# Sure I can protect data with CryptProtectData, but how do I remove the ability to decrypt it?

devblogs.microsoft.com/oldnewthing/20180718-00

July 18, 2018

Raymond Chen

A customer was using the `CryptProtectData` function to protect some information, and they used the corresponding function `CryptUnprotectData` to decrypt the buffer and recover the information. But they wanted to know how to render the protected information un-decryptable when their program is uninstalled.

The decryption key is tied to the user, so there is no way to revoke it. But what you can do is put something in the entropy (nonce). In order to decrypt the data, the caller must be running as the correct user, and the caller must be able to produce the entropy.

For example, you might generate a random number when the program is installed and save that random number somewhere. Whenver you need to encrypt or decrypt data, you combine that random number with whatever entropy source you would normally have used.

Of course, you didn't really "protect" the data since the random number had to be saved somewhere, and the user could fish it out of wherever you saved it. Even if you saved the nonce off the machine (say, on a Web server), or if the nonce were systematically generated from other data, the user can still fish it out: They could hook the `CryptUnprotectData` function and see what you passed as the entropy.

Mind you, the user could also simply hook the `CryptUnprotectData` function and capture the unprotected data! Once that's done, it doesn't matter what you do: You can uninstall the app, intentionally corrupt the data, or delete it outright. The cat is already out of the bag: You decrypted it in a place that the user has access to.

Mind you, you already lost even before this point, because the user could have hooked `CryptProtectData` and captured the unprotected data before it got encrypted in the first place.

Basically, this is a pointless effort. Even if you can make the information to become unrecoverable after the program uninstalls, the user could simply capture the data before uninstalling your program. The only way to keep this from happening is never to let the user

see the unprotected data in the first place.

Raymond Chen

**Follow**