# Possible NSA Decryption Capabilities

**Date:** 29 June 1999

**Version:** Draft 1.0 for Discussion

**John R T Brazier**

## Introduction

This document estimates the possible capabilities of the NSA in breaking certain types of cipher by exhaustive key search. It is a theoretical document, and only covers technical ability: in many cases the NSA will use alternative methods to break target cipher systems. The aim of this document is to give some estimates for the security of certain key lengths for some symmetrical (block) ciphers.

This is a draft discussion document; it still requires more work before being generally released.

## Acknowledgements

This paper could not exist without the work carried out by the EFF in the construction of their DES cracker. Much of this work is based on their information and experience.

## Constraints and Assumptions

Theoretical research such as this always has to be based on some constraints and assumptions. Basic ones are listed here; others are discussed in more detail in the main body of the paper.

1) It is assumed that the NSA knows the cipher system in use, and can achieve a ‘known plaintext’ or ‘known to be ASCII’ attack on the cipher system.
2) The attack is a brute force keysearch, examining all keys until the one is found that produces the known plaintext from the cyphertext. The key can then be used to decrypt other cyphertexts.
3) The cipher systems studied are symmetrical block ciphers. Public key systems such as RSA are not included.

## ‘Cracking DES’: The EFF Machine

The starting point for this study is the DES-cracking machine that has been built by the Electronic Frontier Foundation, and has been fully described by their publication [EFF]. The key points of their publication are outlined below, and overall their device may be regarded as being made up of a PC, as controlling computer, plus massively replicated components for a ‘highly parallelized’ architecture. This architecture is the basis of this paper.

### *1. Search Unit*

The basic unit is a ‘search unit’, which can be loaded with information about the plaintext, the first 16 bytes of the associated cyphertext, its starting key value, and any initial vector (for the cipher block chaining, or CBC, mode of DES operation). It has some other registers that allow the unit to search for partially known plaintexts (ie in the situation where it is known that the plaintext is ASCII, but not what the letters actually are) and to investigate certain specialized challenges.

The search unit can then step through all lower 32 bits of a key, testing the plaintext against cyphertext. It runs at 40MHz, doing one decryption every 16 clock cycles, and so will test 2.5 million keys per second. When it has stepped through all 32 low bits of the key, it halts until the controlling computer reloads a new key and the unit continues. This happens about every half-hour.

Periodically, the unit will register an apparent match: a candidate solution. Many of these will be 'false positives' because of the slightly odd way the unit matches the plaintext against cyphertext (it only does an 'approximate match' for speed). The search unit halts and the controlling computer fully checks out the candidate. Because the number of false positives is relatively low, the controlling computer is not a bottleneck in this process.

### *2. Main Chip*

The main chip contains 24 search units. They share the plaintext and cyphertext information, but each has its own key register as it searches its own part of the keyspace. There are address and data wires, power and ground lines, and also signal wires to indicate when a unit has stopped: this happens when there is a candidate solution or when the unit needs a new starting key. Thus each chip will search 2.5 x 24 = 60 million keys per second.

### *3. Board*

Each board has 64 search chips along with support circuitry, lights showing chip status and parallel interfaces to the PC. Each board can thus try $3.84 \times 10^9$ keys per second.

### *4. Overall Device*

There are 24 boards, fitted into 2 chassis of twelve boards each. Thus the overall machine will check $9.216 \times 10^{10}$ keys per second.

DES has a keyspace of $2^{56}$ keys; on average a search will find the correct key when $2^{55}$ keys have been tested. Thus as $2^{55} = 3.603 \times 10^{16}$, then the cracker will on average find a key in $3.603 \times 10^{16} / 9.216 \times 10^{10} = 390{,}937$ seconds, or 4.52 days.

### *5. Costs*

The costs of the DES breaker are given as follows:

1. Design was $80,000.
2. All construction and materials costs, including chips, boards, chassis, power supplies and PC were £130,000.

Assuming that the costs can be averaged out, and ignoring the cost of the PC (which is minor), we can produce the following cost estimates for the construction of the machine (excluding design):

| Item | Cost |
|---|---|
| Whole Machine | £130,000 |
| Board | $5,417 |
| Chip | $84.6 |
| Search Unit | $3.53 |

Naturally, the costs are 'all in', so the cost per search unit includes its proportion of manufacturing, board assembly, the chassis, the power unit, and so forth.

