The State Of The Art In Algorithmic Encryption

A. Wiesmaier
Department of Cryptography and Computeralgebra
Technische Universitt Darmstadt
Hochschulstr. 10
D-64283 Darmstadt, Germany
wiesmaie@cdc.informatik.tu–darmstadt.de

January 2, 2006

1 Introduction

In most cases the security of a given encryption algorithm is not mathematically proved, but is based on commonly agreed security assessments and on the robustness against known attacks. The more an algorithm is analyzed the more trustworthy it is considered. Most algorithms are disclosed and described in norms and specifications. Unfortunately there is no guarantee that unexpected advances in cryptanalysis weaken or even break an algorithm. It is very important to be prepared with alternative systems as the probability that all of them break at the same time significantly lowers with the number of alternatives. The security of a given cipher text depends highly on the secrecy of the decryption key. Generally longer keys provide better security (particularly against brute force attacks) but usually decrease the performance. The longer the term in which the cipher text must stay secret the longer is the required key size.

2 Symmetric Encryption

Symmetric (secret key) encryption schemes use the same key for encryption and decryption and usually have predefined key lengths. They provide a high security and a high performance, but suffer from the key exchange problem. A group of n entities needs to exchange $\frac{n(n-1)}{2}$ different keys over secure channels.[1] The current state of the art in symmetric encryption is surely given by the five finalists of the AES selection process. Following Nechvatal et al. [12] they all offer adequate security and perform reasonably. We pick the winner of the AES selection process, namely Rijndael, as their representative. Rijndael comes with three possible key lengths (128bit, 192bit and 256bit), provides a

[1] This means the keys must be exchanged in a secure out–of–band way, which is especially a problem if the entities are located at different places.

very high security and very fast soft- and hardware implementations. Due to Aoki and Lipmaa [5, p. 7] Rijndael–128 is able to encrypt a 128bit block within 237 cycles on a 450 MHz Pentium II. This leads to a throughput of 243 Mbit/s. Lipmaa [10] claims to have a Rijndael library which nearly reaches 1.5Gbit/s on a 3.06GHz Pentium IV. Hodjat and Verbauwhede [4, p. 1] report about a Rijndael hardware implementation which reaches a throughput of up to 21.54Gbit/s. Following Schneier et al. [6, p. 1] Rijndael encrypts 20% slower for 192bit keys and 40% slower for 256bit keys. According to Lenstra [8, pp. 13–14] a 128bit symmetric cipher is supposed to be secure against mathematic attacks until at least 2090 (192bit until 2186, 256bit until 2282).[2] The estimates from ECRYPT [7, p. 28] are done much more carefully. They estimate 128bit keys to be secure until 2035. 256bit keys are supposed to be secure within the "foreseeable future" which explicitly includes quantum computers. Buchmann [3, pp. 91–92] reports about the "Vernam–One–Time–Pad" which is mathematically proven unbreakable. But its heavy requirements regarding the keys make it unusable in normal practice.[3]

3 Asymmetric Encryption

Asymmetric (public key) encryption schemes use different keys for encryption and decryption and usually have arbitrary key lengths. It is important that it is practically impossible to compute the decryption key from the encryption key. This reduces the number of different key–pairs in a group of n entities to n. The need for out–of–band mechanisms for key exchange can be avoided by utilizing public key infrastructures (PKIs). Unfortunately PKIs are relatively complicated constructions and bring their own problems with them. Public key systems are based on general mathematical problems which are (supposed to be) relatively hard.[4] Public key algorithms have complex mathematics and need very long keys. Due to this public key cryptography is very much slower than secret key cryptography and needs times which are some orders of magnitude over those of Rijndael. Due to this public key encryption is normally only used in hybrid encryption systems. Thereby the entities use the public key systems to exchange a secret key. This exchanged key is then used to encrypt the actual message with a symmetric encryption system. In opposite to symmetric systems the encryption performance of asymmetric systems may significantly differ from its decryption performance. The first invented public key encryption system RSA [11] is still the most used one. It is based on the factorization problem. According to Lenstra [1, p. 6] RSA currently[5] needs a modulus size[6] somewhere between 2790bit and 3390bit to meet the security of a 128bit Rijndael encryption. Rijndael–192 security is reached by a modulus size somewhere between

[2] This key length computations are done on Keylength.com [9] which claims (and seems) to use the estimations from [8].

