EncriptadorDP: Manual Técnico

# 1. Introducción

EncriptadorDP es un sistema de encriptación diseñado para integrar cifrado seguro en proyectos de VBA mediante el uso de algoritmos RSA y AES.  
  
• RSA se utiliza para generar y proteger la clave AES mediante un par de claves pública y privada.  
• AES se utiliza para encriptar y desencriptar textos o contraseñas dentro del sistema.

# 2. Generación de proyectos

• GenerarRSA: proyecto de consola para generar el par de claves RSA (privateKey.xml y publicKey.xml).  
• GenerarAES: proyecto de consola para generar la clave AES cifrada (aesKey.enc) utilizando la clave pública RSA.  
• CryptoLib: biblioteca DLL en C# que permite usar estas claves para encriptar y desencriptar desde VBA.

# 3. Generación de claves

## 3.1 Generación de la clave RSA

1. Abrir el proyecto GenerarRSA en Visual Studio.  
2. Ejecutar el proyecto para generar los archivos privateKey.xml y publicKey.xml.  
3. Guardar ambos archivos en un lugar seguro (privateKey.xml nunca debe compartirse).

## 3.2 Generación de la clave AES

1. Abrir el proyecto GenerarAES y copiar el archivo publicKey.xml en la carpeta bin/Debug del proyecto.  
2. Ejecutar el proyecto para generar el archivo aesKey.enc.  
3. Copiar aesKey.enc a la carpeta donde estará la DLL final.

# 4. Compilación y registro de la DLL

1. Abrir el proyecto CryptoLib en Visual Studio.  
2. Pegar dentro del código el contenido de privateKey.xml en el apartado indicado.  
3. Compilar en modo Release para generar CryptoLib.dll y CryptoLib.tlb.  
4. Copiar CryptoLib.dll, CryptoLib.tlb y aesKey.enc a una carpeta final.  
5. Registrar la DLL en el sistema utilizando el archivo RegistrarDLL.bat con privilegios de administrador.

# 5. Uso desde VBA

1. En VBA, crear un objeto con CreateObject("CryptoLib.CryptoHelper").  
2. Utilizar los métodos EncryptAES(texto) y DecryptAES(texto) para encriptar o desencriptar cadenas.  
3. No es necesario establecer manualmente la clave ni el IV, ya que son cargados automáticamente.

# 6. Seguridad

• privateKey.xml nunca debe ser distribuido ni compartido.  
• aesKey.enc debe protegerse y no debe ser editable por usuarios no autorizados.  
• Si aesKey.enc o la DLL son comprometidos, se debe generar un nuevo par RSA y una nueva clave AES.  
• Este método protege la clave AES al cifrarla con RSA, evitando que se almacene en texto plano.

# 7. Mantenimiento futuro

• Si se requiere cambiar la clave, basta con generar un nuevo par de claves RSA y un nuevo aesKey.enc.  
• Se recompila CryptoLib con la nueva clave privada y se reemplaza la DLL en el sistema.  
• Documentar cada cambio para futuras actualizaciones.