$config = array(  
"digest\_alg" => "sha256",  
"private\_key\_bits" => 2048,  
"private\_key\_type" => OPENSSL\_KEYTYPE\_RSA  
);  
$keypair = openssl\_pkey\_new($config);  
openssl\_pkey\_export($keypair, $private\_key);  
$public\_key = openssl\_pkey\_get\_details($keypair);  
$public\_key = $public\_key["key"];

// Cifrar el mensaje  
$message = "Mensaje secreto";  
openssl\_public\_encrypt($message, $ciphertext, $public\_key);

// Descifrar el mensaje  
openssl\_private\_decrypt($ciphertext, $plaintext, $private\_key);

// Imprimir el mensaje original y el mensaje descifrado  
echo "Mensaje original: " . $message . "\n";  
echo "Mensaje descifrado: " . $plaintext . "\n";

Este código implementa el algoritmo RSA en PHP utilizando la extensión OpenSSL. A continuación, te explico cada parte del código:

1. En la primera línea, se crea un array $config que contiene los parámetros utilizados para generar el par de claves RSA. Los parámetros incluyen el algoritmo de hash utilizado (sha256), el tamaño de la clave privada (2048 bits) y el tipo de clave (RSA).
2. En la segunda línea, se utiliza la función openssl\_pkey\_new para generar un nuevo par de claves RSA con la configuración especificada en el array $config. El resultado se almacena en la variable $keypair.
3. En la tercera línea, se utiliza la función openssl\_pkey\_export para exportar la clave privada en formato PEM y almacenarla en la variable $private\_key.
4. En la cuarta línea, se utiliza la función openssl\_pkey\_get\_details para obtener los detalles de la clave pública a partir del par de claves generado anteriormente. Los detalles se almacenan en la variable $public\_key.
5. En la quinta línea, se extrae la clave pública de la variable $public\_key y se almacena en la misma variable.
6. En la sexta línea, se define el mensaje que se desea cifrar en la variable $message.
7. En la séptima línea, se utiliza la función openssl\_public\_encrypt para cifrar el mensaje utilizando la clave pública de RSA contenida en la variable $public\_key. El mensaje cifrado se almacena en la variable $ciphertext.
8. En la octava línea, se utiliza la función openssl\_private\_decrypt para descifrar el mensaje cifrado utilizando la clave privada de RSA contenida en la variable $private\_key. El mensaje descifrado se almacena en la variable $plaintext.
9. Finalmente, en las últimas dos líneas, se imprimen el mensaje original y el mensaje descifrado.

Es importante destacar que este código es sólo un ejemplo básico y no debe ser utilizado en producción sin tener en cuenta otros aspectos de seguridad, como la gestión de claves, la autenticación y la integridad de los datos. Además, el cifrado RSA es relativamente lento en comparación con otros algoritmos de cifrado, por lo que se utiliza principalmente para cifrar claves de cifrado simétrico o pequeñas cantidades de datos.