Universal security

from bits and mips to pools, lakes – and beyond

Arjen K. Lenstra[1]
with Thorsten Kleinjung[1] and Emmanuel Thomé[2]

[1] EPFL IC LACAL, Station 14, CH-1015 Lausanne, Switzerland
[2] INRIA CNRS LORIA, Équipe CARAMEL - bâtiment A,
615 rue du jardin botanique, F-54602 Villers-lès-Nancy Cedex, France

Abstract. The relation between cryptographic key lengths and security depends on the cryptosystem used. This leads to confusion and to insecure parameter choices. In this note a universal security measure is proposed that puts all cryptographic primitives on the same footing, thereby making it easier to get comparable security across the board.

Current security levels

The security of a cryptographic primitive is measured by the effort to break it. If that effort is 2^k, measured in some agreed upon unit, then it offers *k-bit security* and it is said to have *security level k*. Although this looks easy, it turns out to be impractical. For symmetric cryptosystems it is indeed easy: with a k-bit key they are supposed to offer k-bit security, measured in the most basic application of the cryptosystem, else they are no good. Cryptographic hash functions are harder: if the hash length is h they should offer and cannot offer more than $h/2$-bit security, measured in basic applications of the hash function. Given comparable speeds of the systems in similar environments – both in software or both in hardware – their security levels can easily be compared in a meaningful manner.

With public key primitives it gets confusing. For RSA the runtimes to factor an RSA modulus can be measured using experiments on small numbers. The results are then extrapolated to derive estimates for larger moduli: if factoring an n-bit modulus takes time T, then the time to factor an m-bit modulus (for m not much bigger than n) is estimated as $\frac{N(m)}{N(n)}T$, where $N(k) = \exp(1.923(\ln 2^k)^{1/3}(\ln\ln 2^k)^{2/3})$. In 1988 the first factorization of a 100-digit modulus required "100 MIPS years": equivalent to a century of computing time on a computer that performs a million "instructions" per second. In 1999, using a better factoring method, a 512-bit challenge modulus took about 40 years of computing time on a then current core (running at several hundred MHz, in four months in parallel on many cores), which was, somewhat questionably, estimated as 8400 MIPS years. The 2009 factorization of a 768-bit modulus took roughly a year on 2000 cores running at 2GHz. With $\frac{N(1024)}{N(768)} \approx 1200$, a 1024-bit modulus can be factored within two million 2GHz core-years. Further extrapolation is trickier, but it is reasonable to estimate that 2048-bit RSA moduli are a billion

times harder to factor than 1024-bit ones. Using generic figures for the speed of software implementations, it follows that 768-, 1024-, and 2048-bit RSA have approximate security levels 66, 76, and 106, respectively.

For discrete logarithm public key cryptosystems additional parameters further confuse the picture. Let G be a well-chosen prime order q group for which the group operation is 2^c times slower than a basic application of a symmetric cryptosystem (or cryptographic hash function). It may be assumed that $0 \leq c \leq 10$. Discrete logarithms in G offer $c + \log_2 \sqrt{q}$-bit security, assuming a large enough finite field F if $G \subset F^*$, the size of which is estimated in the same way as the security of RSA, with a small and uncertain twist $\tau > 0$ and with the potential of all kinds of trouble if an extension field is used. Thus, if q is a 160-bit prime and G an elliptic curve group, the security level is c+80. If, for a similar q-value, $G \subset F^*$ for a cardinality p finite field F, then the security offered by F^* needs to be taken into account as well: the security level is $\min(c + 80, 76 + \tau) = 76 + \min(c + 4, \tau)$ if p is a 1024-bit prime, and $\min(c + 80, 106 + \tau) = c + 80$ for 2048-bit p.

Lattice-based public key cryptosystems still escape proper analysis. Despite enthusiastically riding post-quantum waves and fully homomorphic hot air balloons, they have not found wide-spread application and are not further discussed.

Summarizing the above, for symmetric cryptosystems the key length and the security level are the same, for cryptographic hash functions the security level is half the hash length, for RSA the security level is an obscure but small and quickly decreasing fraction of the modulus length, and for discrete logarithms it is half the size of the largest prime dividing the group order, unless one works in a multiplicative group of a finite field in which case extra care needs to be taken. This is easy enough for those in the know, but incomprehensible to most: it is not uncommon for people to proudly declare that they are using 128-bit RSA moduli for their 128-bit security cryptosystems – or indeed for crypto-practitioners to use 256-bit AES, SHA-512, and 512-bit RSA thinking they get an overall security level of $\min(256, \frac{512}{2}, \frac{512}{2}) = 256$.

Intuitive security levels

The problem is that for all these cryptosystems key length and security level are measured in bits, but that the relationship between the two varies wildly – from trivial, to simple, to rather contrived. If this is not well understood, it is tempting to use just the trivial and simple relationships and to forget about the complicated ones, with potentially disastrous consequences. This has occurred in practice, on multiple occasions and involving major corporations.

To address this problem the relationships between key length and security level must be put on the same footing for all cryptosystems. Because the complicated relationships cannot be simplified, all must be made equally complicated. Below a new definition of security level is proposed that does away with "k-bit security" for "security level k". It has the additional advantage that it allows a more intuitive interpretation of what security actually means.

