RSA LABORATORIES

OVERVIEW
STAFF & ASSOCIATES
RESEARCH AREAS
STANDARDS INITIATIVES
HISTORICAL
CRYPTO FAQ
RSA ALGORITHM
CRYPTOGRAPHIC CHALLENGES
CRYPTOBYTES TECHNICAL NEWSLETTER
TECHNICAL NOTES AND REPORTS
ENHANCING ONE-TIME PASSWORDS FOR PROTECTION AGAINST REAL-TIME PHISHING ATTACKS
SHA1 COLLISIONS CAN BE FOUND IN 2^63 OPERATIONS
HOW THESE DISCOVERIES AFFECT RSA SECURITY'S PRODUCTS: FREQUENTLY ASKED QUESTIONS
HASH FUNCTION UPDATE DUE TO POTENTIAL WEAKNESS FOUND IN SHA-1
RECENT IMPROVEMENTS IN THE EFFICIENT USE OF MERKLE TREES: ADDITIONAL OPTIONS FOR THE LONG TERM
TWIRL AND RSA KEY SIZE
RAISING THE STANDARD FOR RSA SIGNATURES: RSA-PSS
RSA LABORATORIES SUBMITS NEW AES MODE TO NIST
HAS THE RSA ALGORITHM BEEN COMPROMISED AS A RESULT OF BERNSTEIN'S PAPER?
RSA SECURITY RESPONSE

TWIRL AND RSA KEY SIZE

Burt Kaliski, RSA Laboratories
Revised May 6, 2003

Executive Summary

The popular 1024-bit key size for RSA keys is becoming the next horizon for researchers in integer factorization, as demonstrated by the innovative "TWIRL" design recently proposed by Adi Shamir and Eran Tromer. The design confirms that the traditional assumption that a 1024-bit RSA key provides comparable strength to an 80-bit symmetric key has been a reasonable one. Thus, if the 80-bit security level is appropriate for a given application, then TWIRL itself has no immediate effect. Many details remain to be worked out, however, and the cost estimates are inconclusive. TWIRL provides an opportunity for review of key sizes in practice; RSA Laboratories' revised recommendations are given in Table 1 below.

Introduction

The security of RSA keys of particular sizes has been of theoretical and practical importance for many years, due both to the academic interest in the underlying problem of *integer factorization* and the widespread use of such keys for data security. Various estimates of the security for different key sizes have been developed, based both on experimental evidence and extrapolations. This has helped to establish recommendations for minimum key sizes, just as with other algorithms.

One result of the work on RSA key sizes is the traditional comparison, found in many standards documents, of 1024-bit RSA keys to 80-bit symmetric keys. This comparison helps to harmonize key size and algorithm selections. If an application requires security greater than or equal to that provided by 80-bit symmetric keys, then appropriate choices for that application would include algorithms such as triple-DES (112-bit keys), AES (128-bit or greater keys), SHA-1 (160-bit hash value with 80-bit security against collisions), and 1024-bit RSA, among other algorithms. (See Note 1). Such a comparison has recently been documented in the key size schedule proposed by NIST [NIST03].

For a rough estimate for the security of 80-bit symmetric keys, one may consider the 1996 report on symmetric key sizes by Matt Blaze *et al.* [BDR+96]. The report estimates that using 1996 technology, a $10 million machine could search for a single 56-bit DES key in six minutes. Applying Moore's Law that hardware speed per dollar doubles every eighteen months, a machine with the same cost would take about 14 seconds today. An 80-bit key search today (assuming the same complexity per key as DES) would take 2^{24} times longer, or about seven years, a figure that is also in line with data in the NSA report on AES hardware [WBRF00] updated for present technology (see Note 2).

Arjen Lenstra and Eric Verheul's methodical estimates [LV01] give quite similar results for the security of 1024-bit RSA keys. In one model, they project that in the year 2009, a machine costing about $250 million could factor a 1024-bit RSA key in a day — so a $10 million machine would take just under a month. Working the numbers back to the year 2003 requires one to "undo" eight doublings, four due to Moore's Law and four due to anticipated improvements in methods for integer factorization. The $10 million machine would take about 18 years today under this model. In the year 2006, a machine with this cost would take just over one year.

