

AES: The Selection Process

Vincent Rijmen

Cryptomathic, Belgium

Graz University of Technology, Austria

Outline

1. AES: scope and significance
2. AES: the Process
3. The 5 finalists

Who is behind AES ?

- AES: Advanced Encryption Standard
- Initiative of the US National Institute of Standards and Technology (NIST)
known from DES, DES modes of use, SHA-1, DSA, ...
- Goal: to find a successor for DES
- <http://www.nist.gov/aes>

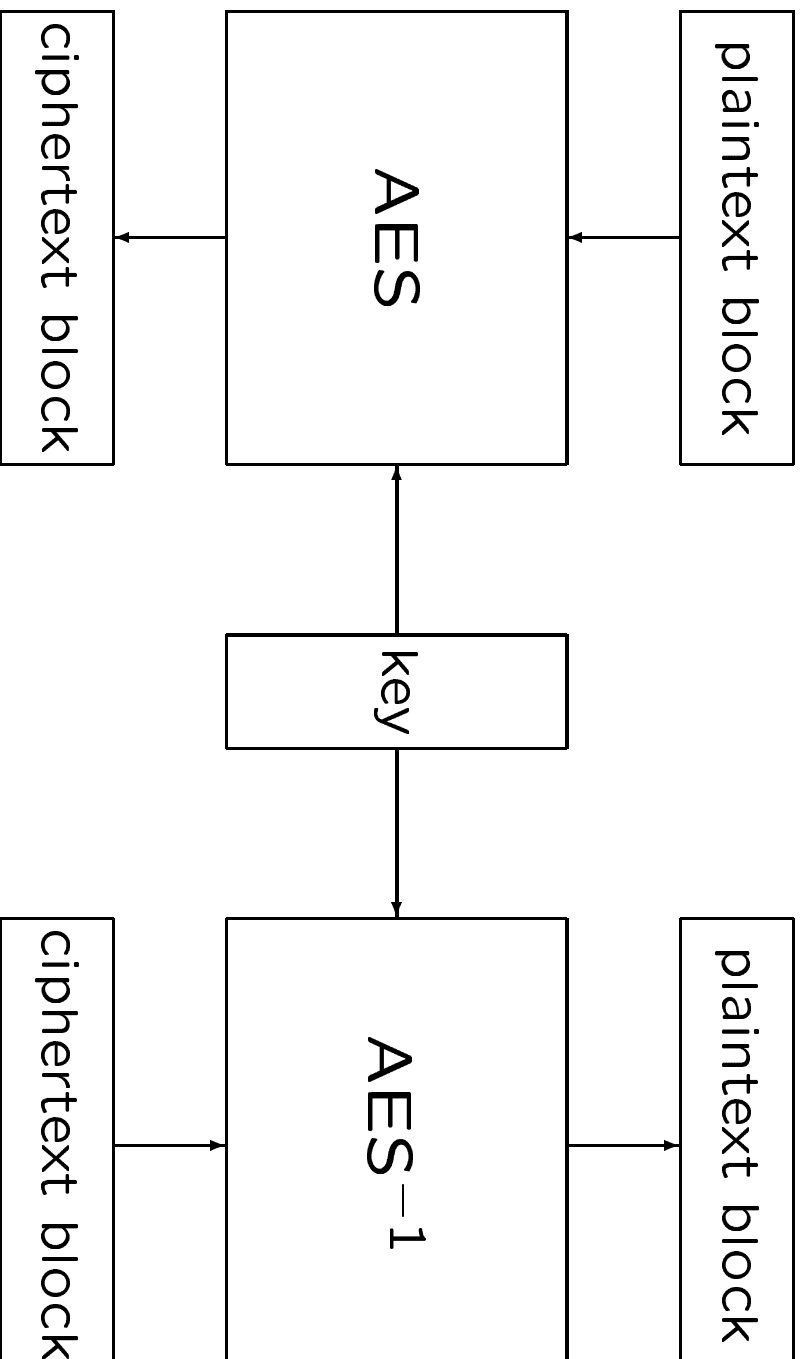
AES: Scope and significance

- Standard signed December 2001, in action May 2002
- Official scope is limited:
 - US Federal Administration
 - Documents that are ‘sensitive but not classified’
- Significance is huge: AES is the successor of DES
- Major factors for quick acceptance:
 - No royalties
 - Low resource consumption
 - High quality

Some AES/Rijndael Commitments

- Standardisation: ISO, IPsec, ETSI, (NESSIE)
- Companies: RSA labs, Entrust, Pijnenburg
- Banking
- Consumer products

AES: what is it ?



AES: what is it ?

- Block cipher: plaintext *blocks* → ciphertext *blocks*
Symmetric encryption algorithm
Can be used to build applications
- Dimensions:
 - Block length: 128 bits
 - Key length: 128, 192, 256 bits
- DES: 64-bit blocks, 56-bit keys
- 3-DES: 112-bit keys

The start of the AES process

Call for Algorithms: January 1997

- Block cipher: 128-bit blocks, 128/192/256-bit keys
- Strength \approx 3-DES, efficiency ↗ ↘
- Documentation, reference C code, optimised C and JAVA code, test vectors
- Algorithm available royalty-free world-wide

