Hettinga's Best of the Month

Submitted by our Contributing Editor Bob Hettinga as his choice of best article of month found on Internet.

Robert Hettinga rah@shipwright.com
44 Farquhar Street, Boston, MA 02131 USA (617) 958-3971

A case for 2560 bit keys

by **David F. Ogren**
Mail: ogren@concentric.net
PGP Key ID: 0x6458EB29

This article first appeared on the cypherpunks list.

Here are a few thoughts on RSA key sizes. There is nothing new or revolutionary herein, but I think it does provide a good case for using large RSA keysizes.

Traditionally, we examine the threat model and determine the approximate ability of the attacker to factor secret keys. Then a keylength is selected that exceeds the attackers ability to factor in a reasonable amount of time.

For example, if we assume that the NSA can factor any number with the speed of the special number sieve, and has 10^9 mips of computing power (doubling every 1.5 years) we can make the following estimations:_1_

Using these assumptions, the NSA could crack a 1024 bit key in ~11 days, a 1536 bit key in 10 years and a 2048 bit key in 26 years. _2_ Note that this would require the full resources of the NSA, however. Thus, even the mighty resources of the NSA could only crack 42 1024 bit keys in 1996 (including Moore's law). _3_, _4_

Similarly, a large corporation with 10^7 mips in computing power (and the same super-efficient factoring algorithm) could crack a 1024 bit key in 2 years, a 1536 bit key in 20 years, and a 2048 bit key in 36 years.

My interpretation of these results: 1024 bit is probably safe for most reasonable threat models. Only individuals with extremely high threat models should be concerned about 1024 bit keys in 1996. Even those with extremely high threat models should be satisfied with 1536 bit keys.

Despite the above, there are convincing arguments for longer RSA keys. Instead of asking "Why should we have longer keys?", perhaps we should be asking "Why _shouldn't_ we have longer keys?"

In a hybrid cryptosystem such as PGP, very little of the computational process is consumed by RSA encryption. Only a tiny fraction of the message is RSA encrypted (the session key), and thus the time-critical operation is the symmetric crypto system (IDEA for PGP).

As an experiment generate a 2047 bit PGP key and a 512 bit PGP key. Encrypt a file (preferably of a reasonable size) using both keys. Depending on the computer you are using, the time difference between the two keys will be a matter of few seconds or even a fraction of a second.

And so we have to ask ourselves, why _not_ use a 2047+ bit key. It has greater longevity and greater security. Why not be overcautious when the cost is so small?

It seems foolish that we use RSA keys that are less secure than our IDEA session keys. Our RSA keys are much more valuable than our session keys. I will use my RSA key to encode hundreds of messages. Each session key I will use only once. An attacker who learns one of my IDEA session keys can decrypt only that message. An attacker who learns my RSA key can decrypt any of my messages, past or present. (He can also impersonate my signature, but that's another discussion entirely.)

If I send one message weekly that my attacker is interested in, and change my RSA key every two years, my RSA key is at least 104 times more valuable than any individual key. Does it not make sense that the RSA key should ideally be 104 times more difficult to crack?

If increasing the RSA keylength was overly cumbersome to the process then designing the RSA keylength to meet minimum acceptable standards could be understood. But since increased RSA keylengths are cheap in terms of computing power, would it not be better to pick RSA keylengths that are more secure than the session keys?

And thus, 2560 bit keys are not unreasonable. They are not significantly slower to use (most of PGP's time is spent IDEA encrypting), and yet are effectively invulnerable. By "invulnerable" I mean that any attacker capable of cracking your RSA key would have an easier time hacking your individual IDEA session keys, and would never have any need to hack the RSA key itself. And if you have threat models this severe you are a) hopelessly paranoid, b) SOL.

Footnotes:

_1_ These approximations of factoring difficulties and the computing resources are taken directly from Applied Cryptography by Bruce Schneier, page 161.

_2_ Taking into account Moore's law, the amount of processing power spent during a period of time is the integral of Power * 2^(t/1.5)dt (from 0 to x) = Power * 1.5 / (ln 2) 2 ^(t/1.5) (also evaluated from 0 to x). Which is approximately equal to Power * 2.164 * (2^(x/1.5) - 1). Thus in three years a corporation starting with 10^7 mips could produce 10^7 * 2.164 * (2^(3/1.5)-1) = 6.492 * 10^7 mips-years.

_3_ Any attempt to determine the computing power and cryptanalysis power of the NSA should be taken with a grain of salt. There are several very critical and arbitrary assumptions made in order to obtain these numbers.

_4_ Additionally, any attempt to discern the future of cryptanalysis should also be taken with a grain of salt. Who can tell what computers will like be in ten years?

*****
Don't know what PGP is? Send a message to me with the subject GETPGPINFO

Need my public key? It's available by server or by sending me a message with the subject GETPGPKEY