

# The Curse of Cryptography Numerology

**T**he problem of cryptographic numerology has plagued modern cryptography throughout most of its life. The basic concept is that as long as your encryption keys are at least “this big,” you’re fine, even if none of the surrounding infrastructure

Tbytes of RAM for the final step, and a 1,280-bit key would require roughly a petabyte, all in a single machine or single-machine equivalent. (A standard distributed cluster won’t work because of interconnect-latency problems.)

For nongovernment users, keys longer than 768 bits still look safe for a considerable while. An analysis of 30 years of data on factoring efforts found that the results were very linear.<sup>3</sup> Using this data, it is estimated that a single 1,024-bit key could be factored by around 2040. That’s a massive effort for one key, with every one of the millions of other 1,024-bit keys in use today still being safe until the same effort gets applied to them, as well.

## Real Threats

Now let’s look at the actual threats that people and organizations using these keys are facing. As one key-length analysis puts it,

Is it reasonable to assume that if utilizing the entire Internet in a key breaking effort makes a key vulnerable that such an attack might actually be conducted? If a public effort involving a substantial fraction of the Internet breaks a single key, does this mean that similar sized keys are unsafe?<sup>3</sup>

The general answer to these questions is no. However, to see why this is so, we must apply the thinking of standard commercial risk management rather than cryptographic numerology.

We actually have a pretty good

IAN GRIGG  
*Financial  
Cryptographer*

PETER  
GUTMANN  
*University of  
Auckland*

benefits from that size or even works at all. The application of cryptographic numerology conveniently directs attention from the difficult to the trivial, because choosing a key size is fantastically easy, whereas making the crypto work effectively is really hard.

In this sense, cryptographic numerology is a prime example of what psychologists call zero-risk bias. That is, people would rather reduce some (often largely irrelevant) token risk to zero rather than address other, more serious risks that are much harder to deal with. Zero-risk bias is particularly common in government agencies facing threats from other governments and from their own country’s paparazzi press. Charged with protecting their government from these threats, the agencies quickly sink into a quagmire of zero-risk bias.

The surface threat is the cryptographers and supercomputers of another government’s intelligence agencies, not the threats we’re more used to in the commercial and Internet world. One such mundane example is the keystroke-logger Trojan that a drive-by download installed on a victim’s PC six months ago and

that has been quietly gathering data ever since. It might be debatable whether the real or imaginary threat of massive key crunching is a serious one to various government agencies. What’s far clearer is that the numerology threat model bears no relation to anything the rest of the world faces.

## Of Key Size and Resources

Consider key sizes for public-key algorithms. A 512-bit RSA key was first successfully factored in 1999 using 35 years of computing time on 300 workstations and a final matrix step requiring nine days on a Cray supercomputer.<sup>1</sup> Ten years later, a half-year effort pushed this out to 768 bits, with the final step occurring on a multinode shared-memory cluster and consuming up to a terabyte of memory.<sup>2</sup> The 768-bit key was several thousand times harder to factor than the 512-bit one, and a 1,024-bit key will be around a thousand times harder to factor than the 768-bit one.<sup>2</sup> The next key size, 1,280 bits, will be half a million times harder than the 768-bit one. To rephrase this in terms of resource requirements, a 1,024-bit key would require around 40

metric for threats facing real-world systems: the losses due to various types of attacks on security systems. On the basis of the past 15 to 20 years of modern cryptography and attacks against the same, we can reasonably predict what will and won't be a problem. With a good degree of reliability, we can say that during that time, no one ever lost money to an attack on a properly designed cryptosystem (meaning one that didn't use homebrew crypto or toy keys) in the Internet or commercial worlds.

On the other hand, we are losing, and continue to lose, billions of dollars (depending on which source you go to) owing to the failure of everything but the cryptography. As cryptographer Adi Shamir pointed out, "Cryptography is usually bypassed. I am not aware of any major world-class security system employing cryptography in which the hackers penetrated the system by actually going through the cryptanalysis. ... Usually there are much simpler ways of penetrating the security system."<sup>4</sup>

So, in practice, cryptography gets bypassed rather than attacked. There's no need to even think about attacking the cryptography when it's so much easier to target the user, the user interface, the application, the protocol implementation, the business and social processes in which it's all used, or absolutely anything but the crypto. Probably the best-known example of this is phishing, which completely negates any effects of SSL/TLS (Secure Sockets Layer/Transport Layer Security) in attempting to protect sensitive communications with Web servers. The encryption doesn't even have to be very strong to be useful, it just must be stronger than the other weak links in the system. Using any standard commercial risk management model, cryptosystem failure is orders of magnitude below any other risk.

Consider the recent hack of Comodo, a certification authority (CA) dependent on its RSA keys. The attacker pointed out that "RSA 2048 was not able to resist in front of me."<sup>5</sup> Even with the 2,048-bit keys required by cryptographic numerologists, a lone Iranian crypto-jihadist simply bypassed the crypto, hacked the website, and stole the account details to issue certificates as if he were the CA's reseller. This story is as old as Ali Baba and the 40 Thieves, and the secret is as well known as Open Sesame!

### ***Not-So-Real Threats***

There's one remaining bogeyman that's often raised in cryptography's defense: the new-attack scenario. But some unknown new attack would be generally ruled out by standard risk management, which says that if it's unknown to us, it isn't likely and is thus dismissed from the risk model. Any attempt to mitigate an unknown, and probably nonexistent, risk becomes subject to Geer's law, after security philosopher Dan Geer:

Any security technology whose effectiveness you can't empirically determine is indistinguishable from blind luck. (Geer's law is a paraphrase of the analysis first presented in "Information Security: Why the Future Belongs to the Quants."<sup>6</sup>) Looking at the scary-new-attack threat another way, if your risk model is going to incorporate imaginary threats, the threat that "someone breaks algorithm X with key size Y" could just as well be "someone breaks algorithm X no matter what the key size is."

