



# ADVANCED ENCRYPTION STANDARD (AES)

## IMPLEMENTATION SOFTWARE

---

Security System

**USO RESTRINGIDO**

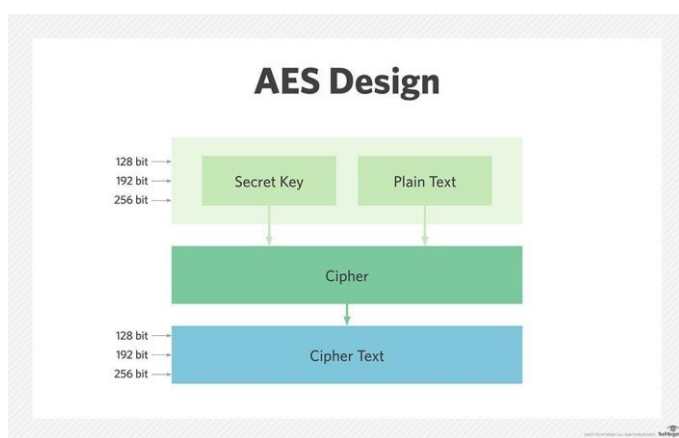
Autor: Yandry Hdez

Cargo: Sysadmin / Programmer

# IMPLEMENTACIÓN ADVANCED ENCRYPTION STANDARD (AES)

## Introducción

Advanced Encryption Standard (AES) es uno de los algoritmos de cifrado más utilizados y seguros actualmente disponibles. Es de acceso público, y es el cifrado que la NSA utiliza para asegurar documentos con la clasificación "**top secret**". Su historia de éxito se inició en 1997, cuando el NIST (Instituto Nacional de Estándares y Tecnología) comenzó oficialmente a buscar un sucesor al envejecimiento cifrado estándar DES. Un algoritmo llamado "Rijndael", desarrollado por los criptografistas belgas Daemen y Rijmen, sobresalía tanto en seguridad como en rendimiento y flexibilidad



Apareció en la cima de varios competidores y se anunció oficialmente el nuevo estándar de cifrado AES en 2001. El algoritmo se basa en varias sustituciones, permutaciones y transformaciones lineales, cada una ejecutada en bloques de datos de 16 bytes - por lo tanto el término blockcipher. Esas operaciones se repiten varias veces, llamadas "rondas". Durante cada ronda, una clave circular única se calcula a partir de la clave de cifrado y se incorpora en los

cálculos. Basado en la estructura de bloques de AES, el cambio de un solo bit, ya sea en la clave, o en el bloque de texto sin cifrado, da como resultado un bloque de texto cifrado completamente diferente - una ventaja clara sobre los cifrados de flujo tradicionales. La diferencia entre AES-128, AES-192 y AES-256 finalmente es la longitud de la clave: 128, 192 o 256 bits - todas las mejoras drásticas en comparación con la clave de 56 bits de DES. A modo de ilustración: El agrietamiento de una clave AES de 128 bits con un superordenador de última generación tomaría más tiempo que la presunta edad del universo. Y Boxcryptor incluso utiliza claves de 256 bits. Hasta el día de hoy, no existe un ataque factible contra AES. Por lo tanto, AES sigue siendo el estándar de cifrado preferido para los gobiernos, bancos y sistemas de alta seguridad en todo el mundo.

### Hablemos un poco de como la matemática juega un papel fundamental

Una encriptación de 256 es lo equivalente a  $2^{256}$  posibilidades de la llave. Es decir  $2^{32}$  es cerca de **4.3 billones**, y sigue incrementando exponencialmente después del 32.

¿Qué significa esto?:

Digamos que hipotéticamente todas las súper computadoras del mundo comienza a realizar un ataque por fuerza bruta por sus siglas en inglés (**brute force attack**) para de esta manera descryptar tu llave de AES-256 para que puedan obtener tu información

Este conjunto de computadoras puede buscar  $2^{50}$  llaves por segundo, el año tiene aproximadamente **31,557,600** segundos. Esto significa que usando un billón de supercomputadoras pueden buscar en  $2^{75}$  llaves por año, a este paso nos tomaría  $2^{34}$  años (**la edad del universo**) en buscar menos del .01% de las posibles combinaciones de la llave. Esto significa que nadie se molestará en descryptar nuestra información.

