УДК 004

Программирование шифра Плейфера на PHP

Стрельцова Марина Николаевна
Приамурский государственный университет им. Шолом-Алейхема
Студент

Аннотация
Защита информации еще с древних времен стала одной из актуальных задач человечества. Одним из способов защиты информации стали шифры, основной миссией которых являлось ограничение доступа к информации третьих лиц, которые не должны знать суть передаваемого сообщения. В данной статье рассмотрено программирование шифра Плейфера на языке программирования PHP.
Ключевые слова: PHP, шифр, Плейфер, программирование

Playfair cipher programming in PHP

Streltsova Marina Nikolaevna
Sholom-Aleichem Priamursky State University
Student

Abstract
Protection of information since ancient times has become one of the urgent tasks of mankind. One of the ways to protect information was ciphers, the main mission of which was to restrict access to information by third parties who should not have known the essence of the transmitted message. This article discusses the programming of the Playfair cipher in the PHP programming language.
Keywords: PHP, cipher, Playfair, programming

1. Введение

1.1 Актуальность исследования

Защита информации еще с древних времен стала одной из актуальных задач человечества. Одним из способов защиты информации стали шифры, основной миссией которых являлось ограничение доступа к информации третьих лиц, которые не должны были знать суть передаваемого сообщения. Одним из таких шифров является Шифр Плейфера - ручная симметричная техника шифрования, в которой впервые использована замена биграмм. Изобретена в 1854 году английским физиком Чарльзом Уитстоном, но названа именем лорда Лайона Плейфера, который внёс большой вклад в продвижение использования данной системы шифрования в государственной службе. Шифр предусматривает шифрование пар символов (биграмм) вместо одиночных символов, как в шифре подстановки и в более сложных системах

шифрования Виженера [1]. В данной статье рассмотрено программирование шифра Плейфера на языке программирования PHP.

1.2 Обзор исследований

В исследовании Е. Д. Жирнякова рассмотрен алгоритм шифрования данных, реализованный на языке программирования Python, с использованием симметрического ключа шифрования данных (AES) [2]. Хеширование данных с помощью библиотек языка программирования Python, с указанием основных функций и области их применения, описывается в научной работе Н. А. Головченко и М. А Печкурова [3]. В статье А. А. Бусенкова и др. описана разработка программного обеспечения для безопасного хранения и обработки персональных данных в облачном хранилище с локальным шифрованием [4]. Д. В. Аратин и К. В. Гилева в научной работе проводят анализ двух методов шифрования: симметричное шифрование и асимметричное шифрование [5]. О наиболее перспективном направлении в криптографии, которое сможет обеспечить безопасность и конфиденциальность данных, хранящихся в облачных сервисах – шифрование с возможностью поиска описывают в работе А. Л. Ханис и других соавторов [6].

1.3 Цель исследования

Целью данного исследования является написание кода, реализующего алгоритм шифра Плейфера, на языке программирования PHP.

2. Алгоритм шифрования и дешифрования

Шифр Плейфера использует матрицу 5x5, содержащую ключевое слово или фразу. Для создания матрицы и использования шифра достаточно запомнить ключевое слово и четыре простых правила. Чтобы составить ключевую матрицу, в первую очередь нужно заполнить пустые ячейки матрицы буквами ключевого слова (не записывая повторяющиеся символы), потом заполнить оставшиеся ячейки матрицы символами алфавита, не встречающимися в ключевом слове, по порядку. Ключевое слово может быть записано в верхней строке матрицы слева направо, либо по спирали из левого верхнего угла к центру. Ключевое слово, дополненное алфавитом, составляет матрицу 5x5 и является ключом шифра.

Для того чтобы зашифровать сообщение, необходимо разбить его на биграммы, например, «Hello World» становится «HE LL OW OR LD», и отыскать эти биграммы в таблице. Два символа биграммы соответствуют углам прямоугольника в ключевой матрице. Далее следует определить положения углов этого прямоугольника относительно друг друга. Затем, руководствуясь следующими 4 правилами, зашифровать пары символов исходного текста:

1. Если два символа биграммы совпадают (или если остался один символ), то добавляется после первого символ «X», зашифровывается новая пара символов и так далее.

