УДК 004.624

Шифрование информации средствами языка программирования PHP

Ересь Артём Владимирович
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье будет рассмотрен процесс шифрования информации с использованием языка PHP. Представлен пример данной процедуры для формы ввода данных на сайте и отражены возможности расшифровки.
Ключевые слова: PHP, данные, шифрование, web

Enciphering of information means of the PHP programming language

Yeres Artem Vladimirovich
Sholom-Aleichem Priamursky State University
Student

Abstract
In this article process of enciphering of information about use of the PHP language will be considered. The example of this procedure for a data input form on the website is presented and possibilities of interpretation are reflected.
Keywords: PHP, data, enciphering, web

При разработке собственного сайта, по определенной тематике, каждый программист осознает важность защиты информации. Это может быть личная информация администратора, но чаще всего это данные полученные от пользователей. Для повышения безопасности информацию можно зашифровать, что придаст уверенности в сохранности и конфиденциальности, защитит от несанкционированного использования.

Целью данной работы является рассмотрение процесса шифрования и расшифровки данных с использованием методов языка программирования PHP.

Тема данной статьи является актуальной в современной научно-исследовательской деятельности. С.А. Яремчук рассматривает способы защиты приложений с помощью программы решений PHP. Автор говорит о популярности и простоте данного языка программирования [1]. Исследователи А.А. Горошки и А.А. Жердев подробности обратили внимание на методы защиты скриптов в PHP, провели анализ существующего программного обеспечения для этих целей и классифицировали его [2]. Авторы А.А. Петров и А.А. Оголюк выделили актуальную проблему современного мира информационных технологий – копирование чужих NET приложений. Оценили возможности защиты в таких

ситуация, выявили положительные и отрицательные стороны [3]. Интернет-источники подробно рассматривают язык программирования PHP с точки зрения шифрования, классифицируют его виды и дают понимание вариантов использования для конкретных методов в зависимости от вида информации [4-5].

В начале работы необходимо создать форму с возможностью ввода текста. Именно он будет шифроваться в работе.

```
<form method="post">
<input type="text" name="str">

<input type="submit" value="Ok">
</form>
```

Рис. 1. Форма для текста

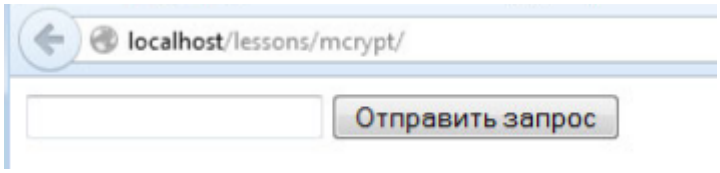

Рис. 2. Вид в браузере

Информация, которую мы будем вводить форму, будет шифроваться и выводить на экран в защищенном символьном виде. Теперь стоит создать переменную, содержащую ключ для расшифровки. Стоит отметить, что доступ к ключу стоит ограничить с целью безопасности, и его хранение должно быть максимально конфиденциально.

Создадим его в index.php:

```
$key = "GDSHG4385743HGSDHdkfgjdfk4653475JSGHDJSDSKJDF476354";
```

Рис. 3. Ключ шифрования

Далее добавим $ str, где будем хранить значение ячейки необходимого массива.

```
$str = $_POST['str'];
```

Рис. 4. Переменная

Открываем скрипт используя функцию mcrypt module open, где задаем алгоритм и режим.

```
//открываем модуль шифрования и получаем его дискриптор
 $td = mcrypt_module_open(MCRYPT_BLOWFISH,'',MCRYPT_MODE_CFB,'');
```

Рис. 5. Модуль для работы

Следующим шагом будет создание вектора, который является частью процесса шифрования. Главная характеристика для него – это длина, определяющаяся как mcrypt enc get iv size.

```
1 // Создание вектора шифрования
2 $iv = mcrypt_create_iv($iv_size, MCRYPT_RAND);
```

Рис. 6. Создание вектора

Чтобы процесс шифрования был начат следует использовать буфер обмена. Функция mcrypt generic init открывает его.

```
1 // открытие буфера обмена, для шифровки данных
2 mcrypt_generic_init($td,$key,$iv);
```

Рис. 7. Буфер обмена

Теперь шифруем наши данные через mcrypt generic. Параметры: модуль и строчка для шифрования. И в конце буфер обмена закрываем обязательно. Для правильного вывода на экран обработаем нашу строку специальным методом.

```
1 echo base64_encode($iv.$crypt_text);
```

Рис. 8. Метод для верного вывода

Проверим работу нашего шифрования. Введем текст hello world.

Рис. 9. Результат шифрования

После удачного результата стоит проверить возможность расшифровки.

Возвращаемся к mcrypt и определяем вектор:

```
1 $td = mcrypt_module_open(MCRYPT_BLOWFISH,'',MCRYPT_MODE_CFB,'');
2 $iv_size1 = mcrypt_enc_get_iv_size($td);
```

Рис. 10. Определение длины

Далее удаляем часть строки до длины $ iv size 1 и получаем:

```
1 $crypt_text1 = substr($iv.$crypt_text,$iv_size1);
```

Рис. 11. Конечная строка

Используем функцию для расшифровки с параметрами такими же как в противоположном процессе:

```
1 mcrypt_generic_deinit($td);
2 mcrypt_module_close($td);
3 echo "<br />".$text;
```

Рис. 12. Функция расшифровки

Вводим hello world и получаем результат:

Рис. 13. Результат расшифровки

Таким образом, в работе был рассмотрен процесс шифрования информации на языке программирования PHP. Так же было дано описание процесса расшифровки.

Библиографический список

1. Яремчук С.А. Защищаем PHP-приложения с помощью SIHOSIN // Системный администратор. 2006. №11. С. 74-77. URL: https://elibrary.ru/item.asp?id=20393988 (Дата обращения: 22.12.2018)
2. Горошкин А.А., Жердев А.А. Выбор программного обеспечения для наилучшей защиты PHP-скриптов // Научный вестник Московского государственного горного университета. 2011. №7. С. 17-24. URL: https://elibrary.ru/item.asp?id=16541244 (Дата обращения: 22.12.2018)
3. Петров А.А., Оголюк А.А. Программные методы защиты NET приложений от несанкционированного доступа // Вопросы защиты информации. 2014. №22. С. 82-84. URL: https://elibrary.ru/item.asp?id=21545950 (Дата обращения: 22.12.2018)
4. Простая и надежная PHP функция для обратимого шифрования URL: https://vladimirkim.livejournal.com/674.html (Дата обращения: 22.12.2018)
5. Шифрование в PHP URL: http://ekimoff.ru/316/ (Дата обращения: 22.12.2018)