## Design for a New Machine and Its Costs

Let us now postulate that some large organisation such as the NSA wishes to produce a high-performance cracking device. This machine should be able to attack a modern, effective crypto system that is likely to be a DES replacement, or is similar to one. One possibility is the RC5 and RC6 family. RC5 was developed by Ron Rivest in 1994 [RLR], and RC6 by Rivest et al for the Advanced Encryption Standard [RRS].

### *1. The RC5 Algorithm*

This algorithm is heavily backed by RSA Laboratories, perhaps the premier crypto design and licensing company in the world. It is the basis of a very public series of cipher-breaking challenges offered by RSA Labs. These challenges get progressively harder due to a unique feature of RC5: it is a 'parametized' crypto system, and so is adjustable in its strength.

RC5 is, in fact, a family of algorithms, and should correctly be specified as RC5-*w*/*r*/*b*, where each of the parameters are:

- *w*: the word size in bits (so usually 32, but 16 and 64 are legal);
- *r*: the number of rounds (currently 12 recommended for $w = 32$, although 0 to 255 are legal; note that an RC5 round is equivalent to two DES rounds);
- *b*: the keylength in bytes (0 to 255 are legal, this parameter is discussed further below).

For typical modern computers, the optimum word size is 32, and most people opt for the recommended 12 ('double') rounds. So RC5 typically becomes RC5-32/12/*b*, with the key length being the variable factor in increasing the strength of the algorithm family. RSA Labs challenges go from RC5-32/12/5 all the way to RC5-32/12/16, with the 5, 6 and 7 byte keys having been broken, and RC5-32/12/8 under attack by Distributed.net's world-wide distributed keybreaking effort [DIN].

RC5 has been very heavily analysed; whilst there have been some theoretical observations about its effectiveness, and a suggestion that 16 rounds rather than 12 should be used, no real attack against RC5 has been mounted except for exhaustive keysearch. Thus RC5 is an excellent model on which to base a keycracking machine because it is meaningful to speak of an 80-bit or 96-bit key for RC5, and the length of the key is the measure of the algorithm's difficulty (all other factors being equal).

However, RC5 is unlikely to be of interest to the NSA, for the simple reason that it is not one of the algorithms proposed for the Advanced Encryption Standard (AES). However RC6 is, and thus may well become the standard algorithm (and thus a primary NSA target).

## *2. The RC6 Algorithm*

The RC6 algorithm is a 'son of RC5'. Although based on RC5, a new primitive operation has been added (integer multiplication), and other modifications made to increase the diffusion achieved in each round (the improvements have come from analyses of RC5). The other main difference is that there are four registers rather than two: the cipher operates on 128-bit blocks of data at a time.

Otherwise, RC6 is a family of algorithms in exactly the same way as RC5, and these are specified in the same way, as RC6-*w*/*r*/*b*:

- *w*: the word size in bits (specified as 32 for the AES);
- *r*: the number of rounds (20 for the AES; note that an RC6 round is equivalent to one DES round again, as the RC5 redefinition caused confusion);
- *b*: the keylength in bytes (for the AES one of 16, 24 or 32, although 0 to 255 bytes are legal).

RC6 actually encrypts slightly faster than RC5 in software [RRS]; although the operations are slightly more complex, the increased block size allows a faster throughput. The key setup is practically identical in both ciphers. In general, for the purposes of this paper, we will assume that breaking RC5 and RC6 by exhaustive keysearch are equally difficult, and that the time taken to examine the key space (when specified for the same key length and number of rounds) will be approximately the same.

## *3. Overall Machine Specifications*

The machine can now be specified in a general sense. As in the EFF DES cracker, the core of the machine is the search unit. This unit would be targeted against the RC6 chip.

We will then state that each dedicated chip will have 64 search units and each board will have 64 chips (to reflect the higher densities that are now achievable). Boards will then be rack-mounted with their supporting electronics. Rather than going for parallel connections to PCs, each board will have its own Ethernet network interface card (NIC) built in and be connected to a LAN. In addition, the boards will have a special controller unit.

A group of boards will effectively have their own 10/100 Mbps LAN, with a host PC managing it. In turn, the PCs will connect back via a logically separate LAN to an overall controlling PC.