[3] The key must have the same length as the plaintext and can be used only one time.

[4] e.g. factorization

[5] This will worsen significantly over the time.

[6] Which is an upper bound for the length of the decryption key.

7160bit and 8200bit. Rijndael–256 security implies an RSA modulus between 14200bit and 15800bit.[7] ECRYPT [7, p. 20] estimates RSA keys with the length of 3072, 7680 and 15360 offer equivalent security to Rijndael 128, 192 and 256. The most prominent alternative to RSA is elliptic curve cryptography (ECC). It is based on the discrete logarithm problem and is faster than RSA because it manages with shorter keys. Due to the table form Lenstra and Verheul [2, p. 32] the security of 1024bit RSA is met by an ECC key between 138bit and 147bit. ECRYPT [7, p. 20] estimates a 160bit ECC key provides RSA–1024 security. All widely used public key cryptosystems are broken by efficient algorithms for sufficiently large quantum computers. There is some research on quantum–safe public key cryptosystems in order to meet this threat.

References

[1] A. Lenstra, *Unbelievable Security*, 2001, http://www.win.tue.nl/~klenstra/aes_match.pdf (Oct. 2005).

[2] Arjen K. Lenstra and E. Verheul, *Selecting Cryptographic Key Sizes*, 2001, http://citeseer.ist.psu.edu/287428.html (Oct. 2005).

[3] J. Buchmann, *Einführung in die Kryptographie*, Springer, 2001, ISBN: 3-540-41283-2, also available in English ISBN: 0-387-21156-X.

[4] A. Hodjat and I. Verbauwhede, *A 21.54 Gbit/s fully pipelined AES processor on FPGA*, Field–Programmable Custom Computing Machines 2004 (FCCM'04), 12th Annual IEEE Symposium, pages 308 - 309, http://www.ee.ucla.edu/~ahodjat/AES/hodjat_fccm.pdf (Oct. 2005)

[5] K. Aoki and H. Lipmaa, *Fast Implementations of AES Candidates*, in proceedings of "The Third Advanced Encryption Standard Candidate Conference", 2000, pages 106–120, http://citeseer.ist.psu.edu/aoki00fast.html (Oct. 2005)

[6] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall and N. Ferguson, *Performance Comparison of the AES Submissions*, Proc. Second AES Candidate Conference, NIST, 1999, pp. 15-34, http://citeseer.ist.psu.edu/651823.html (Oct. 2005)

[7] ECRYPT, *ECRYPT Yearly Report on Algorithms and Keysizes (2004)*, 2005, http://www.ecrypt.eu.org/documents/D.SPA.10-1.1.pdf (Oct 2005)

[8] A. Lenstra, *Key Length*, Contribution to "The Handbook of Information Security", 2004, http://cm.bell-labs.com/who/akl/key_lengths.pdf (Oct. 2005)

[7] The stated modulus sizes are linear interpolated between the values for 2001 and 2010.

[9] Keylength.com, *Cryptographic Key Length Recommendation*, Website, http://www.keylength.com/ (Oct. 2005)

[10] H. Lipmaa, *Fast Implementations of AES and IDEA fro Pentium 3 and 4*, Website, http://home.cyber.ee/helger/implementations/ (Oct. 2005)

[11] RSA Security, *PKCS #1: RSA Cryptography Standard*, Website, http://www.rsasecurity.com/rsalabs/node.asp?id=2125 (Oct. 2005)

[12] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti and E. Roback, *Report on the Development of the Advanced Encryption Standard (AES)*, Journal of Research of the National Institute of Standards and Technology, Volume 106, pp. 511–576, http://nvl.nist.gov/pub/nistpubs/jres/106/3/j63nec.pdf