The comparison of 1024-bit RSA keys with 80-bit symmetric keys has so far been based on rough extrapolations of the running times for smaller keys. It has generally minimized the impact of other factors such as memory size, which can become substantial as RSA key sizes increase, particularly in software implementations [Sil01]. Moreover, the previous estimates have primarily focused on software implementations, with limited attention to hardware speedups. Recent research has thus aimed to estimate the cost of a hardware implementation more accurately, taking into account all these factors.

TWIRL

TWIRL is an acronym for *The Weizmann Institute Relation Locator*, a new hardware design for integer factorization developed by Shamir and Tromer [ST03]. Like most approaches to factoring general integers, it is based on the Number Field Sieve, which consists of two steps:

Sieving step (a.k.a. "relation location"): searching a large *sieving region* for *relations* with respect to a *factor base*
Matrix step: solving for a linear dependency among the relations, which yields the factors of the RSA key

FLAW IN NYBG SIGNATURE SCHEME (NSS)

COUNTERMEASURES AGAINST BUFFER OVERFLOW ATTACKS

RECOMMENDATIONS ON ELLIPTIC CURVE CRYPTOSYSTEMS

OVERVIEW OF ELLIPTIC CURVE CRYPTOSYSTEMS.

RSA HARDWARE IMPLEMENTATION, TR 801

BULLETINS

Shamir and Tromer estimate that using TWIRL, one could complete the entire sieving step for a 1024-bit RSA key in less than a year with only $10 million worth of hardware. They also give a more conservative design that they estimate would cost $50 million.

The primary reason for the impressive performance claimed for TWIRL is a remarkably clever hardware design. In traditional software-based sieving, the CPU does all the work while the memory is idle most of the time. Motivated by Daniel Bernstein's recommendation to replace memory with active processors in hardware implementations of integer factorization [Ber01], TWIRL lets the memory (augmented with some logic) do the work. TWIRL is thus able to exploit very effectively the massive parallelism inherent in the sieving step, leading to a factor of 1000 or more performance improvement in addition to the ordinary speedup one would expect by switching to custom hardware (see Note 3). The hardware design of course requires further review by hardware designers and researchers in integer factorization, but seems plausible at this stage, though a significant engineering challenge.

The cost and time estimates given in the TWIRL paper, however, are based on simple extrapolations, which the paper acknowledges are "ferocious" — not precise analysis or actual experimental data. The analysis employs a new tradeoff between the size of the factor base and the size of the sieving region, one that would also apply (if correct) to software-based implementations. The tradeoff is crucial, because although the total amount of searching in the sieving step grows in proportion to the size of the sieving region, the size of each sieving circuit depends on the size of the factor base. Such tradeoffs are notoriously difficult to analyze as there are so many parameter choices in the Number Field Sieve. Further research is needed to confirm whether the estimates are correct, and it is possible that the correct sizes will be significantly higher. This is true for both the $10 million and the more conservative $50 million designs. (Some initial research along these lines was just posted by Lenstra et al. [LDHL03].)

Since a full implementation of the Number Field Sieve requires more than just sieving circuits, and since the "ferocious" size estimates in TWIRL may be optimistic, the cost of actual hardware based on the TWIRL design is probably higher than the initial $10 million / $50 million estimates. However, further improvements are certainly possible over time. The cost of such a machine, assuming one could be built over the next few years, therefore might not be far from the expectation for the 80-bit symmetric security level when the machine is built. In this sense, TWIRL may be viewed as an affirmation of both the 80-bit security level and Lenstra and Verheul's projection about what might be possible as both hardware speed and integer factorization methods improve.

Significantly shorter keys such as the historic 512-bit RSA key size are essentially compromised by TWIRL (if not already overcome by previous methods). Longer keys, in particular 2048-bit keys, still provide a significant security margin as both the sieving region and the factor base are substantially larger (see Note 4). The TWIRL paper does not give any estimates for 2048-bit RSA keys, and an accurate estimate would need analysis similar to what remains to be done for 1024-bit keys. Previous work based on running time only suggests that 2048-bit keys are about 2^{32} times harder than 1024-bit RSA keys, i.e., at the 112-bit security level.

Impact on 1024-bit RSA Keys

What does this all mean for 1024-bit RSA keys?

Since TWIRL confirms the presumed 80-bit security level for such keys, if the 80-bit security level is appropriate for a given application, then TWIRL itself has no immediate effect. But if one were counting on an extra security margin for 1024-bit RSA keys beyond the 80-bit security level, that expectation has been diminished.