## *4. The Search Unit*

As the search unit is targeted against RC6, it would have the following:

- Two 128-byte registers, one for the plaintext and one for the cyphertext.
- A 32-byte key register, with support for variable key lengths from 0 to 32 bytes. This register will be similar to the DES cracker for speed, in that the bottom 32 bits are incremented automatically by the search unit, and then a controlling device loads a new key.

- A lookup table to support 'partial text matching', to allow searches where the plaintext is not known but its characteristics are (this is discussed further below).
- RAM for the key table, working space and buffers (only a few hundred bytes).
- The four 32-bit working registers.

It is likely that for very fast decryption both the key schedule and the main encryption loops would be 'unwound' to some extent in hardware. The search unit will only do one decryption on a 128-bit block to decide if it is a candidate, so the key schedule now carries a significant overhead on the search unit operations and must be optimised.

In addition, the matching operations would be optimized on the chip, so that while the decryption run on the current key is in progress the results of the previous key decryption are being matched. After matching, the key schedule processing would commence for the next key whilst the current key decryption is still running.

We will make the assumption that the NSA will spend a considerable amount of time designing the search units, chips and boards. With this assumption, we will define that the unit will have the following capabilities:

1) Due to Moore's law, the chip speeds will be at least 100 MHz 'for free' (the contract to manufacture the EFF chips was signed September 1997).
2) Decrypt/test matching will be done four bytes at a time, and can happen during the first four rounds of the decryption.
3) Key increment and schedule processing are done at the last rounds of the decryption, with no delay into the start of the next key's run.
4) That the decryption loops can be 'unwound' at least so that it can do two RC6 rounds (equivalent to one RC5 round) at a time: thus the search unit does a decryption in 10 rounds, at one clock cycle per round. This means that the unit can search 10,000,000 RC6 keys per second, and thus will step through its key-counter in $2^{32}/10^7$ = 429.5 seconds, or 7.16 minutes.

Because of the considerable effort in up-front design, we will take it that the overall cost per search unit will be the same for the RC6 cracker as for the EFF machine, at least in small numbers (production runs will be taken into account later). The increase in complexity is countermatched by the greater effort in design. Note that the chips will be more expensive as they will have more search units.

## *5. False Positive Rate*

The expected rate of false positives has a major effect on the overall system design, as it specifies the frequency and bandwidth of communications between the search units and the external monitoring systems. This puts a limit on what can be done within certain time periods.

We have assumed that for speed a similar mechanism to that specified for the EFF machine is used: that there is one 'Plaintext Vector' of 256 bits, that specifies allowed plaintext byte values

irrespective of their position. This system is very flexible, but can generate quite a large number of false positives.

Consider a message that is to be tested; it is known to start with '#0a394bdf LINK-CIRCUIT … '. The first sixteen characters are all different values (RC6 handles 16 bytes at a time). Thus the Plaintext Vector will be set up with 16 allowable bytes. The probability that by chance the first byte in the message decodes to one of the allowable values is 16/256 = 1/16. However, the probability that all 16 bytes, by chance, decrypt to an allowable value is $(1/16)^{16} = 1/2^{64}$. If the key length being searched for is 64 bits, and thus one is searching for $2^{64}$ keys, then the probability is about 1 false positive in a complete search of the full key space: as often as finding the key.

However, a working machine may have to deal with the case where all that is known is that the text is ASCII. Depending on assumptions about the usage of punctuation, about 64 characters might be used in the message. This will lead to 64 acceptable values in the vector, which means that the probability of one byte matching is 1/4, and 16 bytes matching by chance as $(1/4)^{16} = 1/2^{32}$. If the key is 64 bits long, this means that there will be $2^{32}$ false positives in a complete run. The false positive number will increase with key length (reaching $2^{96}$ for a 128-bit key).

A complete 64-bit keybreak may be seen as $2^{32}$ search unit runs, each one of which steps through $2^{32}$ keys in 7.16 minutes (as specified above). This implies that each search unit is likely to find a false positive in each of its runs. This has implications, as will be discussed below.

## *6. The Chip*

The chip will have 64 search units. It will be more complex than the EFF equivalent, but most of the increased complexity has already been defined in the search unit. The other increase in complexity is due to the fact that in each 32-bit key run in a 'known to be ASCII' test a search unit may well find a false positive (but still a candidate solution). Clearly, the unit cannot stop and wait to be interrogated.