Open process: public comments, international submissions

First AES conference: August 1998

The 15 accepted submissions

CAST-256	Entrust (CA)
Crypton	Future Systems (KR)
E2	NTT (JP)
Frog	TecApro (CR)
Magenta	Deutsche Telekom (DE)
Mars	IBM (USA)
RC6	RSA (USA)
SAFER+	Cylink (USA)
Twofish	Counterpane (USA)
DEAL	Outerbridge, Knudsen (USA–DK)
DFC	ENS-CNRS (FR)
HPC	Schroeppeel (USA)
LOKI97	Brown et al. (AU)
Rijndael	Daemen and Rijmen (BE)
Serpent	Anderson, Biham, Knudsen (UK–IL–DK)

Evaluation: Round 1

NIST effort: portability of code
statistical tests
compiling public inputs

All other inputs: public research community

Evaluation: Round 1

Criteria: Cryptographic strength

Efficiency (platforms)

IP issues

Flexibility

Elegance, absence of trapdoors

Problems: Comparing cryptographic strength (security margin)

Export restrictions (even for submitters)

Nonuniform performance (platforms, compilers)

IP subtleties

Definition of 'elegance'

Second AES conference (Rome '99)

Key phrases:

Designed to resist unknown attacks

Provably absolutely secure ...

... against *some* cryptanalytic attacks

Will profit considerably from Future Hardware

Selection of the Finalists

Security problems: DEAL, Frog, HPC, LOKI97, Magenta

Very slow: DEAL, Frog, Magenta, SAFER+, (Serpent)

'Copy': Crypton

'Losers': CAST-256, DFC, E2

No problems, but not good enough

5 finalists

Mars (IBM)

RC6 (RSA)

Rijndael (Daemen, Rijmen)

Serpent (Anderson, Biham, Knudsen)

Twofish (Counterpane & friends)

Evaluation: second round

Performance in hardware

IP-issues

Cross-cutting analysis (impartial)

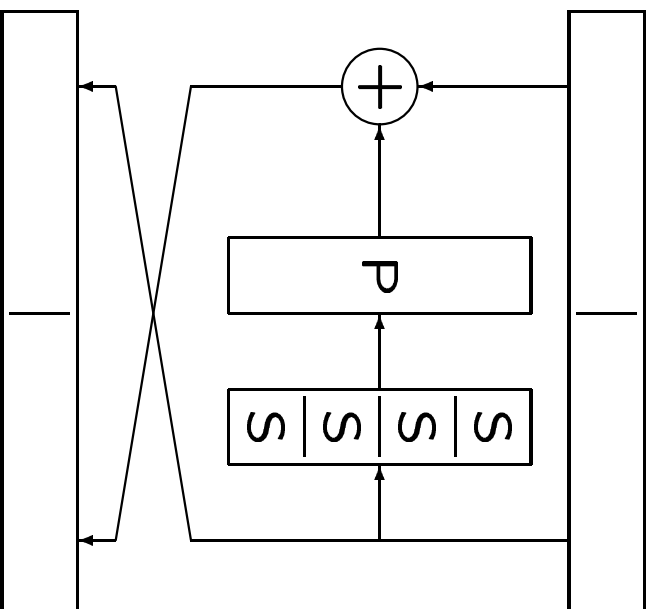
NSA input: hardware implementation

comments on NIST's selection

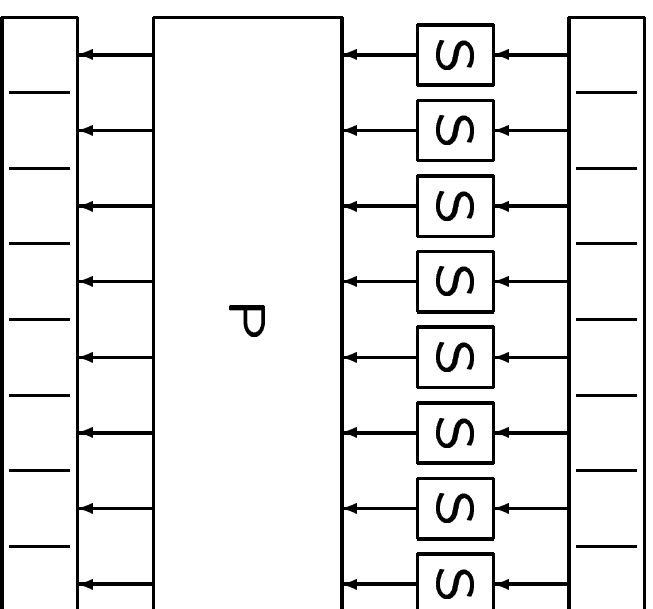
Design principles: common aspects

- Iterated ciphers
repeat a (simple) transformation, or *round*
Small code size, hardware area
- Alternate highly nonlinear elements and
mixing transformations
- Resistant against known attacks

Feistel Ciphers and S-P-Networks



Feistel Cipher



S/P-Network

Design Tradeoff

L. O'Connor:

“Most ciphers are secure after sufficiently many rounds.”

J.L. Massey:

“Most ciphers are too slow after sufficiently many rounds.”

Goal: sufficient security for the smallest ‘cost’
(performance drop, chip area, ...)