2. Если символы биграммы исходного текста встречаются в одной строке, то эти символы замещаются на символы, расположенные в ближайших столбцах справа от соответствующих символов. Если символ является последним в строке, то он заменяется на первый символ этой же строки.

3. Если символы биграммы исходного текста встречаются в одном столбце, то они преобразуются в символы того же столбца, находящиеся непосредственно под ними. Если символ является нижним в столбце, то он заменяется на первый символ этого же столбца.

4. Если символы биграммы исходного текста находятся в разных столбцах и разных строках, то они заменяются на символы, находящиеся в тех же строках, но соответствующие другим углам прямоугольника.

Для расшифровки необходимо использовать инверсию этих четырёх правил, откидывая символы «X», если они не несут смысла в исходном сообщении [1].

3. Результаты исследования

Для начала создадим файл с расширением .php и напишем функцию «prepare» (Рис. 1).

Рисунок 1 – Функция «prepare»

Сначала объявляем алфавит шифра, затем преобразуем ключевое слово в массив без повторений букв в нем (строки 4-5). Далее удаляем буквы в переменной с алфавитом из ключевого слова (строки 6-7). Следующим шагом преобразуем алфавит в матрицу 5x5, пишем обработку дублирований в биграмме, добавляем символ «X», если биграмма неполная и непосредственно формируем сами биграммы (строки 11-25). Последний цикл в данной функции будет отвечать за присваивание каждой букве кодируемого сообщения индексного числа (строки 26-35).

Далее напишем функцию «Encode» для кодирования сообщения с выбранным ключом (Рис. 2).

```php
function Encode($keyword, $code){
    [$index, $matrix] = prepare($keyword, $code);

    $a = 0;
    $b = 0;
    echo "\nencrypted sequence: \n";
    for($i = 0; $i < strlen($index)-3; $i += 4){
        // если на одной строке
        $a = $index[$i + 3] + 1;
        if($index[$i] == $index[$i + 2] && $a < 5){
            echo $matrix[$index[$i]][$index[$i + 1] + 1] . $matrix[$index[$i + 2]][$a] . " ";
        }
        // если на одной строке и у края
        if($index[$i] == $index[$i + 2] && $a >= 5){
            $a = 0;
            echo $matrix[$index[$i]][$index[$i + 1] + 1] . $matrix[$index[$i + 2]][$a] . " ";
        }
        // если в одном столбце
        $b = $index[$i + 2] + 1;
        if($index[$i + 1] == $index[$i + 3] && $b < 5){
            echo $matrix[$index[$i] + 1][$index[$i + 1]] . $matrix[$b][$index[$i + 3]] . " ";
        }
        // если в одном столбце и у края
        if($index[$i + 1] == $index[$i + 3] && $b >= 5){
            $b = 0;
            echo $matrix[$index[$i] + 1][$index[$i + 1]] . $matrix[$b][$index[$i + 3]] . " ";
        }

        if($index[$i] != $index[$i + 2] && $index[$i + 1] != $index[$i + 3]){
            echo $matrix[$index[$i + 2]][$index[$i + 1]] . $matrix[$index[$i]][$index[$i + 3]] . " ";
        }
    }
}
```

Рисунок 2 – Функция «Encode»

Заменяем согласно четырем правилам шифра буквы кодируемого сообщения на буквы из алфавита и, на примере ключевого слова «DOG» «TEACHER», получаем зашифрованную последовательность букв «HRHDIFWS» (строки 49-81). Также выведем матрицу, биграммы и индексные значения каждой буквы (Рис. 3).

```
Terminal:  Local  +

D:\projects\plaifer>php shifr.php
D O G A B
C E F H I
J K L M N
P R S T U
V W X Y Z
TE AC HE RX
3311031013113142
encrypted sequence:
HR HD IF WS
```

