

Re: How good an encryption algorithm is this?

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2004-11/1037.html>

From: Ian Griffiths [C# MVP] (*ian-interact-sw_at_nospam.nospam*)

Date: 11/25/04

Date: Thu, 25 Nov 2004 16:07:47 -0000

>> *What threats do you envision and are trying to protect against?*
>
> *Here goes again, then:*
> *Someone being able to produce a "crack program" that enables the layman to*
> *find out his colleague's SQL server password by running the "crack*
> *program"*
> *on his computer when he's not looking, the crack program working by*
> *reading*
> *my application's registry key.*

Given the set of requirements you have outlined, you will never be able to protect against this attack. Only by restricting physical access to the machine could you make such a crack impossible. If a skilled attacker gets unrestricted physical access to the machine when the relevant user is logged on, then there is no way on earth you can keep a secret. (Unless you require the user to type in a password or something whenever the data needs to be retrieved, which is contrary to what you're trying to achieve here.)

The fact is, that if your program is able to recover the full credentials, then it will be technically possible to write a program that also recovers these same full credentials.

The best you can do is make it hard for such a program to work. DPAPI does a good job of this. One of the benefits it offers is that an attacker would not only need physical access to the machine, she would also need to be able to run this 'crack program' while logged in under her colleague's user account.

You can also make life **much** harder for the attacker if it's acceptable for your program to ask the user for a password onces when they first start it. (I'm not sure from your description whether that's acceptable or not. I'm not sure if you're just trying to avoid asking the user for passwords time and time again, or whether you're trying to avoid asking them for a password at all.) If it is acceptable, you can use this password to add extra protection with DPAPI – it would mean that even if the attacker had physical access to the user's machine while logged on as the user, they would still need this last password before their crack program would run.

microsoft.public.vc.language: Re: How good an encryption algorithm is this?

This means that the cracker somehow needs to get the password. (That's still not so hard given the power you think your attacker will have. Two obvious attacks are (1) install a keystroke logger, or (2) take a snapshot of your program using NTSD and then trawl through it to find the password...)

Of course there's a much more low-tech attack than the 'crack program' approach which you're not going to be able to prevent: the attacker walks up to their colleague's machine and just uses this program of yours...

```
--
Ian Griffiths - http://www.interact-sw.co.uk/ianqblog/
DevelopMentor - http://www.develop.com/
"Bonj" <Bonj@discussions.microsoft.com> wrote in message
news:B7259926-5995-4B96-91A6-3BBB84E7B84B@microsoft.com...
>> Let's take yet another step back: _why_ do you need the key constantly
>> persisted in software on the client?
>
> Because it's a "password remember" feature for a windows app, to enable
> the
> application to logon to SQL servers that don't use windows authentication,
> so
> that the user of the PC doesn't have to constantly type in the password
> every
> time he uses it.
> It's nothing to do with sending an encrypted message down a wire and
> decrypting it again at the other end, and I don't control the part of the
> software that decides whether or not the password is valid, it must be
> enclosed in the connection string.
>
>> What do you need this key for
>
> To log on to SQL server
>
>> , what
>> data are you trying to protect
>
> The connection string
>
>> , who needs access to this data
>
> Only the person that stored it (the application stores it when they type
> it
> in for the first time)
>
>> , how do
>> you plan to know she is the right person to have access to said data?
>
> Again, please take on board that I don't have control of the part of the
> software that decides whether the password is valid. That is SQL server.
>
>> What threats do you envision and are trying to protect against?
>
> Here goes again, then:
> Someone being able to produce a "crack program" that enables the layman to
> find out his colleague's SQL server password by running the "crack
> program"
> on his computer when he's not looking, the crack program working by
> reading
> my application's registry key.
```

Re: How good an encryption algorithm is this?

microsoft.public.vc.language: Re: How good an encryption algorithm is this?

```
>
>>
>> I highly recommend "Writing Secure Code" by Michael Howard et al [1]. It
>> has a chapter on storing secrets securely on Windows machine. All of
>> them essentially boil down to trusting Windows authentication: if the
>> person managed to log into her Windows account, you assume that she is
>> indeed who she says she is.
>
> I think CryptProtectData will enable me to do this.
>
>
>>
>> [1] http://www.amazon.com/exec/obidos/tq/detail/-/0735617228
>> --
>> With best wishes,
>>     Igor Tandetnik
>>
>> With sufficient thrust, pigs fly just fine. However, this is not
>> necessarily a good idea. It is hard to be sure where they are going to
>> land, and it could be dangerous sitting under them as they fly
>> overhead. -- RFC 1925
>>
>>
>>
```