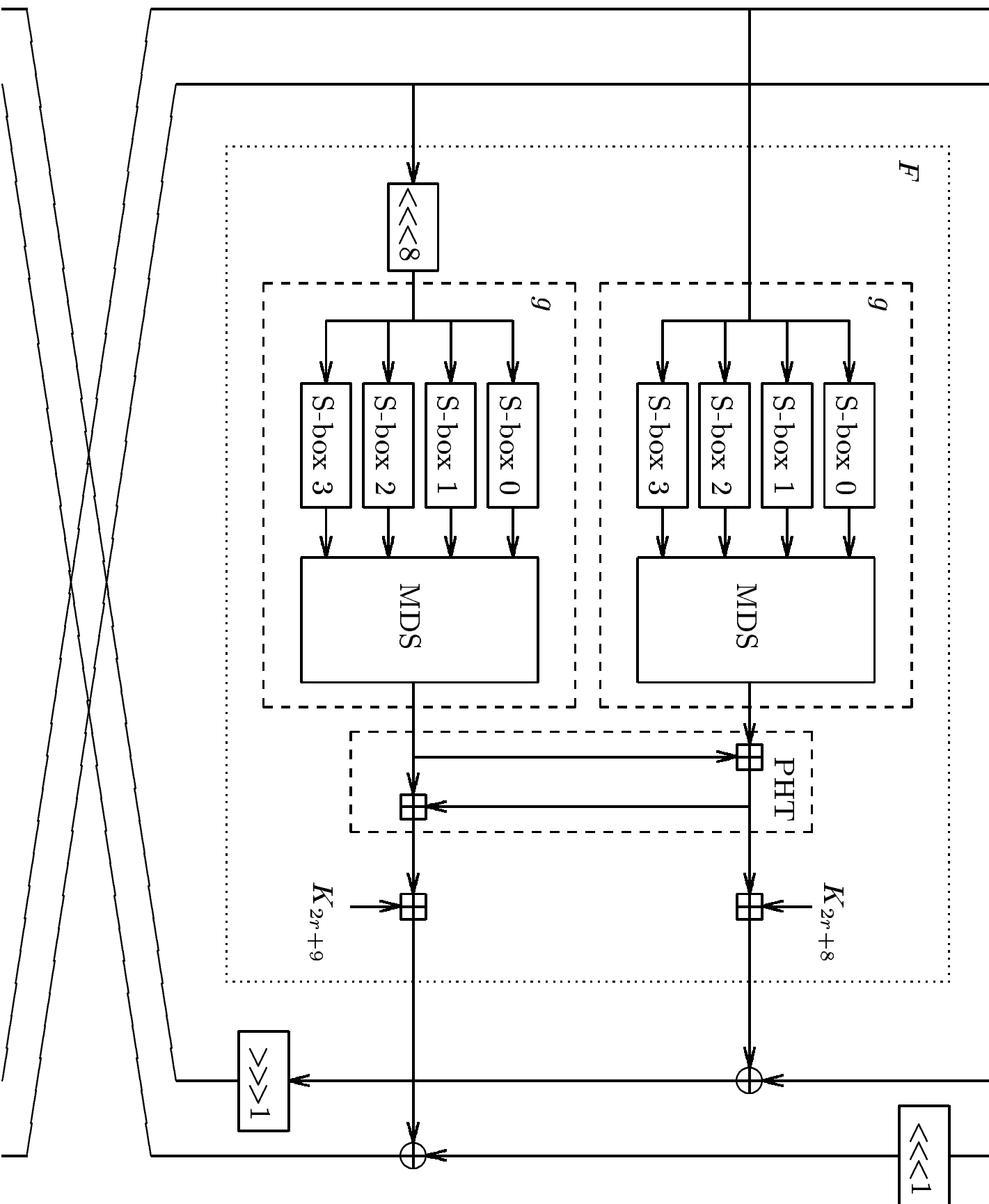
Twofish

Based on the Feistel structure

Combines elements from other ciphers (Square, Safer, Khufu)
Other operations added
“whenever they are cheaply available”

