

Re: Socket Server with Encryption help

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.security/2006-03/msg00100.html>

- *From:* "Andre Azevedo" <xpto@xxxxxxx>
 - *Date:* Thu, 9 Mar 2006 10:07:00 -0300
-

Hi Valery,

Thanks for the post.

But, remember, I don't want to create new methods of encrypt, etc. I just want to put some encrypt mechanism in my TCP server/client implementation using EXISTING classes from .Net Framework 2.0.

I wrote those steps based in a RSACryptoServiceProvider for asymmetric and maybe RijndaelManaged for symmetric keys as my application needs to authenticate a lot of clients in different networks. Before the client sends data, it authenticates and encrypts the message.

If SslStream does the job, ok, I'll use it.

Thanks again,

—

Andre

"Valery Pryamikov" <valery@xxxxxxx> wrote in message
<news:%23g0PWFvQGHA.4976@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Hi, (I'm valery :-)

After reading your post I got a very strong suspicion that regardless of your saying that you read "a lot of papers about Symmetric, Asymmetric, Hash, Envelope and Signature" you didn't read even distantly enough to be able to implement something even distantly secure.

Authentication protocols are fiercely difficult to get right. The classical paper on three party authenticated protocols design was written by Needham and Schroeder "Using encryption for authentication in large networks of computers" in 1978, where they described several protocols, one of which was modified, strengthened and extended a bit later to become what is now known as Kerberos. At the end of their paper, Needham and Schroeder wrote that "protocols such as those developed here are prone to

Re: Socket Server with Encryption help

extremely subtle errors that are unlikely to be detected in normal operations. The need for techniques to verify the correctness of such protocols is great." Since then it was many attempts to develop a formal protocol verification tools. When it concerns authentication protocols, the most successful is BAN logic that stays for the names of their creators Burrows, Abadi, Needham and was first described in their "A logic of authentication" 1990 paper that could be found here <http://www.stanford.edu/class/cs259/papers/ban1990.pdf>. BAN logic was successfully used for analyzing and detecting errors/redundancies in many authentication protocols such as Kerberos (redundancy), Needham–Shroeder (design problem), Denning–Sacco (design problem) and many others (don't recall names from the top of my head, but there was quite a lot).

Another well known formal method of analyzing protocol correctness was NRL protocol analyzer (Navy Research Laboratory). Here you can find more details on it:

<http://chacs.nrl.navy.mil/publications/CHACS/1994/1994meadows-pap.pdf>

With some extension to address problems of Denial of Service that could be found in another paper by Meadows "Formal Framework and Evaluation Method for Network Denial of Service" <http://citeseer.ist.psu.edu/384.html>

Very good survey of formal methods of protocol analysis you can find here: <http://www.csc.liv.ac.uk/~wiebe/pubs/Documents/survey.ps>

And you can check this my blog post for very quick point on single aspect of authentication protocol:

<http://www.harper.no/valery/PermaLink.guid.aa88fdd0-ac5c-4315-969a-6954b4e05ad7.aspx>

But frankly, I think that you really need a lot of cryptographic background before even starting reading papers on development of authentication protocols and think of implementing such protocols yourself!

"Practical Cryptography" book, that was suggested to you earlier, could be a good starting point if you only need overall understanding of basic cryptographic problems and don't plan to make cryptography to be your specialty.

However if you want cryptography to be your speciality you probably have

Re: Socket Server with Encryption help

to start with Bellare–Rogaway's "Introduction to Modern Cryptography"
<http://www-cse.ucsd.edu/~mihir/cse207/classnotes.html> (online)

After that, I'd highly recommend you to read Douglas Stinson's
"Cryptography Theory and Practice" third edition (second edition doesn't
contain chapters on protocols) <http://www.amazon.com/gp/product/1584885084>
(book)

Good alternative (or addition) to Stinson's book is Wenbo Mao's "Modern
Cryptography: Theory and Practice" is also a recommended reading at that
stage.

<http://www.amazon.com/gp/product/0130669431> (book)

After that you MUST! read Goldwasser–Bellare's "Lecture Notes on
Cryptography" <http://www.cs.ucsd.edu/users/mihir/papers/gb.html> (online)

If you want implement algorithms yourself – you should read encyclopedic
"Handbook of Applied Cryptography" <http://www.cacr.math.uwaterloo.ca/hac/>
(available online and as printed book)

After you done with it, for hardcore cryptography and protocol design work
you'll need to read Goldreich's "Foundations of Cryptography" (draft is
available online <http://theory.lcs.mit.edu/~oded/frag.html>, but I'll
highly recommend to buy books <http://www.amazon.com/gp/product/0521791723>

<http://www.amazon.com/gp/product/0521830842>)

Only after you done reading references above (and reading papers that are
most often cited in references section of mentioned books, lectures notes
and papers), then you may try to implement your own authentication
protocols, and I'm sure – you will not be asking the questions that you
are asking today.

From the other side, I would not recommend reading Schneier's "Applied
Cryptography". His non-rigorous discussion of protocols and algorithms
makes it very easy to miss (oversee) difficulties of designing them, plus
there are a number of conceptual mistakes in some of his descriptions
(that was mused many times in discussions that you can find in Usenet
groups and on Internet). Schneier's book was an attempt to create simple
exposition to cryptography for developers; however by Schneier's own words

Re: Socket Server with Encryption help

it was a mistake.

–Valery.

<http://www.harper.no/valery>

"Andre Azevedo" <xpto@xxxxxxx> wrote in message
<news:%23U1j2DtQGHA.3192@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Hi all,

I've started to develop a server and client socket classes with encryption. The main communication/transport classes is working fine and now I will write some encryption process.

After reading a lot of papers about Symmetric, Asymmetric, Hash, Envelope and Signature I still have some doubts and I will explain what I calling the "Authenticate Flow" in client/server socket communication:

1. Client connects into Server and Server accepts the connection.
2. Server send his encryption public–key to Client.
3. Client creates a new symmetric session–key, encrypt it using the Server encryption public–key and send it to Server plus the Client sign public–key.
4. Server decrypt Client symmetric session–key and simply replies to Client, telling "Ok, I have the symmetric session–key and your sign public–key".

Now, every time Client need send some data, it does the following:

5. Client encrypts data with symmetric session–key, sign (hash) the result with sign private–key. Client then sends the hash result and the encrypted data to server.
6. Server sign (hash) the encrypted data with the same Client hash algorithm and save it in Hash1. After, it decrypt the sign (hash) send by client using Client sign public–key to obtain the Hash2. If Hash1 and Hash2 are the same, then is the correct Client. Otherwise, closes the connection.
7. If ok, Server then decrypt data with symmetric session–key.

Well, que questions now:

A – The hash algorithm is know by the Client and Server since it's my implementation of both and I don't need to send the hash algorithm information. Is this acceptable?

B – Sendind the Client sign public–key to Server is ok. But, after that, I'm sending some data using Client sign private–key to Server. Is this secure? Is a normal way to do it?

Re: Socket Server with Encryption help

C – Do I need to do the 5, 6 and 7 steps every time Client needs send some data to Server and vice-versa? Or these steps it's executed only once only certify the Client and, after that, both sides can send messages encrypted only with symmetric session-key?

Sorry for the long post. Any help will be appreciated

TIA,

--

Andre Azevedo