Over time, it should be assumed that machines like TWIRL will be built, and that the performance of such machines may improve as further optimizations are found. Thus, the 1024-bit RSA key size will eventually no longer be sufficient, just like the key sizes that preceded it (56-bit symmetric, 512-bit RSA, etc.) The main question is when to make a transition to larger key sizes.

Key Size Recommendations

RSA Laboratories has from time to time provided key size recommendations, primarily for the RSA algorithm. Eight years ago, in the Summer 1995 issue of *CryptoBytes*, we recommended a minimum key size of 768 bits for user keys, 1024 bits for enterprise keys and 2048 bits for root keys, a practice that has been generally reflected in the industry with the exception of legacy support for "exportable" key sizes such as 512 bits. These recommendations were made without any specific "lifetime" for the data, so would need to be updated periodically. Our current recommendation, consistent with industry standards, is a minimum of 1024 bits for general data, but again without any specific lifetime.

NIST's recently proposed schedule of key sizes does take into account the lifetime of the data, spanning the next several decades [NIST03]. The schedule, still a draft, suggests that the 80-bit security level (i.e., 1024-bit RSA keys) is appropriate for protecting data through the year 2015, and that the 112-bit security level is appropriate through the year 2035. (See Note 5).

Assuming that Moore's Law continues to hold for eight more generations, and starting with estimates based on the Blaze *et al.* report above, it would take a $10 million machine 10 days with year 2015 technology to search for an 80-bit key — which even in 2015 should still be a lot of money for most keys. However, many keys will have greater value, and key size recommendations have a history of taking longer to be fully embraced than one might prefer (consider the lengthy process of upgrading DES). Accordingly, an earlier transition would seem prudent, consistent with the higher security level of 90 bits encouraged by the Blaze *et al.* report [BDR+96] for protecting data through that time period.

The next level in NIST's schedule is the 112-bit security level, matching triple-DES encryption. To put the 112-bit security level in concrete terms, some simple calculations may be done. Starting with the estimates for 80-bit key search today, a 112-bit key search today on a $10 million machine would take about 30 billion years. A machine with the same cost in the year 2030 — 18 generations from now — would take over 100,000 years to do a 112-bit key search. (There is clearly dispute over whether Moore's Law will hold that long, so this is only a starting point for analysis.)

Taking the previously established correspondence between 2048-bit RSA keys and the 112-bit security level as a starting point, one may assume that a "future TWIRL" in 2030 would likewise take 100,000 years to factor a 2048-bit RSA key. It could take more time, due to the larger circuit size. More likely, it would take less, as there may be further improvements in integer factorization. Conservatively applying Lenstra and Verheul's "law", i.e., incorporating 18 "generations" of such improvements, a $10 million "future TWIRL" in the year 2030 would take about five months to factor a 2048-bit RSA key. This brings us essentially back to TWIRL's initial claims for 1024-bit RSA keys today.

The 112-bit security level is somewhat higher than needed now, but it is convenient since triple-DES is already widely implemented,

Protection Lifetime of Data	Present – 2010	Present – 2030	Present – 2031 and Beyond
Minimum symmetric security level	80 bits	112 bits	128 bits
Minimum RSA key size	1024 bits	2048 bits	3072 bits

Table 1. Recommended minimum symmetric security levels and RSA key sizes based on protection lifetime.

These recommendations are similar to NIST's proposed schedule except that the transitions to larger key sizes are in the years 2010 and 2030. As that schedule is further developed based on input from the research community, these recommendations may need to be updated accordingly. Moreover, the recommendations are general minimums. A larger minimum may well be appropriate for enterprise and root keys, a distinction that is already reflected in practice through the common use of 2048-bit root keys today.

Conclusion

TWIRL confirms that the 80-bit security level has been a reasonable assumption for 1024-bit RSA keys, and serves as a reminder that 80-bit security will not last indefinitely. If the industry has learned anything about key size over recent years, it's that we need to plan ahead for transitions. A transition toward the 112-bit security level and larger RSA keys is recommended over the remainder of this decade. When the horizon of 1024-bit RSA keys is finally reached, such keys will hopefully already be in the past.

