**­­Working with Encryption in PowerShell**

By Sam Cummings

**Overview**

This tutorial will outline one way to encrypt and decrypt messages in PowerShell. There is a challenge at the end of this tutorial that requires the practice of encrypting and decrypting messages.

**Prerequisites**

* PowerShell
* Any IDE of choice (VSCode is used in the examples)
* Basic programming knowledge

**Prepare and Learn**

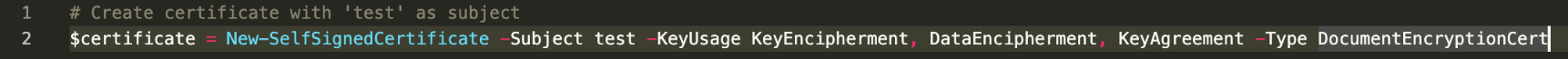
We will use the *Protect-CmsMessage* cmdlet from the *Microsoft.Powershell.Security* module to encrypt messages.

* Open the [Protect-CmsMessage](https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.security/protect-cmsmessage?view=powershell-7.3) documentation.
* Notice that in the description it mentions that “the Protect-CmsMessage cmdlet encrypts content by using the Cryptographic Message Syntax (CMS) format.” Here is more information on the CMS format if you want more of a background of how it works: [Cryptographic Message Syntax](https://en.wikipedia.org/wiki/Cryptographic_Message_Syntax). For this tutorial all you need to know is that CMS encryption uses [Public-key cryptography](https://en.wikipedia.org/wiki/Public-key_cryptography).
* Like the documentation states, there must be an encryption certificate set up before running the *Protect-CmsMessage.*

**Create an Encryption Certificate**

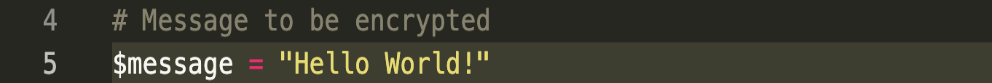
The [*New-SelfSignedCertificate*](https://learn.microsoft.com/en-us/powershell/module/pki/new-selfsignedcertificate?view=windowsserver2022-ps)cmdlet will help us create an encryption certificate.

Create a *New-SelfSignedCertificate* object:

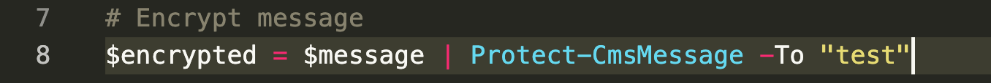


**Encrypt Message**

* Create a message of your choice:



* Encrypt the $message using *Protect-CmsMessage*

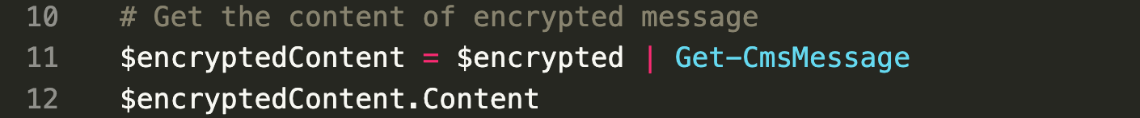


\*Note: The -To param for *Protect-CmsMessage* is the subject of the certificate that we created earlier (test)

That’s it, that’s all you have to do to encrypt a message in PowerShell.

**View Encrypted Message Content**

To view the content of the $encrypted variable we just created we must use the [*Get-CmsMessage*](https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.security/get-cmsmessage?view=powershell-7.3)cmdlet:



**Decrypt Message**

To decrypt the message we use the [*Unprotect-CmsMessage*](https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.security/unprotect-cmsmessage?view=powershell-7.3) cmdlet:



**Challenge**

Now that you know how to encrypt and decrypt data, complete the following task to help practice what you’ve learned. The task is to create a PowerShell script that allows users to send and receive messages from a mock database.

Requirements:

1. Create a Json file named *messages.json.* This is our “mock database” which will contain all of the messages users add. The json file should be a list of objects where each object should have the following two attributes:

Text

Description automatically generated

\*Note that the message is encrypted\*

1. Simple menu display of what the user can do. Options of what they can do should be similar to the following:
   1. Send a message (s)
   2. Get latest message (l)
   3. View all messages (a)
2. Gets a message from the user to save to the “database” and encrypts that message before adding it to the *messages.json* file.
3. Gets a message (or more) from the “database”, decrypts it, and displays the decrypted content.

There are no restrictions to this project other than making sure that the messages stored in the *messages.json* file are encrypted. Remember what we’ve learned about [*Protect-CmsMessage*](https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.security/protect-cmsmessage?view=powershell-7.3)*,* [*New-SelfSignedCertificate*](https://learn.microsoft.com/en-us/powershell/module/pki/new-selfsignedcertificate?view=windowsserver2022-ps)*,* [*Get-CmsMessage*](https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.security/get-cmsmessage?view=powershell-7.3), and [*Unprotect-CmsMessage*](https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.security/unprotect-cmsmessage?view=powershell-7.3)!