Uses XOR, table lookup, bit rotation, 32-bit addition
Many small details

Complex, but fast on many platforms



Rc6

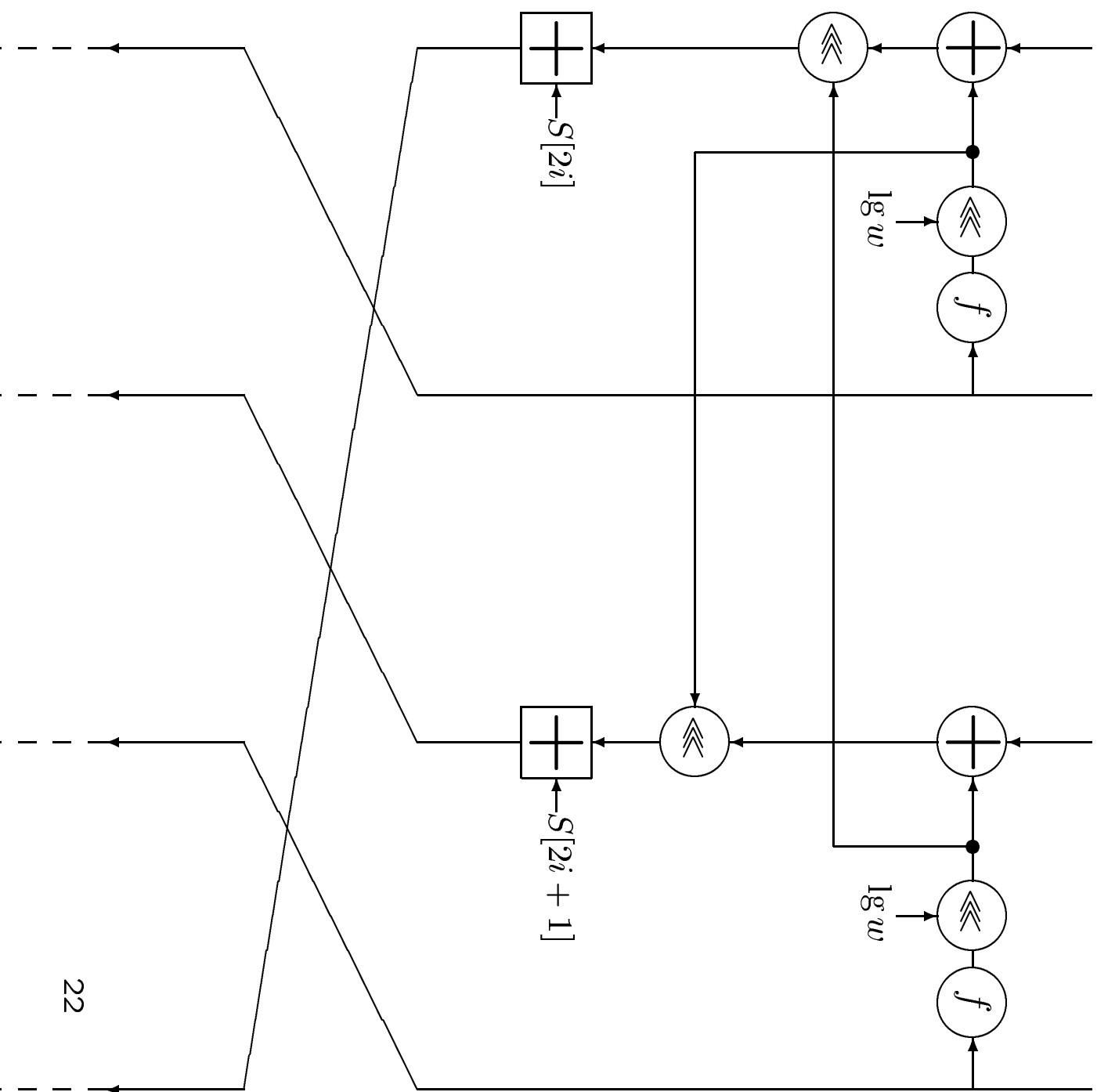
Input: $A, B, C, D, K[44]$

Output: A, B, C, D

```
B = B + K[0]
D = D + K[1]
```

```
for i = 1 to 20 do
{
  t = (B * (2B + 1)) <<< 5
  u = (D * (2D + 1)) <<< 5
  A = ((A ^ t) <<< u) + K[2i]
  C = ((C ^ u) <<< t) + K[2i+1]
  (A,B,C,D) = (B,C,D,A)
}
```

```
A = A + K[42]
C = C + K[43]
```



RC6

Uses the Feistel structure

Uses no table lookups

Very small code footprint (on 32-bit platforms)

Uses bit rotations, XOR, 32-bit addition and squaring

Fastest candidate for bulk encryption on 'PC processors'

Impossible on smartcards (RAM) (*)

