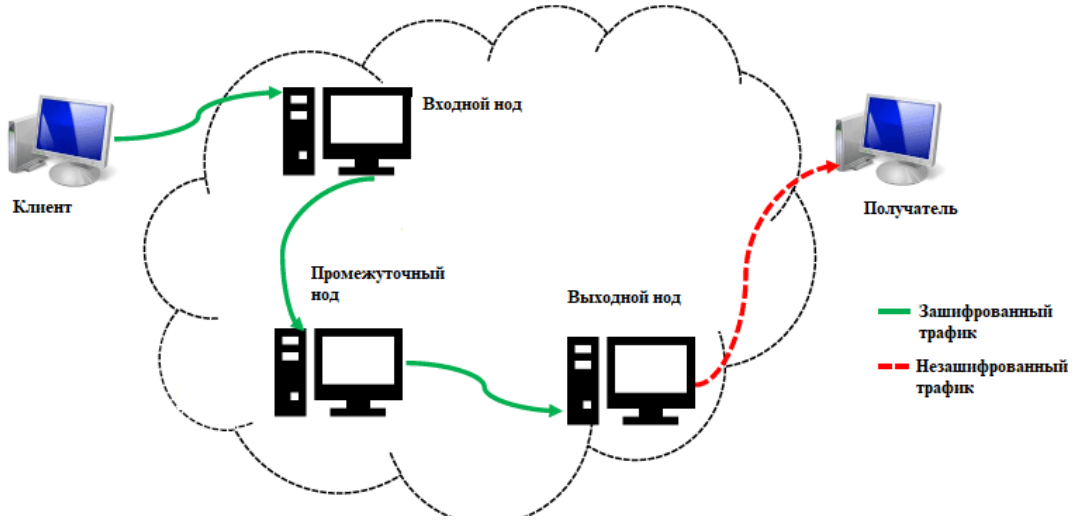


Рисунок. Схема сети LOR

*Нод* – узел сети LOR, способный принимать данные, расшифровывать и передавать их.

Чтобы отправить данные, клиент три раза шифрует их особым методом. Далее зашифрованная информация передается входному ноду, который расшифровывает её один раз. После этого данные отправляются на промежуточный нод, который так же расшифровывает их один раз. Далее промежуточный нод отправляет данные выходному ноду, который расшифровывает их третий раз, получая данные уже в открытом виде. После этого данные в открытом виде отправляются получателю.

Используемая функция шифрования:

$$\mathbf{E(x) = (ax + b) \bmod m},$$ где

**x** – номер шифруемого символа (см. таблицу),

**a** и **b** – ключи, при этом **a** и **m** должны быть взаимно простыми (НОД(a, m) = 1, a < m),

**m** – количество символов в алфавите (m = 30).

| **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **M** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| **N** | **O** | **P** | **Q** | **R** | **S** | **T** | **U** | **V** | **W** | **X** | **Y** | **Z** |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| **.** | **,** | **(пробел)** | **–** | | | | | | | | | |
| 26 | 27 | 28 | 29 | | | | | | | | | |

При первом шифровании ключ **a** выбирается так, чтобы **a** и **m** были взаимно простыми (НОД(a, m) = 1, a < m).

При втором и третьем шифровании ключ **a** равен номеру первого зашифрованного символа сообщения, полученного после применения шифрования. Если номер первого зашифрованного символа не является взаимно простым к **m**, то в качестве ключа **a** берется ближайшее большее число, удовлетворяющее правилу. Если такого числа нет (например, номер символа равен 30), то в качестве ключа используется значение 1.

Для расшифрования используется другая функция:

$$\mathbf{D(x) = a^{-1}(E(x) - b) \bmod m},$$ где

$\mathbf{a^{-1}}$ – число, обратное **a** по модулю **m** ($a * a^{-1} = 1 \bmod m$). При этом, число $\mathbf{a^{-1}}$ так же удовлетворяет условию: $НОД(a^{-1}, m) = 1$, $a^{-1} < m$.

Расшифруйте отправленное клиентом сообщение, если известно, что b = 5 на всех нодах, а исходное сообщение заканчивается символом "**.**". В ответе укажите исходное сообщение, а также ключи шифрования входного, промежуточного и выходного нода.