NOTES

1. In NIST's schedule, and as a general principle, the choice of RSA key size and the choice of symmetric algorithms are independent, both being driven by the desired security level. A 128-bit AES key may be used with a 1024-bit RSA key if the 80-bit security level is adequate for an application, since both choices meet or exceed that security level. Similarly a 128-bit AES key may be used with a 2048-bit RSA key if the 112-bit security level is adequate. A 128-bit AES key does not require a 3072-bit RSA key.
2. The smaller of the pipelined Rijndael implementations developed by NSA during the AES process [WBRF00] requires about 4 million transistors and occupies an area of about 300mm^2. (Rijndael is the algorithm subsequently selected for the AES.) The implementation achieves a throughput of about 4 Gbit/second or 32 million encryptions/second. With the same wafer technology as in TWIRL, the implementation would occupy an area of about 10mm^2 and would thus run about 30 times faster (using the rule of thumb that clock speed is inversely proportional to feature area). A single AES key search circuit using this technology would thus presumably have a throughput of nearly 1 billion ~ 2^{30} encryptions/second. Key setup is a relatively small overhead in this Rijndael circuit, so the key search rate should be similar.
   Each of the 600 wafers in the $10 million TWIRL budget is capable of holding about 7000 of these key search units. The 600 wafers could thus search about 2^{52} keys/second or 2^{77} keys/year. Applying the $50 million figure for the more conservative TWIRL budget, the comparable AES search space would be nearly 2^{80}.
3. Interestingly, each sieving circuit in TWIRL takes roughly the same amount of silicon as an Intel® Xeon™ processor [Intel] plus 1Gbyte memory, yet searches perhaps 20,000 times faster than software on that processor would due to the parallelism and the custom hardware. Custom hardware generally gives a significant speedup — consider the difference between the DES Cracker and ordinary PCs — but TWIRL's speedup is even more dramatic because of the inherent parallelism of the sieving step.
4. Alternate designs for sieving circuits whose size does not depend on the size of the factor base were revisited recently by Bernstein [Ber01]. Such circuits hold promise for very large key sizes, but it is not clear yet if they have any benefit for key sizes of current interest. Further work along these lines is warranted in assessing the cost of factoring 2048-bit and larger RSA keys.
5. "Through the year Y" means that the protection of the particular data is needed through that year. For encryption, this means that an application needs the assurance it will be infeasible to decrypt messages that have been encrypted with a given public key through year Y. For signatures, this means that it will be infeasible to forge signatures that can be verified with a given public key through that year. In many cases a much shorter lifetime is adequate. For instance, in network authentication with an ephemeral signature key, the only assurance required is that it should be infeasible to forge signatures before the signature key expires (i.e., the adversary has only a few hours total). Keys obviously will not suddenly become insecure at any given year, but instead their strength will slowly erode over time.

REFERENCES

[Ber01] Daniel Bernstein. *Circuits for Integer Factorization: A Proposal.* November 9, 2001. Available via http://cr.yp.to/papers.html.

[BDR+96] M. Blaze, W. Diffie, R. Rivest, B. Schneier, T. Shimomura, E. Thompson and M. Wiener. *Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security.* Report of ad hoc panel of cryptographers and computer scientists, January 1996. Available via http://www.crypto.com/papers/.

[LV01] Arjen K. Lenstra and Eric R. Verheul. Selecting cryptographic key sizes. *Journal of Cryptology* 14(4):255–293, 2001. Available via http://citeseer.nj.nec.com/lenstra99selecting.html.

[Intel] Intel Corporation. *Intel® Microprocessor Quick Reference Guide.* http://www.intel.com/pressroom/kits/quickreffam.htm.

[LDHL03] Arjen K. Lenstra, Bruce Dodson, James Hughes and Paul Leyland. *Factoring Estimates for a 1024-bit RSA Modulus*. April 28, 2003.

[NESSIE03] NESSIE Consortium. *Portfolio of Recommended Cryptographic Primitives*. February 27, 2003. Available via http://www.cryptonessie.org/.

[NIST03] NIST. *Special* Publication *800-57: Recommendation for Key Management. Part 1: General Guideline.* Draft, January 2003. Available via http://csrc.nist.gov/CryptoToolkit/tkkeymgmt.html.

[ST03] Adi Shamir and Eran Tromer. *Factoring Large Numbers with the TWIRL Device*. Draft, February 9, 2003. Available via http://www.wisdom.weizmann.ac.il/~tromer.

/CryptoToolkit/aes/round2/r2anlsys.htm.

REVISION HISTORY

May 1, 2003: Initial version
May 6, 2003: Added reference to [LDHL03]

Top of Page