PATENT NONSENSE AND THE JOINED KEYS LEMMA

ARJEN K. LENSTRA

Dedicated to Jan Karel Lenstra on the occasion of his retirement as General Director of the Centrum Wiskunde & Informatica

Abstract. This true story sketches the developments that took place as a result of a previous *liber* contribution: patent nonsense, the *Joined Keys Lemma*, and the resulting rogue authority.

Key words: JKL, chaos and whimsical inertia, rogue authority.

Cooking RSA. One of the contributions to the *liber* [2, Page 75] consists of two 438-digit numbers (n and m below) followed by a poem singing the praises of A.J. Lenstra. Upon superficial inspection, the number n looks suspicious. Indeed, writing n as the concatenation $u|\ell$ of 214 leading digits u and a 224-digit tail ℓ, and applying the transformation `01` $\rightarrow$ a, `02` $\rightarrow$ b, ... from [1], it is found that u is a Dutch sentence. It says how to use n as *RSA modulus* and that its factorization can be found "in polynomial time ...". After a search in the family archives [3, Page 132] the number m can be decrypted into a limerick befitting [2]. However, not the limerick but the RSA modulus is the subject of this story.

A regular RSA modulus is the product of two large randomly chosen prime numbers. Because the resulting leading digits cannot be expected to be coherent Dutch, the RSA modulus n was evidently not constructed in the regular fashion. A moment's thought – a moment that occurred between two of many inspirational sips meant to facilitate the creative process of above poem and limerick – learns that an RSA modulus may be constructed in an entirely different way: just fix its initial part u upfront (for instance, as done for n, in such a way that u is a Dutch sentence), randomly select a large prime p, and then craft ℓ so that $u|\ell$ becomes a large prime multiple of p. The resulting moduli do not seem to be easier to factor than regular RSA moduli (unless, as in n's case, the initial part u gives the factorization away), and they are barely harder to construct. As the construction works for any u and p, the number of different RSA moduli that share their u-value is about the same as the number of primes of size similar to p.

If one accepts the presumption that an RSA modulus consists of orderly and meaningful information, chaos, or its numerical equivalent, can now easily be dealt with: any random number may play the role of u above and can thus be turned into the leading digits of a respectable RSA modulus $u|\ell$. Because transforming chaos into order has been Jan Karel's lifelong hobby, another application of this idea is described below. First, however, a brief detour to another consequence of [2].

Patent nonsense. In RSA moduli $u|\ell$ as above, u may contain information (as in [2]) or may be random, but u may also be chosen to facilitate RSA-related calculations. For instance, for $2N$-bit RSA moduli the choice $u = 2^{N-[\log N]}$ or, at the expense of slower generation, $u = 2^{N-1}$ (i.e., blowing $N-1$ zero bits into the RSA moduli), leads to any number of RSA moduli that allow a substantial speedup and smaller footprint of the RSA engine. This is potentially useful.

Intellectual property – even when invented during off-hours – is normally speaking owned by one's employer. So, the threat of endless meetings with patent attorneys was imminent, and could only be staved off if similar prior art could be shown to exist. After all, the idea is so straightforward that it is almost inconceivable that someone had not come up with it before. That suspicion turned out to be correct. At least three independent different parties had done so much earlier: to save space and gain speed a proprietary 1984 French banking standard uses $u = 2^{N-\cdots}$, Adi Shamir had used the trick at a consulting stint, and Don Coppersmith knew about it as well. None of this prior art, however, sufficed to keep the lawyers at bay, since none of it was published. Despite vigourous attempts to unearth even the least shred of evidence, no relevant publication could be found and the unavoidable had to be faced. The rest of this part of the story can be found in US patents 6,404,890 and 6,496,929, assigned to Citibank.

Hash collisions. In August of 2004 Xiaoyun Wang showed how for any *prefix*, different values u_1 and u_2 can quickly be found so that *prefix*$|u_1|$*suffix* and *prefix*$|u_2|$*suffix* have the same MD5-hashvalue for any *suffix* (cf. [7]). Bad news, because the security of the widely used cryptographic hash function MD5 relies on *not* being able to find such *collisions*. Nevertheless, cryptographers rejoiced in the breakthrough and in the job opportunities it implied. Others inertly shrugged their shoulders. The collision-causing u_1 and u_2, though carefully constructed, have no structure that would occur in practice and are hundreds of bits long. MD5-applications could safely be left untouched; at least so it was – whimsically – believed.

