Szyfrowanie w PHP

1. Korzystając ze strony php.net odszukaj informacje na temat poniższych funkcji. Dla każdej z nich zapisz jak działa (co zwraca) i jakie ma argumenty.

- sha1()
- sha1_file()
- crc32()
- md5()
- hash()
- crypt()
- password_hash()

2. Zapoznaj się z informacjami poniżej:

Szyfrowanie haseł

Niezależnie od tego, czy dane użytkowników będą przechowywane w pliku, czy w bazie danych, lepiej nie decydować się na ryzykowne zapisywanie haseł dostępu w formie otwartego tekstu. Jednokierunkowy algorytm mieszający zapewni większe bezpieczeństwo bez dodatkowego nakładu pracy.

PHP udostępnia szereg jednokierunkowych funkcji mieszających. Najstarszym, ale też najmniej bezpiecznym rozwiązaniem jest algorytm Unix Crypt realizowany przez funkcję `crypt()`. Algorytm Message Digest 5 (MD5) zaimplementowany w funkcji `md5()` jest silniejszy i dostępny w większości wersji PHP. Jeżeli zgodność z wcześniejszymi wersjami PHP nie jest wymagana, można wykorzystać algorytm Secure Hash Algorithm 1 (SHA-1).

Funkcja PHP `sha1()` jest silną mieszającą jednokierunkową funkcją szyfrującą. Jej prototyp jest następujący:

```
string sha1(string łańcuch [, bool łańcuch_binarny])
```

Mając dany łańcuch znaków `łańcuch`, funkcja zwróci pseudolosowy ciąg 40 znaków. Jeśli parametrowi `łańcuch_binarny` przypisana zostanie wartość `true`, zamiast tego łańcucha znaków zwrócony zostanie 20-znakowy łańcuch danych binarnych. Na przykład, jeśli łańcuch wejściowy ma postać `"haslo"`, to efektem wykonania funkcji `sha1()` jest ciąg znaków w postaci `" ff12bbd8c907af067070211d87bdf098be17375b"`. Nie ma możliwości przekształcenia ciągu wynikowego z powrotem do postaci `"haslo"`, nawet przy wykorzystaniu funkcji `sha1()`; na pierwszy rzut oka mogłoby się więc wydawać, że nie jest ona zbyt użyteczna. Właściwością, która decyduje jednak o jej wysokiej przydatności, jest fakt, że jej wynik jest zawsze taki sam. Dla takiego samego ciągu wejściowego funkcja za każdym razem zwróci taką samą wartość.

Zatem zamiast poniższego fragmentu kodu

```
if($uzytkownik=='uzytkownik' && $haslo=='haslo')
{
  //identyfikator i hasło prawidłowe
}
```

lepiej zapisać

```
if($uzytkownik=='uzytkownik' && sha1($haslo)=='
ff12bbd8c907af067070211d87bdf098be17375b')
{
  //identyfikator i hasło prawidłowe
}
```

Nie trzeba znać pierwotnego ciągu znaków, który został przekształcony za pomocą funkcji `sha1()`. Wystarczy tylko, że istnieje możliwość sprawdzenia, czy podane przez użytkownika hasło dostępu ma taką samą postać jak ciąg wejściowy funkcji `sha1()`.

Jak już wspomniano, nie zaleca się zapisywania bezpośrednio w kodzie źródłowym identyfikatorów użytkowników i ich haseł dostępu. Dane te powinny być zapisywane w oddzielnych plikach lub przechowywane w bazie danych.

W przypadku przechowywania tych danych w bazie MySQL można używać zarówno funkcji `sha1()` udostępnianej przez PHP, jak i funkcji MySQL o nazwie `SHA1()`. MySQL udostępnia jeszcze więcej algorytmów mieszających niż PHP, lecz wszystkie one spełniają tę samą rolę.

Chcąc wykorzystać funkcję `SHA1()`, należy przekształcić zapytanie z listingu 16.2 na postać:

```
select count(*) from uwierzytelnieni_uzytkownicy where
      uzytkownik = '$uzytkownik' and
      haslo = sha1('$haslo')
```

Wynikiem tego zapytania jest liczba wierszy z tabeli `uwierzytelnieni_uzytkownicy`, w których wartość pola `uzytkownik` jest taka sama jak wartość zmiennej `$uzytkownik`, a pole `haslo` ma wartość identyczną jak wynik wykonania funkcji `SHA1()`. Ciągiem wejściowym w tej funkcji jest hasło podane przez użytkownika. Ponieważ założyliśmy, że każdy użytkownik ma niepowtarzalny identyfikator, wynikiem powyższego zapytania może być tylko `0` lub `1`.

Należy pamiętać, że funkcje mieszające zwykle zwracają dane o stałym rozmiarze. W przypadku SHA1 dane te mają długość 40 znaków, jeśli funkcja zwraca zwykły ciąg znaków. Należy się upewnić, że kolumna w bazie danych ma wystarczającą szerokość.

Przeglądając ponownie listing 16.3, można zauważyć, że utworzyliśmy jednego użytkownika (`'uzytkownik'`) z hasłem niezaszyfrowanym oraz jednego (`'testowy'`) z hasłem zaszyfrowanym, w celu zobrazowania obu możliwych rozwiązań.

3. ***Zadanie:*** Używając funkcji **sha1()** zaszyfruj ciąg znaków zawierający Twoje imię i nazwisko:
    a. Utwórz zmienną dane
    b. Przypisz do zmiennej dane swoje imię i nazwisko
    c. Utwórz zmienną szyfr1, zaszyfruj wartość zmiennej dane tak by otrzymać szyfr 20 znakowy
    d. Utwórz zmienną szyfr2, zaszyfruj wartość zmiennej dane tak by otrzymać szyfr 40 znakowy
    e. Wyświetl na ekranie obydwa szyfry jeden pod drugim poprzedzone opisem: *„Moje dane zaszyfrowane przy użyciu funkcji sha1():”*
    f. Zgłoś nauczycielowi wykonanie zadania.