Funcionalidades en PHP para cifrado de datos

1. password_hash() y password_verify()

Descripción: Estas funciones se utilizan principalmente para cifrar y verificar contraseñas. Utilizan un algoritmo de cifrado moderno y generan automáticamente una "sal" para el cifrado, lo que las hace seguras y fáciles de usar.

Ejemplo:

```
$password = "miContraseña123";
$hash = password_hash($password, PASSWORD_DEFAULT);

if (password_verify($password, $hash)) {
    echo "¡Contraseña verificada!";
} else {
    echo "Contraseña incorrecta";
}
```

2. mcrypt_encrypt() y mcrypt_decrypt()

Descripción: Aunque la extensión mcrypt ha sido declarada obsoleta en PHP 7.1.0 y eliminada en PHP 7.2.0, se usó ampliamente en versiones anteriores de PHP para cifrado y descifrado.

Ejemplo:

```
// NOTA: ¡mcrypt está obsoleto en versiones modernas de PHP!

$key = "clavesecreta";
$input = "Texto a cifrar";

$ciphertext = mcrypt_encrypt(MCRYPT_RIJNDAEL_256, $key, $input,
MCRYPT_MODE_ECB);
$decryptedtext = mcrypt_decrypt(MCRYPT_RIJNDAEL_256, $key, $ciphertext,
MCRYPT_MODE_ECB);

echo "Texto cifrado: " . $ciphertext;
echo "Texto descifrado: " . $decryptedtext;
```

3. OpenSSL

Descripción: La extensión OpenSSL proporciona fuertes capacidades de cifrado utilizando la biblioteca OpenSSL. Es recomendable usar OpenSSL sobre mcrypt en las versiones modernas de PHP.

Ejemplo:

```
$key = "clavesecreta";
$plaintext = "Texto a cifrar";

$ciphertext = openssl_encrypt($plaintext, "aes-256-cbc", $key, 0, $key);
$originaltext = openssl_decrypt($ciphertext, "aes-256-cbc", $key, 0,
$key);

echo "Texto cifrado: " . $ciphertext;
echo "Texto descifrado: " . $originaltext;
```

4. hash()

Descripción: Esta función se utiliza para generar hashes de un string utilizando un algoritmo específico como "sha256" o "md5" (aunque MD5 ya no se recomienda debido a vulnerabilidades).

Ejemplo:

```
$input = "Texto a hashear";
$hash = hash("sha256", $input);

echo "Hash SHA-256: " . $hash;
```

Es crucial entender que mientras `hash()` y `md5()` generan un hash irreversible (es decir, no puedes obtener el texto original a partir del hash), funciones como `password_hash()`, `mcrypt_encrypt()` y las funciones de OpenSSL proporcionan cifrado reversible (es decir, puedes obtener el texto original descifrándolo con la clave adecuada). Es esencial usar el tipo correcto de cifrado o hash según el caso de uso. Por ejemplo, para almacenar contraseñas, siempre se debe usar un hash irreversible y nunca cifrado reversible.