The Case For Serpent

Ross Anderson
Eli Biham
Lars Knudsen

AES Selection Criteria

- NIST's main criterion :

  "The security provided by an algorithm is the most important factor in the evaluation"

- In detail :

  - the most secure algorithm which is also acceptably fast (i.e., faster than 3DES) on a wide range of platforms

  NOT

  - the fastest algorithm on a particular processor against which no-one has a production attack yet

- We engineered Serpent exactly to the specification

Protection Requirements

- NIST: "An algorithm for the twenty-first century"
- Basic minimum: a useful service lifetime, plus a human lifetime after that – so at least 100 years
- May need more than 100 years:
  - end of Moore's law
  - more long-lived sensitive data, e.g. genomic
  - persistence of standards – such as speed limits example
- So what does it mean for a cipher to be secure for over a century?

Cryptosecurity

- No-one should discover an exploitable shortcut attack, even after a century (and more) of progress in mathematics, physics, ...

- The best solution we can offer is:
  - a simple, easy to analyse cipher
  - uses well-understood primitives and builds on the huge amount of work done on DES
  - has many more rounds than are needed today, to give a large margin of safety

- These were the principles behind the design of Serpent

Assurance

- Algorithm assurance - did we implement it right? Easiest to check with simple operations, structure
- System assurance - did we use it right? This is where most <u>real</u> failures happen:
  - bad random number generators
  - memory remanence
  - ...
- Having keys that are 'only just long enough' is fatal in the presence of many of these common flaws
- So 256 bit keys are important - and must be the default - regardless of Moore's Law

Confidence

- Recall what happened with DES:
  - was there a trapdoor?
  - was the keylength 'deliberate' and if so for what purpose?
  - was DES 'broken' by differential cryptanalysis?
- Many first-round AES candidates were rejected outright by the community because of 'certificational' attacks
- The risk of using 16R Serpent (or any of the other finalists) is a certificational attack at 'Crypto 2050' - what's the economic cost of this?
- AES must also withstand certificational attacks - not just production attacks - for at least 100 years!

Speed

- The 'conventional wisdom' on Serpent is that it is the most secure candidate by far, but is half as fast as the others because we used twice the number of rounds we needed to
- BUT:
  - Serpent is best in hardware
  - Serpent is excellent in smartcards
- So the critical question is: was Serpent's security bought at an unacceptable price in software speed on larger processors?
- What's the best benchmark?

The Best Benchmark

- Some candidates are heavily optimised for throughput on Pentium. But apps such as encrypting file systems aren't the most common, or the most demanding
  - IPSEC : 4 blocks / key
  - ATM : 3 blocks / key
  - protocols : 1 block / key
- Natural benchmark : ATM. Average of 3 encryptions + decryptions including key setup
- With this benchmark, Worley et al gives :

	MARS	RC6	Rijndael	Serpent	Twofish
IA-64	2965	3051	504	2269	2991
PA-RISC	3409	2686	666	2415	3453

Serpent is second fastest after Rijndael !

- On Pentium, with Osvik's s-boxes, Serpent is 3rd or 4th depending on processor

Conclusions (1)

- Serpent's security was not bought at an unacceptable cost in speed
  - it's best on hardware
  - it's second on smartcards
  - it's second on IA64/PA-RISC in the most likely critical apps of the 21st century
  - even encrypting large files on the Pentium, it more than meets the specification of 'faster than 3DES'
- We propose 32-round 256-bit Serpent as the Advanced Encryption standard
- If 128 and 192 bits need explicit support, the standard should still recommend 256
- If 2 algorithms win, Serpent should be the primary algorithm and the weaker one the alternate

Conclusion (2)

Serpent should be chosen as the Advanced Encryption Standard because it is the most secure of the candidates