Рисунок 3 – Полученный зашифрованный текст

Аналогичным образом декодируем зашифрованное сообщение в функции «Decode» (строки 83-117) (Рис. 4).

```php
function Decode($keyword, $code){
    [$index, $matrix] = prepare($keyword, $code);

    $a = 0;
    $b = 0;
    echo "\ndecrypted sequence: \n";
    for($i = 0; $i < strlen($index)-3; $i +=4){

        // если на одной строке
        $a = $index[$i + 3] - 1;
        if($index[$i] == $index[$i + 2] && $a >= 0){
            echo $matrix[$index[$i]][$index[$i + 1] - 1] . $matrix[$index[$i + 2]][$a] . " ";
        }
        // если на одной строке и у края
        if($index[$i] == $index[$i + 2] && $a < 0){
            $a = 4;
            echo $matrix[$index[$i]][$index[$i + 1] - 1] . $matrix[$index[$i + 2]][$a] . " ";
        }

        // если в одном столбце
        $b = $index[$i + 2] - 1;
        if($index[$i + 1] == $index[$i + 3] && $b >= 0){
            echo $matrix[$index[$i] - 1][$index[$i + 1]] . $matrix[$b][$index[$i + 3]] . " ";
        }
        // если в одном столбце и у края
        if($index[$i + 1] == $index[$i + 3] && $b < 0){
            $b = 4;
            echo $matrix[$index[$i] - 1][$index[$i + 1]] . $matrix[$b][$index[$i + 3]] . " ";
        }

        if($index[$i] != $index[$i + 2] && $index[$i + 1] != $index[$i + 3]){
            echo $matrix[$index[$i + 2]][$index[$i + 1]] . $matrix[$index[$i]][$index[$i + 3]] . " ";
        }
    }
}
```

Рисунок 4 – Функция «Decode»

Проверим результат работы кода, передав ключевое слово и зашифрованное сообщение (Рис. 5).

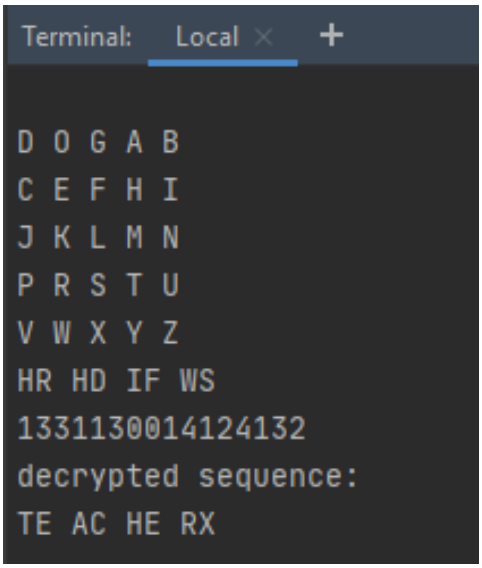

Рисунок 5 – Результат дешифровки

4. Выводы

Шифрование или кодирование информации — это распространенный способ ее защиты. Конфиденциальность и недоступность информации играет очень большую роль и поэтому одним из надежных способов ее защиты является шифрование. В настоящее время с развитием компьютерных технологий огромное значение также уделяют защите различных персональных данных с помощью шифрования и криптографии. В рамках данной статьи был написан код на языке программирования PHP для реализации алгоритма шифра Плейфера.

Библиографический список

1. Шифр Плейфера: https://ru.wikipedia.org/wiki/Шифр_Плейфера (дата обращения: 30.01.2023).
2. Жирняков Е. Д. Защита конфиденциальной информации в общеобразовательной организации с помощью симметричного шифрования данных // ББК–66.5 я43 Н 35 Национальная безопасность и молодежная политика: киберсоциализация и трансформация. 2021. С. 32.
3. Головченко Н. А., Печкуров М. А. Обеспечение информационной безопасности данных с помощью библиотек python // Состояние и перспективы развития современной науки по направлению «Информационная безопасность». 2020. С. 369-373.
4. Бусенков А. А. и др. Разработка алгоритма и программная реализация средства защиты персональных данных в облачных хранилищах // Инновации и инвестиции. 2021. №. 12. С. 264-271.
5. Аратин Д. В., Гилева К. В. Шифрование информации симметричным способом // Россия молодая. 2021. С. 31415.1-31415.7.
6. Ханис А. Л. и др. Шифрование данных в облаке с возможностью поиска // Интеллектуальные информационные системы: тенденции, проблемы, перспективы. 2019. С. 85-100.
7. PHP URL: https://www.php.net/ (дата обращения: 10.01.2023).
8. PHPStorm URL: https://www.jetbrains.com/ru-ru/phpstorm/ (дата обращения: 10.01.2023).