The new approach was inspired by a remark made by the third author during his presentation of the factorization of the 768-bit RSA challenge at Crypto 2010: *We estimate that the energy required for the factorization would have sufficed to bring two* $20^\circ C$ *Olympic size swimming pools to a boil.* This amount of energy was estimated as half a million kWh. Thus, a cryptosystem is said to offer *pool security* if breaking it requires as much energy as it takes to boil a single Olympic size swimming pool (i.e., 2500 cubic meters of water). It follows that 65-bit symmetric cryptosystems, 130-bit cryptographic hash functions, and 745-bit RSA currently all offer pool security (because 768-bit RSA offers 66-bit security, $66 - 1 = 65$, $\frac{130}{2} = 65$, and $\frac{N(768)}{N(745)} \approx 2$).

Larger quantities of water need to be considered to fully appreciate the security offered by practically relevant cryptosystems. On average, The Netherlands enjoys a healthy daily average of 82 million cubic meters of rain (i.e., 2^{15} pools); with $65 + 15 = 80$ and $\frac{N(1130)}{N(768)} \approx 2^{14}$ it is found that 80-bit symmetric cryptosystems, 160-bit cryptographic hash functions, and 1130-bit RSA offer *rain security* or *dagelijkse neerslagverdampingsenergiebehoeftezekerheid*. Equivalently, with each German citizen boiling a cubic meter of water, one may refer to rain security as *German security* or *JedeRtausendliterbierverdampfungssicherheit*[1].

Slightly more than 2^{25} pools suffice to fill the lake of Geneva (89 cubic kilometers of water), so 90-bit symmetric cryptosystems, 180-bit cryptographic hashes, and 1440-bit RSA all offer *lake security* or *sécurité lémanique*. Boiling all water on the planet (including all starfish) amounts to about 2^{24} lakes of Geneva and leads to *global security*: 114-bit symmetric cryptosystems, 228-bit cryptographic hashes, and 2380-bit RSA. This needs to be done 16 thousand times to break AES-128, SHA-256, or 3064-bit RSA.

With all water evaporated and no bodies of water left to fire the imagination, the above new security levels have run out of steam. *Solar security* can still be defined as offered by cryptosystems that can be broken in a year requiring that year's total solar energy: 140-bit symmetric cryptosystems, 280-bit cryptographic hash functions, and 3730-bit RSA. Reaching further in an intuitively appealing manner requires a different approach.

To infinity – and beyond

There is a big gap between the energy required to bring a gram of water to a boil (as usual starting at 20°C), namely $93 \cdot 10^{-6}$ kWh, and the mass-energy of that same gram of water, namely 25 million kWh. With $\frac{25 \cdot 10^6}{93 \cdot 10^{-6}} \approx 2^{38}$, cryptanalysis suddenly looks a lot easier – maybe a bit too easy. Breaking AES-128, SHA-256, or 3064-bit RSA using the mass-energy of the lake of Geneva (where $90 + 38 = 128$) still looks more or less reasonable, but that is mostly because it is inconceivable. What about a 0.02 gram scrap of paper having the mass-energy

[1] The "2+"-bit difference between *bringing to a boil* and *Verdampfung* (which Alexander Kruppa kindly reminded us of) easily falls within the margin of Moore's eternal law which ensures that keys requiring evaporation a few years ago can now be boiled.

(namely $0.02 \cdot 25 \cdot 10^6\,\mathrm{kWh} = 500\,000\,\mathrm{kWh}$) needed to break 768-bit RSA? It implies *paper-thin security* for 1024-bit RSA: five A4 sheets of $80\mathrm{g/m}^2$ paper together weigh 25 grams, $25/0.02 \approx \frac{N(1024)}{N(768)}$, and therefore have enough mass-energy to break 1024-bit RSA. This could have been cooked up by Certicom's PR department, but does not feel intuitively right.

Finally, with the estimated mass-energy of the observable universe 2^{190} times what is required to boil two pools, $190 + 66 = 256$-bit symmetric cryptosystems, 512-bit cryptographic hash functions, and 14954-bit RSA offer *universal security*.

Summary

Determining the newly proposed security levels requires a trained professional. This considerably raises the bar for uneducated guesswork and thereby significantly diminishes the risk of insecure parameter choices. Worldwide adoption of these intuitive security levels will have a beneficial effect on Internet security.

The table lists the new[2] security levels[3], with "sea" referring to the Mediterranean Sea. For metrically-challenged prism-enabled readers it is noted that using the same numbers of gallons or cubic miles as the table's liters or cubic kilometers, respectively, requires adding two to the figures in the "symmetric" column, adding four to the figures in the "hash" column, and replacing n in the "RSA" column by the m for which $\frac{N(m)}{N(n)} \approx 4$. Note, however, that this requires longer showers, a larger lake, etc., which is only appropriate.

Table 1. Intuitive security levels.

security level	volume of water to bring to a boil	bit-lengths		
		symmetric key	cryptographic hash	RSA modulus
teaspoon security	0.0025 liter	35	70	242
shower security	80 liter	50	100	453
pool security	2 500 000 liter	65	130	745
rain security	$0.082\,\mathrm{km}^3$	80	160	1130
lake security	$89\,\mathrm{km}^3$	90	180	1440
sea security	$3\,750\,000\,\mathrm{km}^3$	105	210	1990
global security	$1\,400\,000\,000\,\mathrm{km}^3$	114	228	2380
solar security	-	140	280	3730

Acknowledgements We gratefully acknowledge usage of Wikipedia and gladly blame its contributors for any errors in our data.

[2] Unfortunately, but not surprisingly, cloud security is not well-defined.

[3] Going further down the scale than in Table 1 would lead to *meat security*, the amount of computing humans can be expected to do, but we rather see computers as meat's next step on the evolutionary ladder (`instruct.westvalley.edu/lafave/meat_in_space.html`).