### ***How Cryptographic Numerology Hurts Security***

So how does the blind application of cryptographic numerology negatively affect security? For public-key algorithms, an increase in key size isn't free. The mandatory switch from 1,024-bit to 2,048-bit keys decreed by government agencies such as the US National Institute of Standards and Technology and similar agencies in other countries results in an

order-of-magnitude increase in processing for each key. This processing is expensive and will cause behavior shifts. After 15 years of Internet commerce, we're now well past the point at which 1,024-bit keys can be employed without much of a slowdown in our user experience, but this isn't true for 2,048-bit keys. The massive slowdown arising from the application of cryptographic numerology will encourage developers and IT managers to continue to run protocols in an unsecured manner, instead of opportunistically deploying encryption everywhere because there's little reason not to.

Consider an embedded print server on a corporate network that, for whatever reason, the vendor has configured with a dinky 512-bit key to protect the remote printing interface. An attacker can break this in a couple of months, or several weeks if he or she has access to a distributed-computing grid. After all that effort, the attacker can now hijack communications with the print server and ... delete entries from the print queue and turn off toner saving on the printer. What's more, to do this, the attacker must have already penetrated the corporate network with the ability to actively manipulate traffic on it, a far more serious threat than accessing a print server. No rational attacker would even consider targeting this key, because it has no value apart from generally deterring casual misuse.

The problem is that cryptographic numerology operates in a vacuum, ignoring all other operational considerations that affect the overall system's security. Consider the goal of "SSL everywhere,"<sup>7</sup> of running as much traffic as possible over SSL/TLS simply because it's slowly becoming cheap enough that in many cases it can be turned on by default. Although there might be no showstopper vulnerability to justify deploying

SSL-everywhere, its presence does mitigate a slew of long-lived lesser security problems.

One such problem is the ability to hijack session authenticators such as cookies sent out over unprotected channels. This ability was aptly demonstrated by the Firesheep add-on for Firefox in late 2010, after years of unsuccessful attempts to fix the problem. Even SSL-everywhere with a trivial 512-bit key would have stopped this attack dead in its tracks. This is because it was a purely opportunistic attack that exploited the fact that the authentication data was unprotected and could be obtained through a passive sniffing attack.

Unfortunately, cryptographic numerology doesn't admit such operations-level thinking. An opportunistic attacker passively sniffing authentication cookies is as unlikely to break a 512-bit key as a 2,048-bit key. But the latter, requiring 80 times the work of the 512-bit one, is the only one that cryptographic numerology allows. Even the step up from 1,024-bit keys (which are mostly cheap enough to allow SSL-everywhere in many sensitive Internet situations) to 2,048-bit keys would require a tenfold increase in server processing power or the number of servers to handle the same number of clients.

So, rather than making us more secure, the focus on cryptographic numerology falls afoul of the law of unintended consequences. Concentrating on fighting the threat of numerically endowed foreign powers makes us significantly less secure by excluding the use of SSL-everywhere. Key-cracking attempts by foreign intelligence interests might threaten a few government agencies, but the response of mandating key size increases affects all defenders alike. They're more likely to not deploy any cryptography at all than to try

to convince their users to wait for slow cryptography. □

### Acknowledgments


This article is based partly on a draft of an as-yet unfinished book on engineering security.

### References

1. "RSA-155 Is Factored!" RSA Laboratories, 22 Aug. 1999; [www.rsa.com/rsalabs/node.asp?id=2098](http://www.rsa.com/rsalabs/node.asp?id=2098).
2. T. Kleinjung et al., "Factorization of a 768-Bit RSA Modulus," report 2010/006, Cryptology ePrint Archive, 6 Jan. 2010; <http://eprint.iacr.org/2010/006>.
3. R. Silverman, "A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths," *RSA CryptoBytes Bulletins*, no. 13, Apr. 2000; [www.rsa.com/rsalabs/node.asp?id=2088](http://www.rsa.com/rsalabs/node.asp?id=2088).
4. A. Shamir, "Cryptology: A Status Report," Turing award lecture (video), 2002; <http://awards.acm.org/citation.cfm?id=0028491&aw=140&ao=AMTURING&yr=2002>.
5. P. Bright, "Independent Iranian Hacker Claims Responsibility for Comodo Hack," blog, 28 Mar. 2011; [www.wired.com/threatlevel/2011/03/comodo\\_hack](http://www.wired.com/threatlevel/2011/03/comodo_hack).
6. D. Geer, K.S. Hoo, and A. Jaquith, "Information Security: Why the Future Belongs to the Quants," *IEEE Security & Privacy*, vol. 1, no. 4, 2003, pp. 24–32.
7. "HTTPS Everywhere," Electronic Frontier Foundation; [www.eff.org/https-everywhere](http://www.eff.org/https-everywhere).

*Ian Grigg writes the Financial Cryptography blog and is part of CAcert, the community Certification Authority. Contact him at [iang@iang.org](mailto:iang@iang.org).*

*Peter Gutmann is a researcher at the University of Auckland. Contact him at [pgut001@cs.auckland.ac.uk](mailto:pgut001@cs.auckland.ac.uk).*

 Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.