The chip needs to be able to put the search unit ID, the chip ID and the candidate key into registers that can be read by the board controller. It then allows the unit to carry on. The read/write process might delay the unit for a couple of cycles, but this is trivial over a search run.

This will increase the cost of the chip slightly for the increased logic above and beyond the EFF chip, and for the extra registers.

## *7. The Board*

The boards have 64 chips, but their physical size would depend on the actual fabrication method of the chips. Thus they may or may not be larger than the EFF ones. The boards would service all the chips, but depart from the EFF ones in two basic ways:

1) They have a 10mbps Ethernet NIC, with RJ45 socket or equivalent suitable for a Cat-5 network infrastructure. IP is assumed to be the protocol. Addresses would be set by dip switches, as dynamic addressing is not required for an isolated machine.

2) There is an on-board controller. This device carries out the following tasks:
   - Loads the chips and units with their address spaces, under control from the host PC.
   - Monitors the chips for their candidate keys (the chip will actually signal the on-board controller). The controller will then read the key, chip and search unit IDs, and send them to the host PC via the Ethernet NIC, along with its own ID.
   - Alert the host PC about any fatal errors.

It is proposed that there will be $2^{20}$ = 1,048,576 boards in the machine, racked into cabinets. This means that there are $2^{20}$ * 64 * 64 = $2^{32}$ search units. We can see that for a 64-bit key search, every search unit will do one run. Thus the maximum length of a search will be 7.16 minutes; on average it will be 3.6 minutes for a 64-bit key.

## *8. Host Controllers and Network*

Each board will be connected onto an Ethernet with Cat-5 cabling and switches. There will be 512 PCs, thus each PC will manage 2048 boards. Although the physical cabling and how it will be switched will depend on an optimization study, for the purposes of this paper we can regard each PC as being on a private network to 2048 boards. Each board will connect to a hub at 10mbps, and the switches will feed back to the PC at 100mbps (the backbone). A number of hubs will connect to a switch.

The PCs will in turn be linked to an overall control PC, which starts the process off and gets the final results. There will be a logical host PC network, which has a much lower throughput then the PC to search boards one.

## *9. The Overall Machine*

There are 1,048,576 boards. If we assume high chassis we can envisage 64 boards to a chassis: we are not assuming the 9U VMEbus specified by EFF. This gives 16,384 chassis. If each chassis sits on a square metre, and requires a square metre access space, on average 1.5 square metres will do per chassis (as two facing chassis can ‘share’ the access space, and they would be in back-to-back rows). This leads to a requirement of 24,576 square metres, and will take up about 21% of the Pentagon (which is quoted at 117,400 square metres).

However, because the machine is so modularized, each PC with its 2048 boards could be physically quite separate, especially if the host PC network backbone were upgraded to fibre (at costs which would be minor compared to the overall cost of the machine, see below). Thus the machine could be distributed over a university campus or even a city, making its footprint more manageable.

## *10. Communications*

The greatest potential bottleneck in the machine is related to the communications. This is under the situation where on a 64-bit key run the machine may generate $2^{32}$ false positives as candidate results. Each PC will see 1/512 of the total false positives in a full run, or $2^{23}$ positives. There are two potential bottlenecks: network throughput and PC processing.

### 10.1 Network Throughput

For each candidate result, the following information would be sent:

- Search unit ID (1 byte).
- Chip ID (1 byte).
- Board ID (2 bytes).
- Candidate key (up to 32 bytes).

Given Ethernet frame (26 bytes) and IP headers (24 bytes), we could say that the whole package would be about 100 bytes to report back a key. There are 2048 ($2^{11}$) managed by each PC, thus each board will generate $2^{12}$ false positive candidates (64 chips with 64 search units = $2^{12}$). Thus each board will generate $2^{12}$ * 100 = 409,600 bytes of candidate key reports in one run.

A run is 7.16 minutes, so a board will generate 953.4 bytes per second of communications, or 7,628bps. This is well within the capability of the 10Mbps link back to the hub.

There are 2048 boards linked to the PC, so the PC will see 7,628 * 2048 = 15.62Mbps on the 100Mbps links: 16% utilization should not be a problem, and be well within the switches' capacity. Naturally, the PC will need a high-performance network card to handle input. Note that the actual amount of data is only about a third of the network throughput.