(*): NIST's original criteria did not emphasize smartcard applications

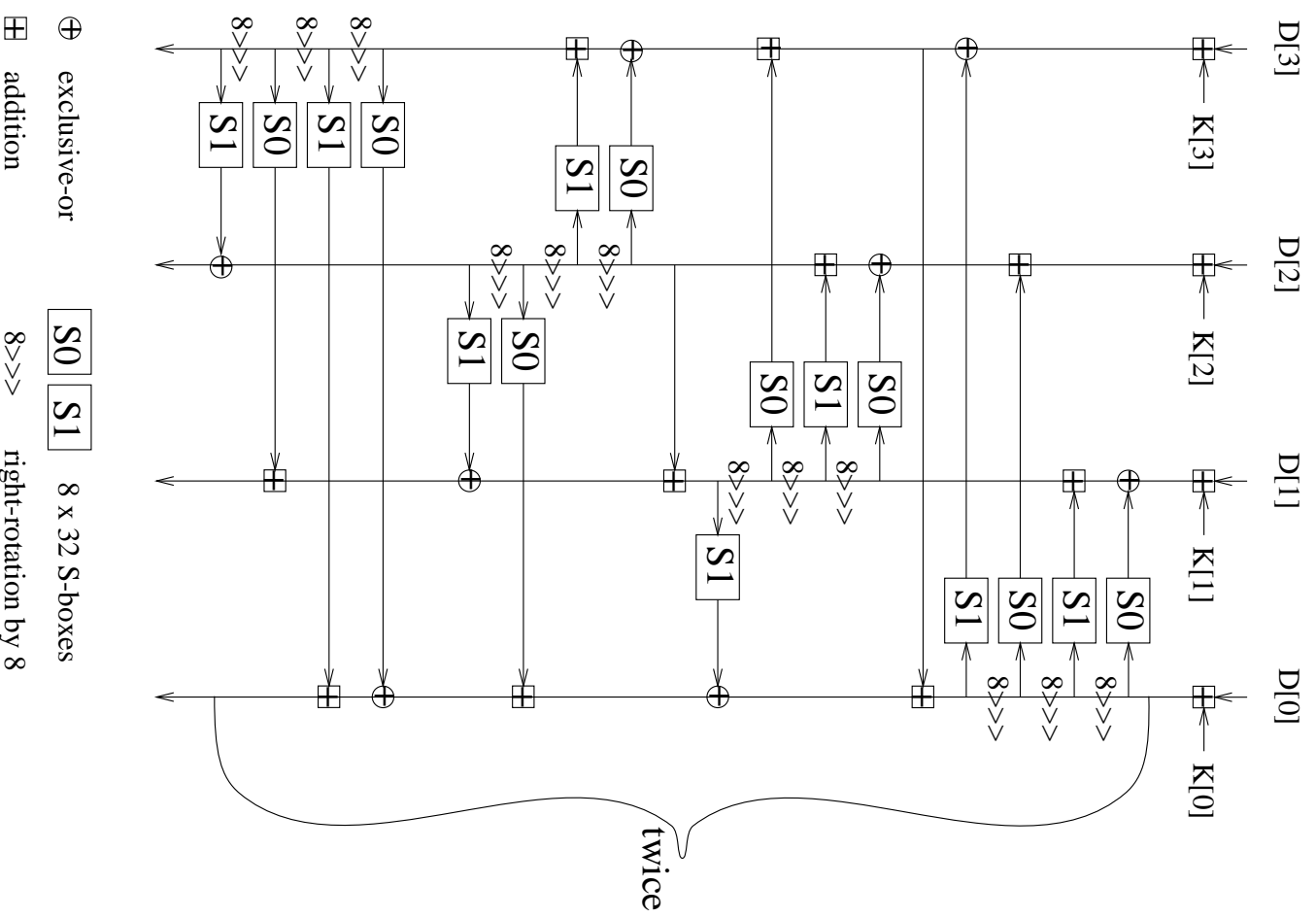


Figure 2: Structure of the forward mixing phase

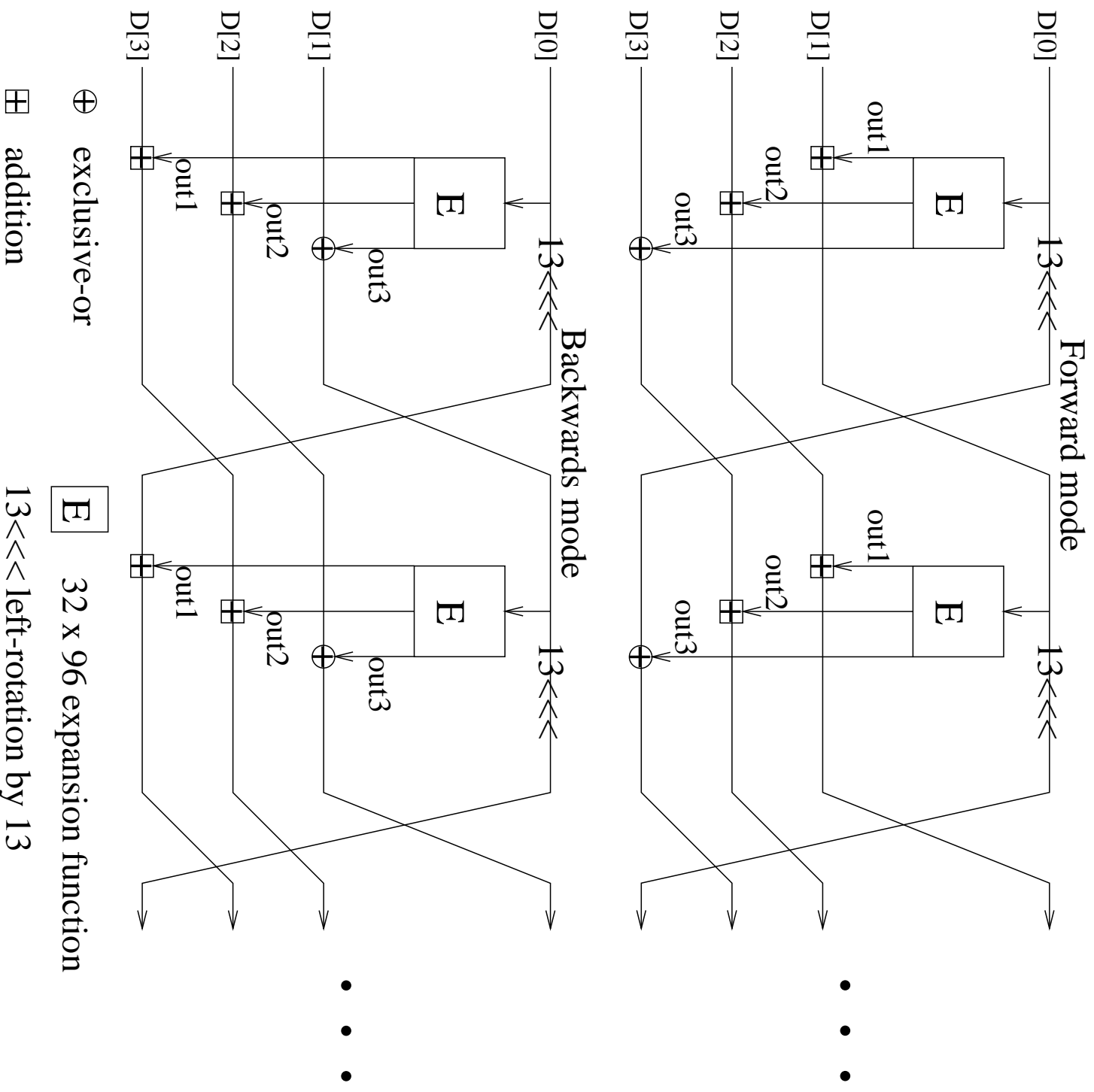


Figure 3: The type-3 Feistel network of the main keyed transformation.

