

Ejercicios de seguridad: realizar una comunicación TCP segura (utilizando Cryptographic Message Syntax)

El Cryptographic Message Syntax (CMS) es un estándar de la IETF para proteger mensajes criptográficamente. En él se definen las sintaxis para firmar digitalmente, realizar hash, realizar mac o para cifrar cualquier tipo de dato. CMS se basa en la sintaxis de PKCS#7, que a su vez se basa en el estándar Privacy-Enhanced Mail. La versión más moderna de CMS se define en la RFC 5652 (ver también la RFC 5911 de módulos ASN.1 acordes al estándar ASN.1 2002).

CMS se construye alrededor del sistema de manejo de claves certificate-based así como el perfil definido por PKIX working group.

CMS se usa como el componente clave para muchos otros estándares criptográficos como S/MIME, PKCS#12 y el protocolo de sellado digital (RFC 3161).

OpenSSL es un software de código libre que permite cifrar, descifrar, firmar, verificar, comprimir y descomprimir documentos acordes a este estándar.

```
##Client
$TcpClient = New-Object System.Net.Sockets.TcpClient([IPAddress]"127.0.0.1", "2050")
$Stream = New-Object System.IO.StreamWriter $TcpClient.GetStream()
# Enviar mensaje cifrado
# Crear un certificado para firmar el mensaje
# https://www.jesusnino.com/11/18/exports-a-certificate-to-a-personal-information-exchange
$Cert = New-SelfSignedCertificate -DnsName jesusnino3445122 -CertStoreLocation "Cert:\CurrentUser\My"
# Encrypts content by using the Cryptographic Message Syntax format
$hola = "hola"
$Sval = [string](Protect-CmsMessage -To cn=jesusnino3445122 -OutFile secret.txt)
$Stream.Write($Sval)
$Stream.Dispose()

# Enviar mensaje cifrado
# Crear un certificado para firmar el mensaje
# https://www.jesusnino.com/11/18/exports-a-certificate-to-a-personal-information-exchange
$Cert = New-SelfSignedCertificate -DnsName jesusnino3445122 -CertStoreLocation "Cert:\CurrentUser\My"
# Encrypts content by using the Cryptographic Message Syntax format
$hola = "hola"
$Sval = [string](Protect-CmsMessage -To cn=jesusnino3445122 -OutFile secret.txt)
$Stream.Write($Sval)
$Stream.Dispose()
PS C:\Users\juan>
```

```
##Server
$TcpListener = New-Object System.Net.Sockets.TcpListener (New-Object System.Net.IPEndPoint([IPAddress]"127.0.0.1", 2050))
$TcpListener.Start()
$Stream = (New-Object System.IO.StreamReader ($TcpListener.AcceptTcpClient().GetStream())).Read()
# Para descifrar el mensaje es necesario tener el certificado
# https://www.jesusnino.com/11/20/imports-certificates-and-private-keys-from-a-personal-information-store
$Stream > cifradoparadescifrar.txt
$mensaje = Unprotect-CmsMessage -Path cifradoparadescifrar.txt
"Texto descifrado: $mensaje"
$TcpListener.Stop()
```

PS C:\Users\juan>

Servidor

[crayon-6a9d57db70b29257380492/]

Cliente

[crayon-6a9d57db70b2f466378101/]