### 10.2 PC Throughput

Each PC will have to analyse $2^{23}$ false positives. It will do this by decrypting the second, third and fourth 16-byte blocks in software with the candidate key (the first has already passed). Even under 'known to be ASCII' conditions the chances of this matching by luck is $(1/4)^{48}$, or one in $2^{96}$. Thus this test will efficiently weed out false positives for up to a 128-bit key (more decipherments would have to be done for longer keys).

The PC must analyse $2^{23}$ keys in the 7.16 minute run, or 19,927 keys per second. Currently, the Distributed.net client running the RC5 64-bit key breaking trial does 748,982.85 keys per second on a Pentium-II 300MHz processor running the Windows client. Given that the host PC would have to do three blocks, where the Distributed.net RC5 client may well only do one, at a first estimate we can see that the PCs would have the capacity to do some 249,660 keys per second (assuming that RC6 can be run as fast in software as RC5).

The conclusions are that the throughput should not be limiting. The components should be able to search a 64-bit keyspace in the given time.

## *11. Machine Functioning*

The machine would function as follows:

1) The overall control PC would be fed the plaintext and cyphertext, and the key size.
2) The overall control PC would distribute the information to the host PCs.

3) The host PCs would download to each board's controller the ciphertext and plaintext blocks, and the starting address of its key space. There are $2^{20}$ boards, so each board gets $2^{44}$ keys in its keyspace for a 64-bit key break.
4) The board controller then distributes the keys to the chips and units. It is now only dealing with the upper 32 bits of the key (as each search unit covers the full lower 32 bits of a given key). Thus each chip gets $2^{38}$ keys and each search unit $2^{32}$ keys. Thus the search unit has the top 32 bits 'frozen' while it checks all $2^{32}$ keys in its lowest key register.
5) Candidates are identified and fed back to the host PCs for checking. The final candidate is fed back to the control PC for full decryption.
6) At the end of the search units' run after 7.16 minutes, the board controller relays the finish event back to the host PC (with a special code rather than a key value). If there is more keyspace to search (say for 72 bit key), then the board controller is given a new part of the key space.
7) Lastly, there will be a 'shutdown' message to all board controllers when the key is identified, probably followed by a new plaintext, cyphertext and key for another break.

## *12. Costs*

There are several contributing factors to the cost.

### 12.1 Parallel Component Costs

Identifying the costs for the main bulk of the machine needs some estimations. These may be described as so:

1) The search unit, although more complex than the EFF one, is held at the same $3.53 per unit. This is because the design will ensure that the costs are held constant. It should be restated that this cost includes the unit's share of construction, board manufacture and assembly, power unit, cabinet and so forth.
2) The chip has 64 search units, so its cost = 64 * $3.53 = $225.92. We have also specified a small amount of logic on the chip (the return of the candidate key whilst keeping a unit running), so the cost will be raised by 10%, and thus equals $248.51. Again, this includes the chip's share of all other costs such as manufacturing and assembly, and all support components.
3) The board has 64 chips, so its cost is 64 * $248.51 = $15,904.77. However, we have uprated the board with a network card and a board controller. We will price a NIC at $20 on-board (this is straight from a catalogue for a 6-pack D-Link card, 32-bit PCI 10Mbpcs), and the cost for the extra logic as $100 (equivalent to a Cyrix MII P300MMX). Both of these should be overestimates, and make the board cost $16,025 (rounding up). This cost includes the power units, assembly, cabinets and so forth.
4) This board cost reflects EFF's price, which was for 24 boards containing 1,536 chips and 98,304 search units. We are devising a machine of 1,048,576 boards containing 67,108,864 chips and 4,294,967,296 search units. The costs will be entirely different for such mass production. We have implemented a model that we believe to be conservative: we have taken the above price for the first 1,000 boards. We have then dropped the board cost to 1/5

the previous cost for each order of magnitude increase (this estimate is a result of discussions). The profile looks as follows:

- The first 1,000 boards cost $16,025 each = $16,025,000
- The next 10,000 boards cost $3,205 each = $32,050,000
- The next 100,000 boards cost $641 each = $64,100,000
- The next 1,000,000 boards cost $128.2 each = $128,200,000

We thus get a total cost of $240,375,000 for the boards in their chassis, plus power and all fixings. This production run will give us 1,111,000 boards, an excess of 62,424 (approximately 6% spares).