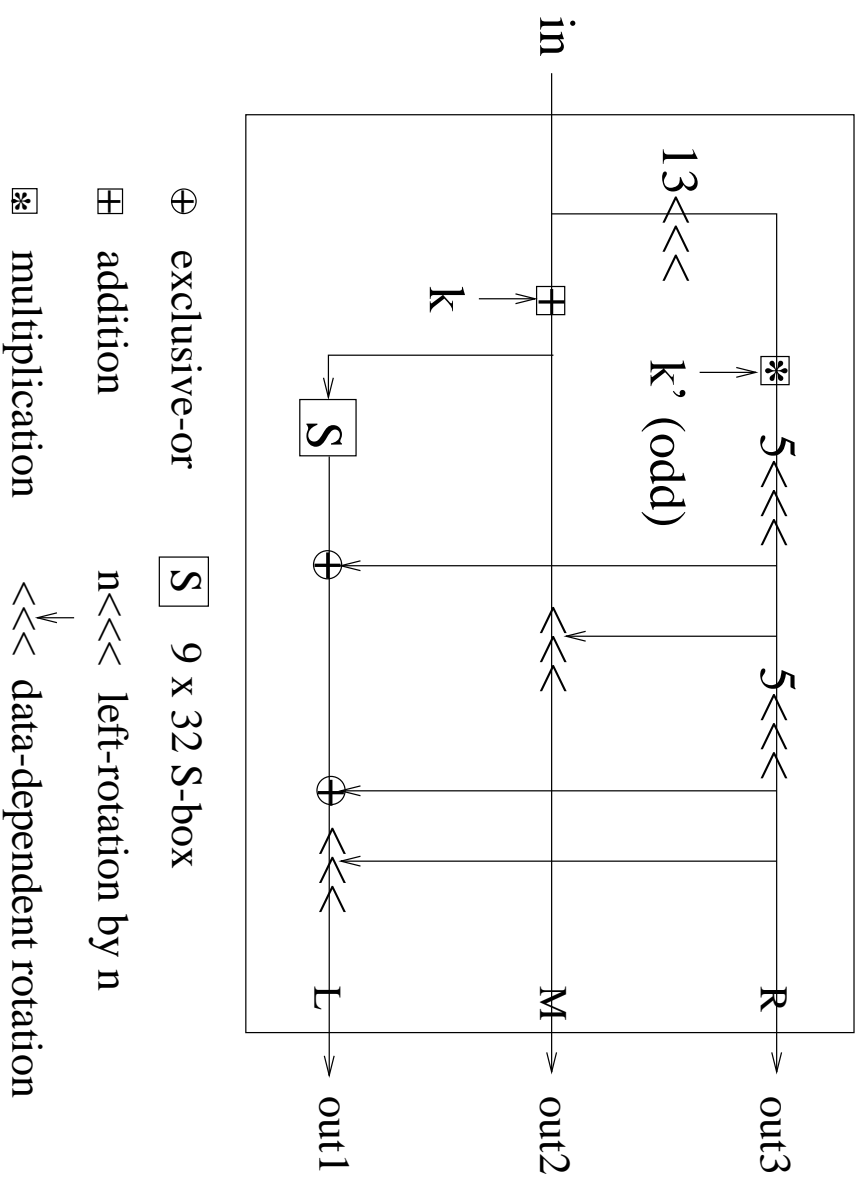


Figure 4: The E-function of the main keyed transformation

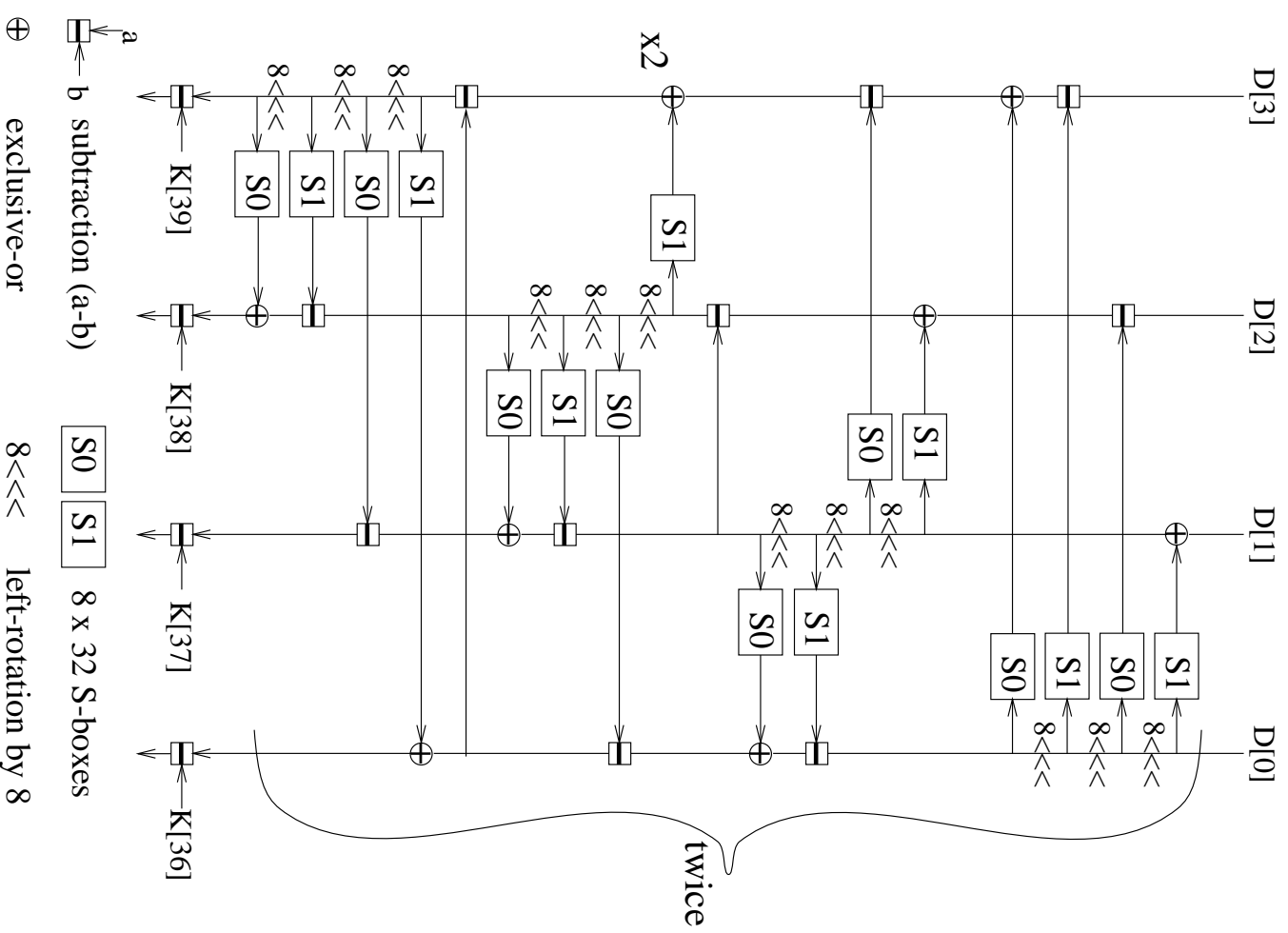


Figure 5: Structure of the backwards mixing phase

Mars

Extension of the Feistel structure

Design team includes 11 researchers,
None of them considers it as 'his design'

Consists of 4 separate parts (different hardware)

Very long key setup

Uses XOR, table lookup, bit rotation, 32-bit addition and multiplication

