

Re: Encryption over the web without SSL?

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2003-02/2551.html>

From: Carlos Moreno (moreno_at_mochima_dot_com@xx.xxx)

Date: 02/13/03

From: Carlos Moreno <moreno_at_mochima_dot_com@xx.xxx>

Date: Thu, 13 Feb 2003 00:29:32 -0500

Thanks for your comments.

Yep, certainly impersonation and/or MITM attacks would be an enemy to consider :-(

I wonder if there might be a way for the server to "agree in advance" with the clients (i.e., some way of transmitting the PK by some other way? E-mail maybe?)

BTW, a few comments:

- > 1) *impersonation of the server with session key compromise*
- >
- > *An attacker A impersonates the Server by choosing his own public key $K_{pub,a}$*
- > *and sends it to C. C encrypts the session key gladly and A can decrypt it.*

Well, say that my domain name is www.crypto-depot.com ... Well, someone might register the domain www.crypto-depot.net, and put a carbon-copy of my site! (hell, I might even type .net one day by mistake, and not even get to notice that I am on the wrong web site!!)

And they may or may not have SSL — they simply impersonated my server *completely*...

Sure, sooner or later this is bound to be discovered... But I wonder how different both situations are. (I might be missing something obvious, I have the feeling :-))

- > 2) *Man in the middle attack*
- >
- > *The attacker A gets the real public key from S (acts as client) and places*
- > *himself in the middle routing messages between A and S, listening in on*
- > *everything.*
- >
- > *A ← S: $K_{pub,s}$*
- >

sci.crypt: Re: Encryption over the web without SSL?

> C <-- A: $K_{pub,a}$
>
> C --> A: $E_{K_{pub,a}}(w)$
> A --> S: $E_{K_{pub,s}}(w)$

So, if I'm understanding correctly, this allows the attacker to decrypt what's coming from the server, and not what's coming from the client (since the attacker does not have the private key of the server, and the client transmitted the session key encrypted with S's public key).

So, the attacker would need to share the same encryption key as the client, which will never be the case, since the first session key is generated by the client, and everything after (including possibly a new session key, generated by the server with better "randomness") will be encrypted with the key that the client was transmitting...

Oh wait... No, of course!! If the attacker knows the protocol, then he will be in the middle, and will pretend to be the client when talking to the server, and pretend to be the server when talking to the client, right? Ok, I think I get it... (I wonder if I should remove the above two paragraphs... Well, I'll let them, in case there's any fun in further discussing it)

So yep, back to good'old SSL certificates... :-)

> Also, your key exchange scheme for changing to a new "more secure" session
> key doesn't provide forward security. If a session key somehow leaks, future
> session keys are compromised.

Well, I was thinking more in the lines that if you use a weak key once, you have once the chances of succesful attacks. If you keep using the same weak key over 10 transmissions, you have ten times more chances of succesful attacks.

But I see where you're going: if the key generated by the client is truly weak, then the protocol is bound to be broken... (I wasn't thinking that the key generated by the client be truly weak... I was more along the lines: the server key is *better*, so we might as well switch to use that one as soon as we can -- which is the moment of the second transmission from the server, or the first transmission after having received the key from the client)

Anyway, no, I won't do it (not at home, not at work! :-)), so you can sleep in peace -- you did help a misguided (however well-intentioned) soul avoid a BIG mistake! :-)

Thanks for your comments and advice!

Carlos

--

Re: Encryption over the web without SSL?