Перехваченное сообщение от клиента:

```
YMXNDXNDYMJDJS L
```

## Задача 4. DLL Hijacking

В разделе импорта заголовка исполняемого файла содержится информация о подключаемых библиотеках (DLL) и импортируемых из них функциях. Для подмены одной из DLL необходимо, чтобы имя библиотеки и набор функций совпадали с именем библиотеки и набором функций, описанными в разделе импорта исполняемого файла. Для упрощения разработки подменяемой библиотеки DLL, из всех библиотек выбирают ту, из которой импортируется наименьшее количество функций.

Из предоставленного образа раздела импорта определите имя библиотеки, из которой импортируется наименьшее количество функций.

В ответе укажите имя библиотеки DLL, имена импортируемых из нее функций и количество аргументов каждой из таких функций.

Структура раздела импорта показана на рисунке.

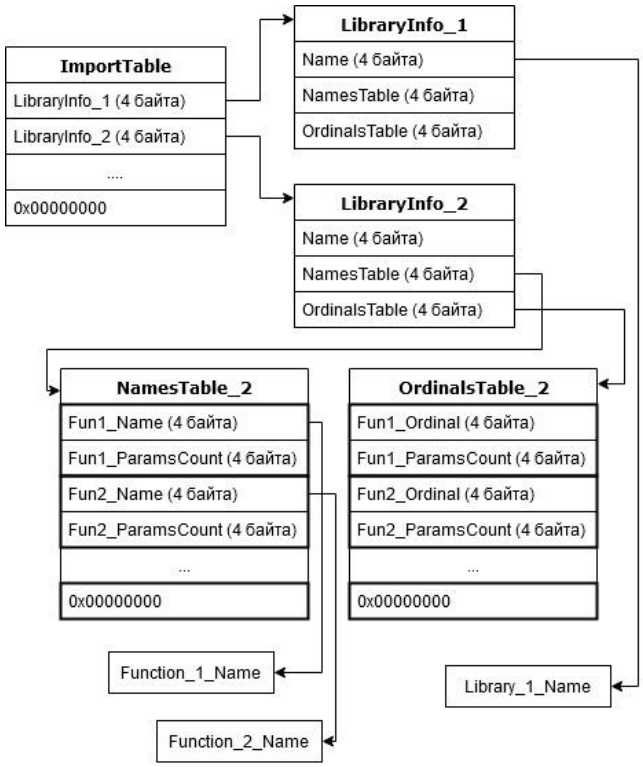


Рисунок. Структура раздела импорта

**ImportTable** – таблица импорта. Хранится в начале раздела импорта. Содержит адреса (4-х байтовые) на структуры *LibraryInfo*. В конце таблицы записывается 4 нулевых байта

(0x00000000).

**LibraryInfo** – структура, содержащая информацию о библиотеке. Содержит поля:

*Name* – адрес строки (4 байта), содержащей имя библиотеки. Строка заканчивается нулевым байтом ('\0');

*NamesTable* – адрес таблицы имен импортируемых функций (4 байта);

*OrdinalsTable* – адрес таблицы идентификаторов импортируемых функций (4 байта).

**NamesTable** – таблица имен импортируемых функций. Каждая запись содержит следующие параметры:

*Fun_Name* – адрес строки (4 байта), содержащей имя функции. Строка заканчивается нулевым байтом ('\0');

*Fun_ParamsCount* – количество аргументов функции (4 байта).

В конце таблицы записывается 4 нулевых байта (0x00000000).

**OrdinalsTable** – таблица идентификаторов импортируемых функций. Каждая запись содержит следующие параметры:

*Fun_Ordinal* – идентификатор функции (4 байта);

*Fun_ParamsCount* – количество аргументов функции (4 байта).

В конце таблицы записывается 4 нулевых байта (0x00000000).

Все адреса и числовые значения хранятся в формате *Little-Endian*. Адреса указываются относительно начала раздела импорта, адрес которого считается равным 0x00000000.

*К задаче прилагается:*
*файл образа раздела импорта dump_v1.bin.*

## Задача 5. Web-сайт

Пользователь хранит на сервере секретное слово, доступ к которому можно получить, авторизовавшись через web-сайт. Сервер выдаст секретное слово только в том случае, если ему будет отправлена верная зашифрованная последовательность, сформированная из логина и пароля. Чтобы не забыть логин и пароль, пользователь оставил себе подсказки на сайте.

Определите секретное слово.

*К задаче прилагается:*
*папка с содержимым web-страницы.*