

Re: SQL 2005 – Searching Encrypted SSN

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.sqlserver.security/2006-06/msg00224.html>

- *From:* "Mike C#" <xyz@xxxxxxx>
 - *Date:* Wed, 28 Jun 2006 16:35:51 -0400
-

Yeah I heard there were a couple of bugs in the implementation over there. Can we expect a fix for that? Also are they planning on allowing users to specify their own salt/IV values for encryption in the future?

"Laurentiu Cristofor [MSFT]" <laur@xxxxxxxxxx> wrote in message <news:OOSv3OtmGHA.3600@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Please also note that RC4 use is not recommended. Do not use RC4 just because it's unsalted and allows you to index the encryption for equality searches. While it may be convenient, it is not secure.

Thanks

—

Laurentiu Cristofor [MSFT]
Software Design Engineer
SQL Server Engine
<http://blogs.msdn.com/lcris/>

This posting is provided "AS IS" with no warranties, and confers no rights.

"Mike C#" <xyz@xxxxxxx> wrote in message <news:uaH7JLgmGHA.4100@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

That was the kicker. Unless you can specify your own salt and IV (and I don't have SQL 2K5 installed here at work, so I can't test it), you'll either:

- 1) have to perform decryption on every column and compare,
- 2) write your own encryption/decryption functions (not too difficult with a SQLCLR hosted .NET System.Security.Cryptography namespace), or 3) Store a hash of the encrypted data. If you only need the SSN for identification purposes, and not for reporting purposes, you could probably hash the SSN and forget about encryption altogether.

BTW, according to BOL, SQL 2005 does not salt the encryption performed with RC4 and RC4_128.

Re: SQL 2005 – Searching Encrypted SSN

"Ron Brent" <RonBrent@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
news:82A6C044-195A-403A-8BD5-E9CF2F163035@xxxxxxxxxxxxxxxxxxxx

Hi Mike,

According to Laurentiu's blog –
<http://blogs.msdn.com/lcris/archive/2005/12/22/506931.aspx>

–
"The encryption algorithms in SQL Server 2005 are salted.
By salting, I
mean
that the encryption algorithms are always using a random
initialization
vector (IV), which leads to the following property:
encrypting twice the
same
piece of data using the same key will produce two different
ciphertexts."

You say that theoretically I should be able to encrypt the
SSN I'm
searching
for and compare that to the already-encrypted columns.
How would the second encryption (that is done within the
query) produce
identical cipher text results to the first encryption given the
fact
that the
encryption algorithm uses a random initialization vector?

Thanks,

Ron

"Mike C#" wrote:

"Ron Brent"
<RonBrent@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote in message
news:1929F115-F507-4448-9AD7-1F900B9A1F3F@xxxxxxxxxxxxxxxxxxxx

Thanks Uri!

It will be helpful once we
decide to implement the
encryption, but my
question is this –
is it possible to search an

Re: SQL 2005 – Searching Encrypted SSN

encrypted column the way
it's described
in the
blog and get the results
quickly (at the same speed it
takes to
search on
a
clear text)?

If you want to search using = in the WHERE
clause, you can get
comparable
speed. Just don't decrypt the SSN's in the
column. Theoretically you
should be able to encrypt the SSN you're
searching for and compare that
to
the already-encrypted columns. Of course
this won't work if your
column
looks like this:

999-99-9999

And you want to search on last 4 digits or
something similar. In that
case
you need to decrypt and compare
row-by-row (unless you split the SSN up
into
separate columns...).