## IMPLEMENTACIÓN ADVANCED ENCRYPTION STANDARD (AES)

### Ejemplo de desarrollo sobre plataforma PHP

#### Paso numero n°-1- Genramos nuestra Key Criptografica

Crearemos un archivo llamado “aeskey.php”, dicho documento guardara nuestra llave criptográfica, y se estructura de la siguiente manera:

```
//Guardamos la calve privada previamente generada para la encriptación y posterior
Desencriptación de nuestra base de datos
```

```
<?php
$aeskey='A42022420AE5FBDE5F490BE9D6B17048' ;
?>
```

#### Paso numero n°-2 – Llamando nuestra calve y realizando un Insert

Ahora teniendo nuestra llave criptográfica operativa tan solo debemos llamarla cada vez que vallamos a encriptar i/o viceversa

#### Ejemplo de Insert Data en la DB

```
<?php
// Realizamos la llamada previa a nuestra calve criptográfica
include('../cifrado/aeskey.php');

if(isset($_POST['submit']))
{
$Nombre=$_POST['Nombre'];
$UserName=$_POST['UserName'];
$Password=md5($_POST['Password']);
$email=$_POST['email'];
$cargo_ocupado=$_POST['cargo_ocupado'];
$privilegio=$_POST['privilegio'];

$sql="INSERT INTO `admin` (Nombre,UserName,`Password`, `email`,`cargo_ocupado`, `privilegio`)
Value (AES_ENCRYPT(:Nombre, ".$aeskey."),:UserName,:Password,:email,:cargo_ocupado,:privilegio)";

//Como se puede ver en los valores a introducir en el sistema, declaramos el campo que deseamos que quede
//Criptografiado, en este caso lo hemos hecho con el campo Nombre
```

```

$query = $dbh->prepare($sql);
$query->bindParam(':Nombre',$Nombre,PDO::PARAM_STR);
$query->bindParam(':UserName',$UserName,PDO::PARAM_STR);
$query->bindParam(':Password',$Password,PDO::PARAM_STR);
$query->bindParam(':email',$email,PDO::PARAM_STR);
$query->bindParam(':cargo_ocupado',$cargo_ocupado,PDO::PARAM_STR);
$query->bindParam(':privilegio',$privilegio,PDO::PARAM_STR);
$query->execute();
$lastInsertId = $dbh->lastInsertId();
if($lastInsertId)
{
    $msg=" El usuario ha sido registrado con exito";
}
else
{
    $error="A ocurrido algo al registrar el nuevo usuario,compruebe los campos y vuelva a intentarlo";
}







}
?>

```

#### Resultado de los datos insertados en la base de datos

☐ Mostrar todo | Número de filas: 25 ▼ Filtrar filas:

+ Opciones



				id	Nombre	UserName	Password
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	7	7	JOSE	4f416421a0d6f0
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	6	6	MIGEU	e0469f664b4568

### Paso numero n°-3 – Realizamos un select y desencriptamos los campos encriptados

```
<?php
//Llamamos nuestro archivo con la clave criptográfica para el proceso de descifrado de la información
include('../..//cifrado/aeskey.php');

$sql = "SELECT *, Nombre, (AES_DECRYPT(Nombre,'" . $aeskey . "')) AS Nombre FROM admin;";
$query = $dbh->prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
    foreach($results as $result)
    {
        ?>
        <tr>
        <td><b><strong><?php print($result->id);?></strong></b></td>
        <td><b><strong> <?php print($result->UserName);?></strong></b></td>
        <td><b><strong> <?php print($result->Nombre);?> </strong></b></td>
        <td><b><strong> <?php print($result->email);?> </strong></b></td>
        <td><b><strong> <?php print($result->telefono);?> </strong></b></td>
        <td><b><strong> <?php print($result->cargo_ocupado);?> </strong></b></td>
        <td><b><strong> <?php print($result->privilegio);?> </strong></b></td>
        </tr>
        <?php $cnt=$cnt+1;}} ?>
```

### Resultado de nuestro select, habiendo previamente desencriptados los datos

Nº	Usuario	Nombre	Email	Telefono
6	 MIGEU	MIGUEL ANGEL	miguel@demo.com	
7	 JOSE	JOSE MIGUEL FERNANDEZ GALLO	jose@gmail.com	

Como podemos apreciar el campo nombre en nuestra base de datos contiene los caracteres cifrado, pero al usuario final se le muestran los datos descifrados. Ejemplo:

Dato cifrado	Dato Descifrado
†• ~=-.bÖøYæ@'¿Äï8mFù¤ * jL¥€ÿ	<b>JOSE MIGUEL FERNANDEZ GALLO</b>

Paso numero nº-4 – Realizamos un Update descriptando y volviendo a encriptar los datos.

```
<?php
//Llamamos nuestro archivo con la calve criptográfica para el proceso de descifrado y posterior cifrado de la
información cuando la actualicemos
Include ('../..//cifrado/aeskey.php') ;

?>
```