**12.2 PC Costs**

Although EFF's board costs included the PC costs, we have decided to calculate them separately. This means that there is, in fact, extra margin of safety in the board cost estimates. The PCs need to be at least 300MHz Pentium II equivalents, with a high-performance 100Mbps network board. There are no special disk space or monitor requirements, nor for multimedia accessories. We have allocated a generous $1,000 per PC, thus making a total PC cost of $513,000 (512 hosts plus 1 overall controller). This includes all hardware and an operating system thrown in.

**12.3 Networking Costs**

A modern network based on switched Ethernet has most of the cost associated with the switches and the hubs. A review of pricing and performance indicates that we do not need special high-throughput switching for the design we have gone for, so a cost of $20 per port seems reasonable (estimate from catalogue prices). This is probably an over-estimate (especially for 1,048,576 + 513 ports, but again leaves room for error). This gives a total electronics cost of $20,981,780.

If we assume that the hub and switch cabinets are distributed amongst their client boards, then we may estimate that one switch/hub cabinet with 2048 ports will be surrounded by some 32 board cabinets with 2048 boards. In a typical configuration, this would be a space 8 metres by 5 metres, with the comms rack in the middle. Assuming that cables have to 'dog-leg' and that there will be twists and turns to get into and out of cabinets (and go via floor and ceiling), the average cable run might be 8 metres.

So 2048 boards will require 8 * 2048 = 16,384 metres of Cat-5 cabling. This is about $100 for 300 metres (catalogue price), which gives us a cost of $5,461 for the board network cabling. We can raise this to $5,500 for the cabling back to the host PC. Thus for 512 PCs we reach a cable cost of $2,816,000. We will assume that the cable costs for the host PC network can be subsumed within this cost.

We need two RJ45 (or better) adapters per port. We have estimated at $1 per pair, giving a cost of $1,049,089 for the machine.

We thus end up with a total cost of $24,846,869 for the network.

### 12.4 Design Costs

The EFF machine cost $80,000 to design. We have taken this base figure, but note that we have made the following important modifications:

1) Redesigned the search unit for RC6.
2) Unwound the key schedule part of the search unit.
3) Unwound the main decryption unit to do two rounds at a time.
4) Increased the density of the search units and chips.
5) Added a small bit of control logic to the chip.
6) Added the network card to the board.
7) Added the logic circuit to the board.

We have thus made the generous assumption that each change doubles the design work (we are not expecting efficiencies). Thus $2^7$ * $80,000 gives us a bill of $10,400,000 for design. It should be noted that at this level of design the chip may be much more efficient than we have specified.

### 12.5 Software Development Costs

Given that there is a model to follow in the EFF machine, we believe that the software development costs should not be excessive. As it is in assembler, we will assume 5 man-years at $100,000 per year, giving $500,000. Again, this is probably a generous overestimate.

### 12.6 Total Costs

Total costs are as follows:

| | |
|---|---|
| Main cracking unit | $240,375,000 |
| PCs | $513,000 |
| Network | $24,846,869 |
| Design | $10,400,000 |
| Software development | $500,000 |
| **Total** | **$276,634,869** |

Or to all intents and purposes $280 million.

## *13. Machine Capabilities*

We have seen that the machine can do a full exhaustive key search of a 64-bit key in 7.16 minutes. On average, only 50% of the keyspace needs to be searched, so the average keybreak will be in 3.58 minutes.

A 72-bit key will take 256 times the effort (it can be regarded as running the 64-bit key run 256 times). The machine is designed to take up to 256-bit keys, so the increase in key length can be directly handled. As each 64-bit key search ends the boards are assigned new parts of the key space. Thus a 72-bit key will take 3.58 * 256 = 916.48 minutes, or 15.27 hours.

An 80-bit key would take 15.27 * 256 = 163 days, and an 88-bit key some 144.3 years, which seems a bit long.

Thus this paper implies that the NSA can if they wish break 64 to 80-bit keys routinely at the investment cost of $280 million.

