**УДК 004.421.5**

# ГЕНЕРАЦИЯ СЛУЧАЙНЫХ (ПСЕВДОСЛУЧАЙНЫХ) ЧИСЕЛ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PHP

***Н.И. ЛАПКОВСКИЙ***
***(Представлено: Д.В. ПЯТКИН)***

*Рассмотрена генерация случайных(псевдослучайных) чисел в языке программирования PHP. Приведём примеры применения псевдослучайных чисел в PHP. Приведены возможные способы нахождения псевдослучайного числа. Исследуем процесс генерации псевдослучайного числа.*

Случайные числа — это неотъемлемая часть программирования, особенно, если это касается систем безопасности. К примеру, криптография основывается на генерации случайных значений, которые невозможно предугадать. Конечно же и в PHP случайные числа играют огромную роль: с их помощью можно генерировать токены, идентификаторы и другие значения.

Псевдослучайные числа — вырабатываемая алгоритмически последовательность чисел, обладающих свойствами случайных чисел и используемых взамен последних при решении на ЭВМ ряда классов задач.

Как сказано выше, генерация псевдослучайных чисел основывается на специальных алгоритмах [1]. Входным параметром, от которого будет отталкиваться алгоритм, может быть как случайное значение, так и заранее определённое.

В PHP псевдослучайные числа используются для различных целей. В основном, они связаны с безопасностью. На их основе генерируются токены, ключи, значения для аутентификации, значения для сброса паролей и многое другое. Всё это делается для того, чтобы получаемые значения невозможно было предугадать.

Некоторые важные примеры применения псевдослучайных значений:

– Генерация значений для криптографии — псевдослучайное число используется, например, для шифрования «в одну сторону», а также для хэширования паролей. Также псевдослучайное значение используется как вектор инициализации в криптографии;

– Генерация псевдослучайных значений, таких как ID сессии — PHP используется для создания огромного количества приложений, где безопасность стоит на первом месте. Многий функционал базируется на работе с сессиями и генерированными ID сессий;

– Генерация токенов для аутентификации, которые практически невозможно предугадать — многие PHP приложения базируются на работе с другими системами через специальные API и интерфейсы. Обычно перед использованием API нужно пройти процесс аутентификации. Получать трудноподбираемые значения для токенов очень сложно. Именно поэтому в данных задачах используются псевдослучайные числа.

API (программный интерфейс приложения) (англ. application programming interface, API) — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах[2].

**Генераторы псевдослучайных чисел.** Псевдослучайные числа, использующиеся в случаях, описанных выше, в PHP генерируются псевдо-генераторами. Всего доступно несколько алгоритмов:

– Линейный конгруэнтный метод, при использовании функции lcg_value(). Функция lcg_value() возвращает псевдослучайное значение в диапазоне (0,1). Функция комбинирует два КГ с периодами 2^31 – 85 и 2^31 – 249. Период этой функции равен произведению базовых [3];

– Вихрь Мерсенна, используется функцией mt_rand(). Функция mt_rand() представляет собой замену старой функции rand(). Она использует генератор псевдослучайных чисел с известными характеристиками, основанный на Вихре Мерсенна, который генерирует псевдослучайные числа в среднем в 4 раза быстрее, чем функция rand(). Вызванная без обязательных параметров min и max, функция mt_rand() возвращает псевдослучайное значение между 0 и RAND_MAX. Если нужно, например, псевдослучайное число между 5 и 15(включительно), следует использовать вызов mt_rand(5, 15)[4];

– Функция rand(), использующая аналогичную функцию в языке Си [5].

Фактически данные функции возвращают не случайные числа, а числа, распределённые таким образом, что они выглядят как случайные. Последовательность этих чисел зависит от базового случайного числа внутри реализованного алгоритма.

**Базовые числа для генераторов.** Базовые числа или вектора таких чисел — это наборы данных, которые используются для генерации псевдослучайных чисел. Псевдо-генераторы случайных чисел работают, только отталкиваясь от них. Если злоумышленник узнает это базовое число, то в будущем сможет предугадать значения ваших псевдослучайных чисел.

В PHP можно задать базовые числа двумя способами. Первый — это используя функцию mt_srand(). Этот способ в основном используется при юнит тестах случайного ряда.

Второй способ — это предоставление PHP право самому генерировать базовые числа. Начиная с версии 4.2, PHP предоставляет эту возможность. В последствии для генерации псевдослучайных чисел будет задействован Вихрь Мерсенна.

PHP генерирует базовое число, в зависимости от операционной системы. На платформах Linux можно воспользоваться функциями mcrypt_create_iv() или openssl_pseudo_random_bytes() в /dev/urandom. Windows предоставляет специальный псевдо-генератор, к которому можно получить доступ через функции openssl_pseudo_random_bytes() и mcrypt_create_iv().

Таким образом, в статье рассмотрены различные способы генерации псевдослучайных чисел и особенности их использования. Исследован процесс генерации псевдослучайного числа в языке программирования PHP.

ЛИТЕРАТУРА

1. Псевдослучайные числа [Электронный ресурс]. – Режим доступа: https://economic_mathematics.academic.ru/3716/Псевдослучайные_числа. – Дата доступа: 24.09.2018.
2. API [Электронный ресурс] // Материал из Википедии – свободной энциклопедии. – Режим доступа: https://ru.wikipedia.org/wiki/API. – Дата доступа: 24.09.2018.
3. lcg_value [Электронный ресурс]. – Режим доступа: http://php.net/manual/ru/function.lcg-value.php. – Дата доступа: 24.09.2018.
4. mt_rand [Электронный ресурс]. – Режим доступа: http://php.net/manual/ru/function.mt-rand.php. – Дата доступа: 25.09.2018.
5. rand [Электронный ресурс]. – Режим доступа: http://php.net/manual/ru/function.rand.php. – Дата доступа: 25.09.2018.