Реалізація безпечного транзиту параметрів HTTP протоколу

Самойленко Д.М., Сівко О.
кафедра електрообладнання суден та інформаційної безпеки
Національний університет кораблебудування ім. адмірала Макарова
Миколаїв, Україна
o-stap@hotmail.com

Realization of secure transit of HTTP protocol parameters

D. Samoylenko, O. Sivko

Department of Ships Electrical Equipment and Informational Security
National University of Shipbuilding
Mykolaiv, Ukraine
o-stap@hotmail.com

***Анотація*—Запропоновано програмну реалізацію символьного кодування для забезпечення безпечної передачі інформації з обмеженим доступом за допомогою Інтернету, наприклад, за допомогою параметрів HTTP протоколу.**

***Abstract*—Program realization of symbolic encryption for the secure information translation via Internet with the HTTP protocol parameters is proposed.**

Ключові слова—інформація з обмеженим доступом,безпечне програмування, захищені ресурси.

Keywords—information with limited access, secure programming, guarded resources.

I. Вступ

При розробці захищених інформаційних систем із використанням web-ресурсів необхідно враховувати потенційні атаки на дані, що передаються у відкритому каналі, у даному випадку – через мережу Інтернет. В окремих випадках при розробці серверної частини веб-ресурсу виникає необхідність передачі ІзОД відкритим каналом, наприклад, аутентифікаційних даних, особистої інформації тощо. При цьому зловмисник може реалізовувати атаки на доступність, конфіденційність або ж цілісність інформації, що зберігається на ресурсі.

Сучасні методи передачі ІзОД між клієнтом та сервером включають криптографічні перетворення даних, які зазвичай представляються у текстовому вигляді. У роботі [1] піднімається проблема відмінностей між представленням текстових даних у двох мовах програмування, що використовуються на серверному та клієнтському боці. Найбільш вразливі при цьому є символи української абетки.

II. Програмна реалізація

У роботі [2] пропонується метод та алгоритм, який дозволяє уникнути невідповідностей при перетворенні текстових даних. При цьому програмної реалізації не наведено. Тож метою даної роботи є реалізацій алгоритму у програмному вигляді з урахуванням питань, які не були вирішені у вказаних роботах.

Для розробки функцій було використано серверну мову програмування PHP. Щоб забезпечити максимальну надійність коду, було застосовано техніки безпечного програмування, як-от: відсутність передчасних переривань циклів при переборі інформаційних послідовностей задля унеможливлення атаки за часом; застосування спеціальної криптографічно безпечної функції random_int() для генерації «солі»[3]. В якості шаблону для перемішування обрано матриці із алгоритму DES, які забезпечують рівномірний розподіл символів при перемішуванні[4].

Текст розроблених функцій:

```
$alphabetTable = array(
"А" => 0, "а" => 1,
"Б" => 2, "б" => 3,
"В" => 4, "в" => 5,
"Г" => 6, "г" => 7,…
// таблиця алфавітної
//відповідності
```

```
   function
sequenceMixing($outputSequence)
   {//функція      для      змішування
початкової
   //послідовності символів
   $mi xingMatrix= array(57,  49,  41,
33,  25,  17,  9,  1,…//матриця IP з
   //алгоритму DES; при розшифровці
   //використовується матриця IP-1
   $textBuffer= array();
   //копіруємо до буферу
   foreach($outputSequence as $value)
   {$textBuffer[]= $value;    }
   // «змішуємо» за правилом матриці
   for($i=0;$i<count($mixingMatrix);$i
++)
   {
       $outputSequence[$mixingMatrix[$
i]]= $textBuffer[$i];        }
   return $outputSequence;    }

   function
translate($type,$inputSequence,$alpha
betTable)// «переклад» з
   //текстового формату до числового
та навпаки
   {   $outputSequence= array();
       $tmp=false;
       reset($alphabetTable);
   if(1==$type){//літери -> цифри
   for($i=0;$i<(count($inputSequence))
;$i++)
   {reset($alphabetTable);
   $tmp=false;
   for($j=0;$j<count($alphabetTable);$
j++)
       {$value=current($alphabetTable)
;
   if(key($alphabetTable)==$inputSeque
nce[$i])
   {$outputSequence[]=
$value;$tmp=true;}
   next($alphabetTable);}      }     }
   else      {//цифри -> літери
   for($i=0;$i<(count($inputSequence))
;$i++)
       {reset($alphabetTable);
       $tmp=false;
   for($j=0;  $j<count($alphabetTable);
$j++)
       {$value                        =
current($alphabetTable);
   if($value == $inputSequence[$i])
   {$outputSequence[]=key($alphabetTab
le);
   $tmp=true;      }
   next($alphabetTable);}      }      }

   if(false==$tmp)
   {//повідомлення про помилку}
   return $outputSequence;     }

   function
involutiveTranslation($inputSequence,
$password,$alphabetTable)
   {//інволютивне перетворення
   for($i=0;$i<count($inputSequence);$
i++)
   {$invNum = ( count( $alphabetTable
)- $inputSequence[ $i ] + $password[
$i % count($password)]) %
count($alphabetTable);
   $inputSequence[$i]= $invNum;      }
   return $inputSequence;      }
```

III. Висновок

Сукупність факторів, вказаних вище, робить даний набір функцій придатним до реального використання при розробці захищених інтернет-ресурсів, де необхідно не тільки зберігати, але й передавати ІзОД відкритим каналом.

Література References

1. Самойленко Д. М. Комплексна система захисту інформаційного ресурсу - Інформаційна безпека, 2013. – № 1 (9)
2. Самойленко Д. М. Web-орієнтована система шифрування URI-параметрів - Проблеми кібербезпеки інформаційно-телекомунікаційних систем: Збірник матеріалів доповідей та тез; м. Київ, 23-24 березня 2017 р.; Київський національній університет імені Тараса Шевченка. – К.: ВПЦ «Київський університет», 2017
3. Довідник по PHP. Довідник функцій. Режим доступу: http://php.net/manual/ru/function.random-int.php
4. FIPS Publication 46-3, Data Encryption Standard (DES). Режим доступу: https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf