

Re: TripleDES output size

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.security/2006-10/msg00041.html>

- *From:* "Joe Kaplan" <joseph.e.kaplan@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 10 Oct 2006 10:21:02 -0500
-

It is probably a good idea to use either Unicode or UTF-8 if you may need to support non-ASCII characters, as you'll have much better luck getting the data back into the right format when you decrypt if you just use Unicode consistently. That's what it is there for.

You could save some actual size in some cases by using UTF-8, but since it is variable length, it will be harder to predict how big the encrypted data will be. Using Unicode, if your strings have a maximum length, your size will be very predictable.

You also have the option of not storing the data in SQL as string but as binary instead. That will save you 1/3 with the overhead generated by base64, but you'll have to be a little more careful getting the data in and out of the database. Base64 is easiest.

I'd suggest trying not to worry about it too much and just making your SQL columns be the size they need to be. :)

Joe K.

--

Joe Kaplan-MS MVP Directory Services Programming
Co-author of "The .NET Developer's Guide to Directory Services Programming"
<http://www.directoryprogramming.net>

--

"Arturo Buonanni" <leave_this_out_deer.chief.this.also@xxxxxxxx> wrote in message news:91imi2t86kakj0ompqk9nevoh41o2i0h1m@xxxxxxxx

On Mon, 9 Oct 2006 12:15:07 -0500, "Joe Kaplan" <joseph.e.kaplan@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

It depends on how you are converting your string data to binary (what encoding) and how you are converting your binary cipher data back to a string.

If you are using UTF-8 for converting string to binary (generally a good idea), then the downside there is that the binary data can be variable length, depending on the characters in the string. If you use Unicode

Re: TripleDES output size

encoding, then you should always get two bytes per character.

Then, you can figure out how many bytes of cipher data you'll get by rounding up to the block size.

If you convert from your binary data to string with Base64 (which is probably the best way to go, as it is smaller than hex string and won't result in lossy conversion), then the base64 string will be 4/3 the size of the binary data.

Make sense?

It makes perfect sense. Thanks.

I indeed use unicode encoding to feed the encryption provider and then convert the binary data to string with Base64. It turns out to be:

```
CipherTextLength = _  
(((Trunc((PlainTextLength * 2) \ 8) + 1) * 8) \ 3) + 1) * 4
```

The problem is that only now I realize I'm making my strings quite big...:-(

Anyway, as I need to use Italian character set, I think I'm forced to use at least UTF-8. In that case I should however consider the worst case in which every character of my string is a 2 byte code so I end up having field the same size as using Unicode. :-(

Are you just trying to decide how big to make your columns in SQL given input strings of a certain length?

That's exactly what I'm trying to do.
Thanks for your help.

Joe K.

--

Joe Kaplan—MS MVP Directory Services Programming
Co-author of "The .NET Developer's Guide to Directory Services
Programming"
<http://www.directoryprogramming.net>