### *14. NSA Budget*

It may be suggested that $280 million would not be credible. However, the NSA budget as of 1998/1999 is estimated at slightly under $4 billion, of which about $900 million is payroll and $2 billion operations [FAS]. This leaves some 1 billion for other purposes, of which around $250 million would appear to be procurement. Thus the proposed machine would be within 1 years' NSA purchasing budget, and the actual cost would probably be spread over at least two or three years. Given that the project would also be experimental, there would probably be other budgets (such as direct military and R&D) that could help contribute.

Given the sort of money the US Government is willing to invest in projects (such as $8 billion so far for the new Space Lab), a budget of $280 million does not seem incredible. Whether the NSA really wants to build such a machine is another issue (see below).

### *15. Validation*

There are a number of weak points in the machine specification. They are covered in the next section. However, there is evidence that indicates the estimates in this document are not unreasonable.

Matt Blaze *et al* in 1996 [BDR] estimated a $300 million machine could crack DES in 12 seconds. If we imagine an equivalent machine being capable of dealing with RC6 and longer keys, then a 64-bit key would take 12 * 256 = 51.2 minutes. If we apply Moore's law, their machine would now do it in a quarter of the time (doubling raw capability every 18 months), which would imply that their machine could do it in 12.8 minutes compared to the 3.58 minutes for the machine proposed in this paper. The two estimates are extraordinarily close given that they appear to have been calculated from completely different premises.

## Future Work

Whilst it is unlikely that the machine described in this paper would ever be built, there are a number of areas where the estimates and design could be improved:

1) The estimate is mostly dependent on the cost curve for the boards given in 12.1. Small changes in these costs could radically change the price of the machine, so this estimate needs to be examined further.
2) All the other costs have tended to be 'worst case'. Effort should be put in to better estimate costs when ordering such large quantities of items.
3) The core of the machine is the search unit. Designing an effective one should be undertaken; in addition, a study should be made on how small the unit can be made, and then how many can be fitted into a chip.

4) Similarly, the overall chip design should be studied further, and how many can be fitted onto a board.
5) The current machine form factor reflects the EFF design, which should be reviewed. There may be better solutions.
6) Points (3) and (4) above will have a major effect on reducing its physical size. Other options, such as double-sided boards, should be investigated in trying to get the board count down.
7) The actual construction technology should be investigated and costed. Selection of the right technology might produce significant savings (ie avoiding 'bleeding edge' technology may improve the price/performance ratio and stability of the system).

## Cautions

The machine specified in this document would never be built, mostly because the NSA is not going to spend $300 million to break a few keys. This is because there are more economical ways of attacking people's cryptography, such as:

1) It is easy to subvert a person in the target organisation, who will give you the information.
2) It is most likely that the crypto system will be used in an incompetent way, allowing a break on a general-purpose machine. It is also not unknown for people to be sending in clear when they believe the message to be encrypted.
3) Often, the actual implementation of the system is flawed in some way by the manufacturer, allowing relatively simple breaks.
4) If the system is in software, it is relatively easy (again by subversion) to undermine it with a replacement system (so it appears to function, but at your bidding).

Lastly, such a machine is of no use unless it is targeted against the 'standard cipher' and it can break keys in seconds (so allowing for 'trawling' over bulk messages from target locations). Thus whilst it is possible that the NSA have a DES cracker, they will not construct any replacement until the candidate AES is chosen, and if such a machine becomes feasible against the AES cipher.

## References


[BDR] M Blaze, W Diffie, R L Rivest, B Schneier, T Shimomura, E Thompson and M Wiener. *Minimal Key Lengths For Symmetric Ciphers To Provide Adequate Commercial Security*, www.bsa.org/policy/encryption/cryptographers_c.html, January 1996.

[DIN] www.distributed.net.

[EFF] *Cracking DES. Secrets of Encryption Research, Wiretap Politics & Chip Design.* Electronic Frontier Foundation, May 1998, ISBN 1-56592-520-3.

[FAS] *National Security Agency. Budget and Personnel.* Maintained by John Pike, www.fas.org/irp/nsa/nsabudget.html.

[RLR] R L Rivest. The RC5 Encryption Algorithm. *Proceedings of the 2nd workshop on Fast Software Encryption*, Springer, 1995.

[RRS] R L Rivest, M J B Robshaw, R Sidney and Y L Yin. *The RC6™ Block Cipher.* www.rsa.com/rsalabs/aes/rc6v11.pdf, 1998.