

RE: Cannot decrypt files encrypted using Crypto API on a different

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.platformsdk.security/2009-01/msg00049.html>

- *From:* lelteto <lelteto@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 21 Jan 2009 09:54:04 -0800
-

You don't have public key, so PUBLICKEYBLOB is not doing anything good for you. (Unless you go and implement the proper protocol I outlined in the previous message which uses the recipient's public key.) SIMPLEBLOB needs a KEK (key encryption key) to protect the session key. You have two options for that

(1) use key derivation – in your case probably from some fixed (secret?) text in your client app and server code

(2) directly inject the key encryption key into your container (again, this would be something fixed, embedded into your client app and server code).

Direct injection can use eg. the PLAINTEXTKEY, you would need to manually format the blob before calling CryptImportKey.

Note that it is recommended that the KEK is at least as strong as the session key you are protecting. (If you use AES128 as session key, you can also use AES128 KEK.) Of course, in your case IF you embed your key (or data used to derive the key) in the client application, it is not very secure – but you indicated you don't want high security.

Laszlo Elteto
MVP Developer Security
SafeNet, Inc.

"vishalchowdhary" wrote:

Hi Lelteto,

You raise a good question. The basic intent of this is that we just want to make it difficult for a dishonest client to read our binary files without doing the decryption. We are not looking at a strong encryption but just a basic encryption which suffices our needs.

I have been able to do this for the moment using PLAINTEXTKEY which works across machines. But when I use Exchange key with SIMPLEBLOB or PUBLICKEYBLOB, this doesn't work.

Thanks,
Vishal

RE: Cannot decrypt files encrypted using Crypto API on a different

"lelteto" wrote:

So you

- 1) want to distribute the same file for MANY users you don't even know before the server sends the encrypted data? or
- 2) you want to send the file to the user without any authentication / prior contact?

Yes, it is possible to include the decryption key WITH the encrypted data – but what is the point to encrypt the data if ANYBODY can decrypt it (since the key is there)???

Try to explain HOW the server knows the clients, how they communicate (ie. the server just sends something to somebody or first the client contacts the server?) and I may suggest something.

Laszlo Elteto
MVP, Developer security
SafeNet, Inc.

"vishalchowdhary" wrote:

Hi Lelteto,
This still won't be helpful in my case. In my case, the server will not be knowing any of the client's public keys. The server is just supposed to somehow encrypt the file and distribute it to the clients. The client must then get the key from the encrypted file and use that key to decrypt the file.

We just need this level of encryption.

Thanks,
Vishal

"lelteto" wrote:

Sure. I suggest the exact same thing – with two small changes
(1) IF you want to allow multiple users to be able to access the delivered content on the computer (possibly storing it somewhere and able to decrypt / use multiple times) you would need to use a machine key container. Just add

RE: Cannot decrypt files encrypted using Crypto API on a different

the flag CRYPT_MACHINE_KEYSET to CryptAcquireContext on the client's computer.

Of course, if the intent is NOT to share content among users then don't add this.

(2) On the client computer you app first would try to open the container (CryptAcquireContext) and only if it is not found (error == NTE_BAD_KEYSET) then you create it (CryptAcquireContext with CRYPT_NEWKEYSET the CryptGenKey). From that point the sequence is the same as below:

- the client sends the public key to the server
- the server generates session key, wraps it with the client's public key, encrypts the content with the session key and sends both the wrapped session key and encrypted content to the client
- the client unwraps the session key and decrypts the content

Note that the last step could be repeated as many times as the user wants assuming you stored the received data (wrapped session key and encrypted content) on the local computer.

Is this helpful?

Laszlo Elteto
SafeNet, Inc.

"vishalchowdhary" wrote:

Hi,
Let me take a moment in here and explain u the business requirement.

We have a desktop application which is used by many clients. We generate some binary files within the application and then we need to encrypt them and send it over to the clients (we don't know anything about their public key). The reason for sending files

RE: Cannot decrypt files encrypted using Crypto API on a different

this way is that we don't want dishonest clients to open the binary files since it is worth tens of thousands of dollars. So they can only view the binary files from the application which must internally decrypt the files. This whole process should be transparent to the client.

Can you suggest something now?

Thanks,
Vishal

"lelteto" wrote:

Depends on HOW you want to get the decryption key on the second computer. Did you export the encryption key (on the first computer) using the second computer's public key? Here is how normally you do this:

1. On the second computer create a permanent

RE: Cannot decrypt files encrypted using Crypto API on a different

container
(CryptAcquireContext
with
CRYPT_NEWKEYSET
and with a
unique
container
name)
2. create a
private/public
key pair
(CryptGenKey
with
AT_KEYEXCHANGE).
3. export
the public
key into a
blob
(CryptExportKey;
hKey is
what you
get
from step 2,
hExpKey is
NULL, blob
type is
PUBLICKEYBLOB)
and send
this blob
to the other
computer
4. on the
first
computer
start a temp
session
(CryptAcquireContext
with
CRYPT_VERIFYCONTEXT)
5. generate
a session
key
(CryptGenKey
with algo
preferably
AES)
6. encrypt
your data
with this
key
(CryptEncrypt)

RE: Cannot decrypt files encrypted using Crypto API on a different

RE: Cannot decrypt files encrypted using Crypto API on a different

7. import
the other
computer's
public key
(CryptImportKey,
hPubKey
NULL,
blob is what
you got
from step 3)
8. export
the session
key
(protected
by the other
computer's
public key:
CryptExportKey
with hkey =
key from
step 5,
hExpKey =
key from
step 7, blob
type =
SIMPLEBLOB)
9. send the
encrypted
data AND
the blob
from step 8
to the other
computer
10. On the
second
computer
open the
container of
your private
/ public key
pair
(CryptAcquireContext)
11. get your
key pair
(CryptGetUserKey
with
AT_KEYEXCHANGE)
12. import
the session
key
(CryptImportKey

RE: Cannot decrypt files encrypted using Crypto API on a different

RE: Cannot decrypt files encrypted using Crypto API on a different

with blob
from step 8,
and
hPubKey is
the key
handle from
step 11)
13. Now
you can
decrypt the
data with
the key you
got in step
12

Hope this
helps,

Laszlo
Elteto
SafeNet,
Inc.

"vishalchowdhary"
wrote:

Hi,
I'm
new
to
the
Crypto
API
and
used
it
to
encrypt
a
bunch
of
files.
The
decryption
works
fine
on
my
machine.
However,

RE: Cannot decrypt files encrypted using Crypto API on a different

RE: Cannot decrypt files encrypted using Crypto API on a different

when
I
try
to
decrypt
the
encrypted
files
on
a
different
machine,
I
get
the
error
code
8009000d
for
CryptImportKey()

Can
anyone
please
help
me?

Thanks,
Vishal