

Re: Designing a secure message format

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2006-06/msg01443.html>

- *From:* daw@xxxxxxxxxxxxxxxxxxxxxxxxxxxx (David Wagner)
 - *Date:* Fri, 30 Jun 2006 04:21:56 +0000 (UTC)
-

imposterrific@xxxxxxxx wrote:

I am designing an application which will allow users to send encrypted messages (anything from a few characters to a gigabyte or so) over insecure channels. I want the system to be extremely simple to use and so I envision a user entering a passphrase into the application to decode a message.

Have you considered re-using a standard scheme, like "gpg -c" or SRP?

If the sender and receiver be online at the same time, a password-authenticated key exchange scheme (like SRP; there are many others as well) may be best. Most users don't choose very good passwords, so an attacker who can try many passwords offline may have some partial success. Password-authenticated key exchange schemes prevent offline password-guessing attacks; they require the attacker to mount an active attack. Consequently, schemes like SRP are more resilient even if some users don't choose great passwords.

If the sender and receiver won't both be online at the same time, or if you can't use a password-authenticated key exchange scheme, then something like "gpg -c" is about as good as it gets. Encourage users to choose hard-to-guess passphrases.

If there's any way you can arrange for the users to have public keys (e.g., using a PGP or SSH like mechanism), then that will probably provide better security than conventional password mechanisms like "gpg -c".

But no matter what you do, I highly encourage you to reuse existing standard crypto tools if at all possible. If you try to design your own crypto mode of operation, you are risk getting something wrong. (Example: It seems you have made the textbook error of forgetting about the need for message authentication, in the scheme outlined in your post.)

.