Cifrar desde PowerShell y descifrar desde PHP

AES

También conocido como Rijndael (pronunciado «Rain Doll» en inglés), es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos, creado en Bélgica. El AES fue anunciado por el Instituto Nacional de Estándares y Tecnología (NIST) como FIPS PUB 197 de los Estados Unidos (FIPS 197) el 26 de noviembre de 2001 después de un proceso de estandarización que duró 5 años. Se transformó en un estándar efectivo el 26 de mayo de 2002. Desde 2006, el AES es uno de los algoritmos más populares usados en criptografía simétrica.

EN 2011 un equipo de investigadores descubrieron la primera vulnerabilidad en el estándar de cifrado AES (Advanced Encryption Standard) que reduce la longitud de la llave efectiva del algoritmo en dos bits. Esto significa que las longitudes usuales de la llave de 128, 192 y 256 bits son reducidas a 126, 190 y 254 bits. Los investigadores emplearon un ataque Meet-in-the-middle, un enfoque que hasta ahora ha sido utilizado principalmente con algoritmo de hashing, combinándolo con un ataque «Biclique». Su método permitió a los investigadores calcular la llave de un simple par de texto plano / texto cifrado, de forma más rápida que lanzando un ataque de fuerza bruta en el espacio de claves completo.

Los algoritmos de cifrado por bloques toman bloques de tamaño fijo del texto en claro y producen un bloque de tamaño fijo de texto cifrado, generalmente del mismo tamaño que la entrada. El tamaño del bloque debe ser lo suficientemente grande como para evitar ataques de texto cifrado. La asignación de bloques de entrada a bloques de salida debe ser uno a uno para hacer

el proceso reversible y parecer aleatoria.

Para aplicar un algoritmo por bloques es necesario descomponer el texto de entrada en bloques de tamaño fijo. Esto se puede hacer de varias maneras:

1. **ECB (Electronic Code Book)**. Se parte el mensaje en bloques de k bits, rellenando el último si es necesario y se encripta cada bloque. Para desencriptar se trocea el texto cifrado en bloques de k bits y se desencripta cada bloque. Este sistema es vulnerable a ataques ya que dos bloques idénticos de la entrada generan el mismo bloque de salida. En la práctica no se utiliza.
2. **CBC (Cipher Block Chaining)**. Este método soluciona el problema del ECB haciendo una o-exclusiva de cada bloque de texto en claro con el bloque anterior cifrado antes de encriptar. Para el primer bloque se usa un vector de inicialización. Este es uno de los esquemas más empleados en la práctica.
3. **OFB (Output Feedback Mode)**. Este sistema emplea la clave de la sesión para crear un bloque pseudoaleatorio grande (pad) que se aplica en o-exclusiva al texto en claro para generar el texto cifrado. Este método tiene la ventaja de que el pad puede ser generado independientemente del texto en claro, lo que incrementa la velocidad de encriptación y desencriptación.
4. **CFB (Cipher Feedback Mode)**. Variante del método anterior para mensajes muy largos.

Cifrar desde PowerShell

```
$unencryptedString = "This is my plain text"

# Cifrar
$bytes                                                    =
[System.Text.Encoding]::UTF8.GetBytes($unencryptedString)

$aesManaged                    =                    New-Object
```

```
"System.Security.Cryptography.AesManaged"
$aesManaged.Mode                                    =
[System.Security.Cryptography.CipherMode]::ECB
$aesManaged.BlockSize = 128
$aesManaged.Key = [System.Text.Encoding]::UTF8.GetBytes('This
is my secre')

$encryptor = $aesManaged.CreateEncryptor()
$encryptedData = $encryptor.TransformFinalBlock($bytes, 0,
$bytes.Length);

$encrypted = [System.Convert]::ToBase64String($encryptedData)

$web                          =                          iwr
("http://localhost:81/fich.php?cifrado="+$encrypted)
$web.content
```

Descifrar desde PHP (http://localhost:81/fich.php)

```
[crayon-6a9d6023eb120408229136/]
```