Fast on MMX

Complex analysis (errors in the specification)

Complex in hardware

Impossible on smartcards (code size, execution time, RAM)

Serpent

Based on S/P-structure

Uses XOR, AND, ... and bit rotations

Many operations (too many ?) → slow

No attempt at realistic tradeoff

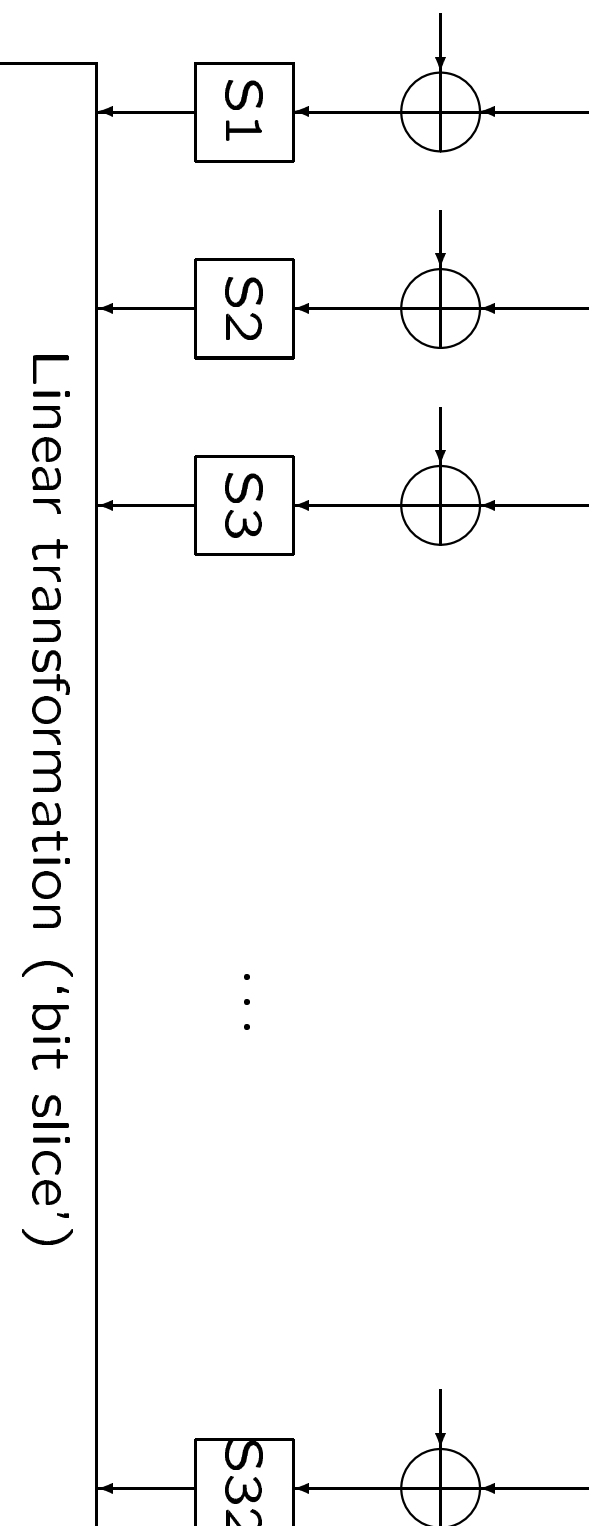
Simple operations

Relatively simple structure

Secure

Slow on most platforms (very slow)

