X-Authentication-Warning: teak.ii.uib.no: larsr owned process doing -bs
Date: Mon, 15 May 2000 14:37:52 +0200 (MET DST)
From: Lars Ramkilde Knudsen <Lars.Knudsen@ii.uib.no>
To: AESRound2@nist.gov
cc: Lars Ramkilde Knudsen <Lars.Knudsen@ii.uib.no>
Subject: recommendation

Dear NIST,

Please find document with recommendations for the AES by Haavard Raddum and myself.

Lars R. Knudsen, Assoc.Prof., Univ. of Bergen, Dept.of Informatics,
PB 7800, N-5020 Bergen, Norway +47 55 58 41 57, (fax +47 55 58 41 99),
Lars.Knudsen@ii.uib.no, http://www.ii.uib.no/~larsr/

Recommendation to NIST for the AES

Lars R. Knudsen and Håvard Raddum
Dept. of Informatics, University of Bergen, Norway

May 15, 2000

Abstract

In this note we give recommendations to NIST on how to proceed in the selection of the algorithm(s) for the Advanced Encryption Standard.

Efficiency and Security

In the request for candidate algorithm nominations for the AES NIST says "For interoperability and other purposes, NIST strongly desires to select a single block encryption algorithm to be specified in the AES with a strength equal to or better than that of Triple DES and significantly improved efficiency."[1].

We think it is fair to say that all of the five final candidates are significantly more efficient than Triple DES. Therefore we recommend that NIST do not pay too much attention to speed issues when selecting the final algorithm(s). Personally we think the performances of the candidates have received far too much attention at the AES conferences.

Performance will only increase

It is naive to think that one can construct encryption algorithms which give very high security, which we presume is what people/NIST wants, and at the same time operate at very high speeds.

NIST asks for security levels equal to or better than that of Triple DES. Assuming we are talking about three-key triple-DES, we think that the next 5-10 years will show that several of the final five candidates, if chosen, will not meet this requirement. We say, "if chosen", since in that case the algorithm will get the focus of most or all cryptanalysts and better results are bound to emerge.

Security will only decrease

We urge NIST to be conservative in the choice of the AES. The DES has been a huge success. The AES will probably be a success too, but likely a lesser success than the DES. Today there are many more alternatives to a NIST encryption standard than in the late seventies. To make AES a success NIST should ensure that it will not be broken in the next decade or two (or perhaps three). By "broken" we mean broken in a theoretical sense (faster than an

exhaustive search for the key). If in, say, 3 years someone comes up with a theoretical attack on AES, it might remove people's attention from the AES and towards one of the other encryption standards which can be expected to emerge.

Which algorithms?

In this section we comment on the five final candidates.

Serpent is undoubtedly the safest bet for a secure algorithm today and in 20-30 years. The only criticism there has been of Serpent is that it is too slow in software. However, as mentioned above, the differences in speed between the five candidates in software are small. Security comes as a cost, and yet Serpent is considerably faster than Triple DES. Also, it is the only of the five candidates, where we have no problems in assessing the security as "at least as secure as Triple DES".

MARS is a complex algorithm. Having many exors, additions and rotations does not necessarily give a strong cipher since it yields many dependencies between the intermediate ciphertext values. It is very hard to estimate what the consequences of the combined operations are. For example, Robshaw and Yin have found two perfect linear approximations through the R strand in the E-function, there are approximations through the S-box with higher bias than the MARS team thought existed, and the S-box does not have all the properties they required. We think there could be other weaknesses in MARS that are not yet discovered. Also, MARS is too large to fit on all but the high-end smart-cards, making it difficult to use in some applications.

RC6 is the candidate with the easiest-to-remember description except for the key-schedule. It is an extension to 128-bit blocks of the 64-bit block cipher RC5. While cryptanalytic attacks on RC5 revealed problems, these have been sought solved in introducing an additional confusion function ($f(x) = x(2x+1)$). The security of RC6 seems to rely heavily on the data-dependent rotations. The weakness in these have been exploited in the attacks of Knudsen-Meier and Gilbert-Handschuh-Joux-Vaudenay. It is hard to estimate the implication on the security of RC6 from these attacks.

Rijndael is probably the most elegant of the final five candidates. However, it is also the candidate for which the safety margin with respect to the known attacks is smallest. If Rijndael becomes the AES and remains unbroken for 20-30 years, this would be a tribute to science. However, we also think that Rijndael is the one algorithm of the five which is most vulnerable to a devastating attack. All elements of the algorithm can be described as simple mathematical functions, including the S-boxes, and there are no "randomly looking" elements. However, as of today, Rijndael with 10 rounds resists known attacks and NIST could give the authors the benefit of the doubt. Rijndael should only be chosen if multiple algorithms are considered, e.g. in an 'OR' standard.

Twofish is a complex algorithm. It borrows elements from Square and Rijndael (the MDS-based linear transformation), from Khufu (the key-dependent S-boxes), and from SAFER (the PHT-transform). In addition to this, there is a

mixed use of the group operations '+' (addition modulo 2^{32}) and '$\oplus$' exclusive-or), and two one-bit rotations introduced to destroy the byte structure. We feel that the many different components of Twofish makes it hard to become convinced of the strength of the algorithm.

We recommend that NIST picks Serpent as the Advanced Encryption Standard

References

[1] NIST.
`http://csrc.nist.gov/encryption/aes/pre-round1/aes_9709.htm`