

## Re: Copy protection for a .NET application

**Source:** <http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2004-12/0182.html>

---

**From:** William Stacey [MVP] ([staceywREMOVE\\_at\\_mvps.org](mailto:staceywREMOVE_at_mvps.org))

**Date:** 12/02/04

Date: Thu, 2 Dec 2004 17:13:57 -0500

I read the docs. They talk about things like "Protective Shields" and "Train Cars" a lot. But when you wade past the marketing fluff, it is an RSA key store plain and simple. It stores private keys and can take an encrypted string (encrypted with public key) and decrypt it. Here is the basic process from what I read:

- 1) Encrypt exe, or dll with 128-bit Rijndael using random Secret and IV and store. Done at developers station.
- 2) Encrypt the secret and IV with the public key. Save the cipher text and public key with the cipher dll from Step 1.
- 3) One and 2 are the "package". Other transforms and/or "magic" splitting/blocking/train-car may be done.
- 4) Create the wrapper/loader around the cipher "package" and distribute that.

When app starts:

- 1) Loader \*.exe that is called by user.
- 2) Loader first checks that the HardLock(HL) is present with a ping msg. If not, return failure.
- 3) Loader loads "package" and extracts cipher Secret and IV and sends to HL.
- 4) HL decrypts the string(s) using private RSA key stored in the lock.
- 5) HL returns \*clear secret and IV to caller. This is the potential weak link in the procedure.
- 6) Loader decrypts "package" in memory using Rijndael keys.
- 7) Loader can call an entry point to start the app.
- 8) Loader may also use a timer thread to periodically check the HL is always there with some ping msg – or exit if not.
- 9) Loader has some magic to tell if a debugger is running. Not sure about validity of this statement or what is possible here.

So basically it uses Rijndael and RSA key pairs to encrypt the Rijndael keys like you would do in normal digital envelope. But the RSA private key is stored in the lock. Like all crypto, it comes down to protecting the keys and to use the resulting clear text for only a very short time and remove it from memory. Here is where it comes back to the same problem we all have – protecting the clear text in memory. The driver api call to decrypt the rj keys \*has to return clear text to init Rijndael with – just has to be done. If I can get the key, the HL is not much use anymore as I can run Rijndael myself with the known key and iv. The various "mixing" they also do would

slow you down a bit more. The "crashing debugger" part is the most interesting I think. If they can do that in a fool proof way, then they really have something there. But they don't talk about the tech here other than marking points. I don't know enough about it to say either way. If they can't stop debuggers, then it would not be that much harder to crack than a sw only solution.

Actually I have done all the same things in a sw-only license solution with digital envelopes and ran into same issues. It all comes down to protecting that darn key – the rest is not so important. Wonder if a USB sniffer would show the returned clear text rj keys? Cheers.

```
--
William Stacey, MVP
http://mvp.support.microsoft.com
"C-Services Holland b.v." <cs@REMOVEcsh4u.nl> wrote in message
news:qPednUd2kqC5pzLcRVnyig@zeelandnet.nl...
> William Stacey [MVP] wrote:
> >>The problem with decrypting the code yourself is that you have a readily
> >>accessible decryption module in your software to decrypt the DLL. That
> >>would make it a breeze for any decent cracker to break your protection.
> >
> >
> > True. But you have equal access to the either code in memory using a
> > debugger. So it does raise the bar, but not too high for crackers.
They
> > use debugger anyway (after ildasm fails), so it is not much difference
for
> > them.
> >
> >
> > Not really. The decrypting part is in the hardlock, unaccessable by
> > debuggers :) From the tech docs I've read, they also employ
> > anti-debugging techniques to prevent on the fly inspection of things
> > going on.
> >
> > But as I said. Nothing is impossible to crack, it's just how much time
> > and effort is one willing to spend.
> >
> --
> Rinze van Huizen
> C-Services Holland b.v.
```