Serpent



Rijndael design philosophy

Simplicity

- 1 S-box
- simple and fast key schedule
- no arithmetic operations; only XOR and table-lookup

Modularity

- composed of small components with clear interaction

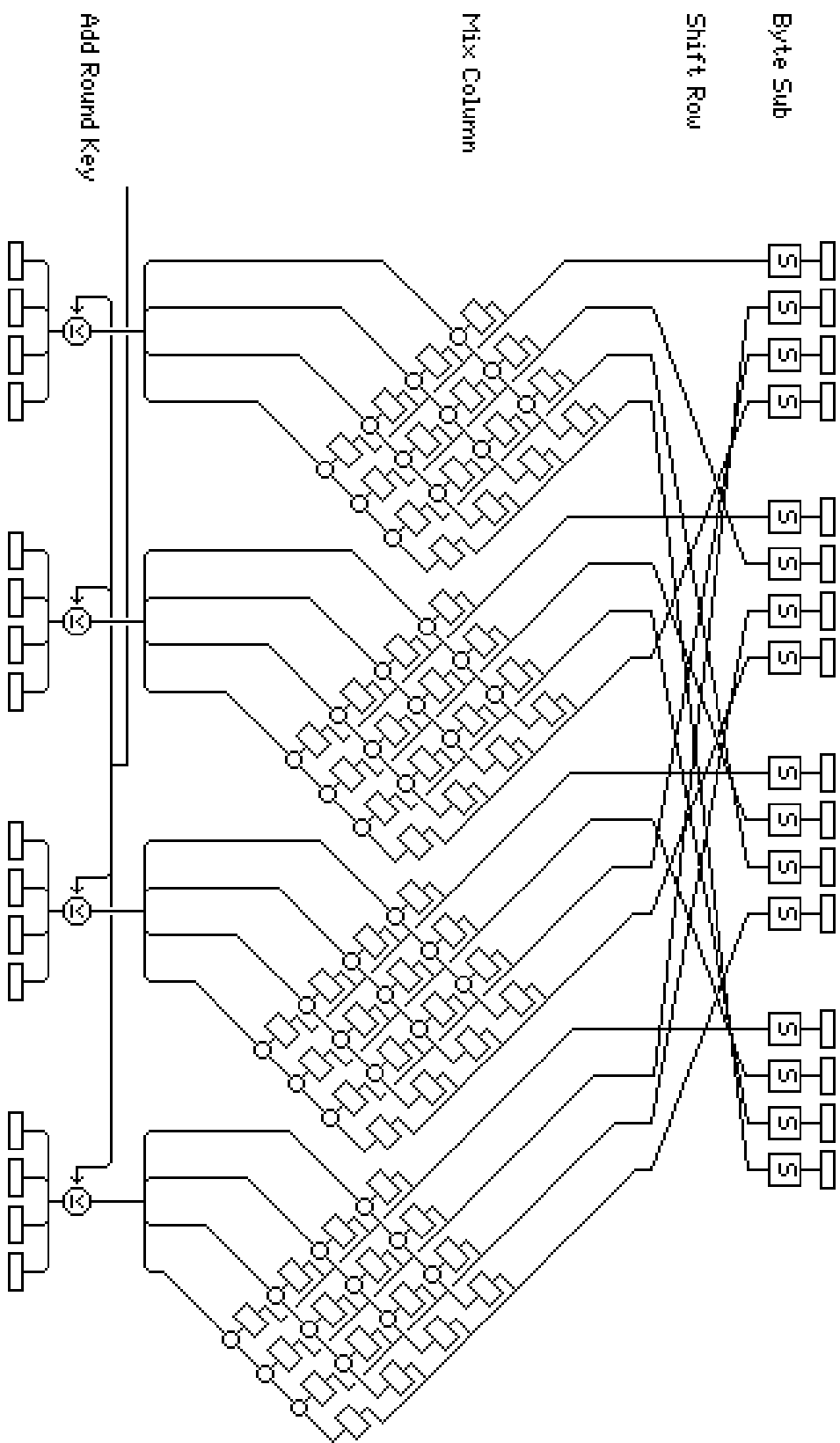
Rijndael structure

Based on S/P-structure

10/12/14 iterations of the round transformation

uniform and parallel round transformation, composed of:

- byte substitution
- shift rows
- mix columns
- round key addition



Rijndael Features

Symmetrical and parallel structure

Compact and fast on smartcards

Fast on high-end processors

Encryption and decryption use different routines

S-box is not very small in hardware

Too simple ???

Third AES Conference (New York 2000)

No breakthrough in cryptanalysis

Mars is very fast in IBM's latest GaAs technology

IP issues (?)

Hitachi patent on the use of bit rotations for encryption.

Applicable to Mars, RC6, Serpent, Twofish (?)

Third AES Conference

Rijndael never performs the worst of the 5 finalists

Rijndael is usually among the best

Rijndael is sometimes way ahead of the pack

Rijndael is the public's favourite

NIST's conclusion

- All 5 finalists offer adequate security.
- Rijndael is selected because of
 - its consistently good performance and
 - its flexibility.

Status now

Software implementations everywhere

Commercial hardware implementations underway

Many cryptographers looking at Rijndael

Many new encryption algorithms re-use Rijndael components (e.g. IBM's newest encryption algorithm: 'Scream')