Few realized that the safety margin was thin. Upon certification, RSA moduli are hashed, and RSA moduli can be made to usurp hundreds of bits of unstructured information, as shown above. Thus, for any *prefix* that one would typically use when certifying an RSA modulus, different u_1 and u_2 can be constructed such that *prefix*$|u_1$ and *prefix*$|u_2$ collide under MD5, after which ℓ is chosen so that $u_1|\ell$ is an RSA modulus. This leads to colliding c_1 =*prefix*$|u_1|\ell|$*suffix* and c_2 =*prefix*$|u_2|\ell|$*suffix* for any *suffix* that would be appropriate when certifying the RSA modulus $u_1|\ell$. A miscreant who had the sensible string c_1 certified, may claim that the certificate is valid for c_2 instead. This is not a big deal because c_2 is useless, but it is undesirable nonetheless.

JKL. The method from [2] to turn any u into an RSA modulus $u|\ell$ inspired a Chinese remaindering based extension that does the same for any pair (cf. [4]): the *Joined Keys Lemma* joins different u_1, u_2 into $u_1|\ell$ and $u_2|\ell$ that are both hard to factor. With the JKL the above strings c_1 and c_2 can be made equally useful, simply by picking an ℓ that "works" not just for u_1 but for u_2 as well. More in general, the JKL can transform any amount of chaos $\{u_1, u_2, \ldots, u_s\}$ into a set of orderly, hard to factor integers $\{u_1|\ell, u_2|\ell, \ldots, u_s|\ell\}$.

From JKL to a rogue authority. Because the JKL made actual threats conceivable (though they remained far-fetched), it should have been a sufficient argument to discontinue usage of MD5. It wasn't, but it triggered further developments that ultimately convinced most that giving up MD5 may indeed be better.

MD5-practitioners wriggled themselves out of JKL-caused colliding moduli trouble using a variety of arguments. The shared prefix, for instance, implies a single, apparently untrustworthy owner who can easily be traced if irregularities occur. It led to the more general MD5-collisions in [6] which, at considerable computational expense, allowed different prefixes. This enabled c_1's miscreant owner to target anyone as owner of c_2; it also resulted in abnormally large RSA moduli, thus obviating realistic threats.

Another argument was that the certifier includes hard to predict data such as certification time and serial number in *prefix*. Combined with the computational effort involved in constructing collision-causing u_1 and u_2, this offered adequate protection against attacks; at least so it was – still – believed.

Cryptanalysts, however, are a dogged bunch. One certification authority that happened to insert not-so-hard-to-predict data in *prefix* was incentive enough to further sharpen the knives: the RSA modulus length and computational effort were sufficiently reduced, and levelheaded persistence did the rest. The resulting *rogue certification authority certificate* – and the potential to seriously undermine internet security if even a single certification authority keeps using MD5 – is described on [5]. It should be noted that JKL, though crucial in the developments leading to the rogue authority, plays no role in the latter: [2] suffices.

References

[1] M. Gardner, *Mathematical games, A new kind of cipher that would take millions of years to break*, Scientific American, August 1977, 120–124.

[2] R. Koning, H.W. Lenstra, J.K. Lenstra, T.J. Wansbeek, *Andries en zijn kameraden*, ISBN 90-9011507-2, March 1998.

[3] A.K. Lenstra, *Polynomial-time algorithms for the factorization of polynomials*, Ph.D. thesis, 1984.

[4] A.K. Lenstra, X. Wang, B.M.M. de Weger, *Colliding X.509 certificates*, March 2005, `http://eprint.iacr.org/2005/067/`.

[5] A. Sotirov, M. Stevens, J. Appelbaum, A.K. Lenstra, D. Molnar, D.A. Osvik, B.M.M. de Weger, *MD5 considered harmful today, creating a rogue CA certificate*, 25th annual Chaos Communication Congress, Berlin, December 2008, `http://www.win.tue.nl/hashclash/rogue-ca/`.

[6] M. Stevens, A.K. Lenstra, B.M.M. de Weger, *Chosen-prefix collisions for MD5 and colliding X.509 certificates for different identities*, Proceedings EuroCrypt 2007, Springer LNCS 4515 (2007), 1–22.

[7] X. Wang, H. Yu, *How to Break MD5 and Other Hash Functions*, Proceedings EuroCrypt 2005, Springer LNCS 3494 (2005), 19–35.