

Re: FileCopy overwrites the existing file

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.kernel/2006-12/msg00312.htm>

- *From:* Grzegorz Wróbel </dev/null@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 11 Dec 2006 06:30:39 +0100
-

anton bassov wrote:

Grzegorz,

I would advise you to look through "Applied Cryptography", 2nd Edition by Schneier – as far as I remember, he provides some examples of seemingly perfect symmetric algorithms being broken.....

I'm not familiar with the book, but examples of some symmetric algorithms being broken doesn't mean every symmetric algorithm is breakable. That how algorithms were broken in early ages of cryptography. As for smart method of compromising cryptosystems, there are pretty fine examples of successful attacking asymmetric ones as well.

No matter how you look at it, large target data set will always remain a potential risk to the symmetric algorithms – even if you minimize "plain text risk factor"

There is no such thing like "large data sets" with well designed algorithm. You do realize that hundreds of gigabytes, terabytes or petabytes are nothing compared to the number like 2^{2000} , don't you?

I can implement symmetric algorithm which not only will be unbreakable but the encrypted code will be undistinguishable from random data (with limited time and resources of course, but I can easily push the time limit required for that far beyond, say the estimated time of existence of our universe, even if one had unattainable computing power).

I cannot say the same about any asymmetric algorithm and no one can.

Anton Bassov

Grzegorz Wróbel wrote:

Re: FileCopy overwrites the existing file

anton bassov wrote:

> Here we speak about the data samples of the size of *hundreds of GB*

with some "plain-text" (i.e. OS-related stuff) known in advance, so that symmetric algorithms that are perfectly safe for encrypting some files or messages may be not-so-reliable here – probably, you would have to go for asymmetric ones, and, hence, pay performance penalties

The problem of "plain-text" messages and risk of "finding patterns" is usually solved (at least I'll do it this way) by mixing the plain text with values returned by good pseudo-random number generator (say 256bit seed + at least 2^{1000} period). Distinguish an output of pseudo-random generator from random output is not doable in polynomial time so you are safe with this technique, and mixed plain-text with such pseudo-random sequence is as chaotic as such sequence itself.

I consider symmetric algorithms much safer than asymmetric ones (there is no proof there aren't breakable in polynomial time, no one just did it) and there are quite advanced factoring algorithms (like GNFS) that pushes the limit of breakable keys further and further (1024bit RSA encryption in few years won't be considered that much secure).

The asymmetric algorithms have wider range of usage, but for disk compression nothing more than symmetric encryption is needed.

--

Grzegorz Wróbel

<http://www.4neurons.com/>

677265676F727940346E6575726F6E732E636F6D

--

Grzegorz Wróbel

<http://www.4neurons.com/>

677265676F727940346E6575726F6E732E636F6D

.