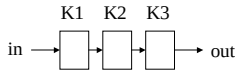
ECE 297:11 Lecture 7

Advanced Encryption Standard

Why a new standard?

1. Old standard insecure against brute-force attacks
2. Straightforward fixes lead to inefficient implementations
   - Triple DES



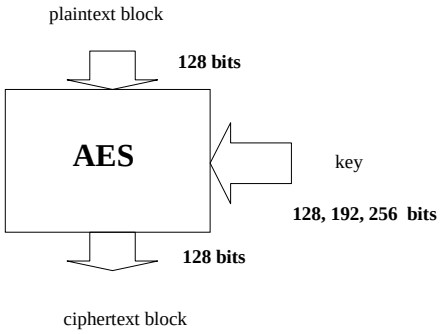
3. New trends in fast software encryption
   - use of basic instructions of the microprocessor
4. New ways of assessing cipher strength
   - differential cryptanalysis
   - linear cryptanalysis

Why a contest?

- Focus the effort of cryptographic community

  Small number of specialists in the open research
- Stimulate the research on methods of constructing secure ciphers
- Avoid backdoor theories
- Speed-up the acceptance of the standard

External format of the AES algorithm



Rules of the contest

Each team submits

Detailed cipher description	Justification of design decisions	Tentative results of cryptanalysis

Source code in C	Source code in Java	Test vectors

AES Contest Effort

June 1998

15 Candidates
from USA, Canada, Belgium, France, Germany, Norway, UK, Isreal, Korea, Japan, Australia, Costa Rica

Round 1
Security
Software efficiency

August 1999

5 final candidates
Mars, RC6, Rijndael, Serpent, Twofish

Round 2
Security
Hardware efficiency

October 2000

1 winner: Rijndael
Belgium

AES contest - First Round

15 June 1998	Deadline for submitting candidates **21** submissions, **15** fulfilled all requirements
August 1998	1st AES Conference in Ventura, CA Presentation of candidates
March 1999	2nd AES Conference in w Rome, Italy Review of results of the First Round analysis
August 1999	NIST announces five final candidates

AES: Candidate algorithms

North America (8)

Canada:
CAST-256
Deal

USA:
Mars
RC6
Twofish
Safer+
HPC

Costa Rica:
Frog

Europe (4)

Germany:
Magenta

Belgium:
Rijndael

France:
DFC

Israel, GB, Norway:
Serpent

Asia (2)

Korea:
Crypton

Japan:
E2

Australia (1)

Australia:
LOKI97

First round June 1998-August 1999

Security

Resistance
to known attacks,
randomness tests

Software implementations

PC

Smart cards

Survey filled by 104 participants of the Second AES Conference in Rome, March 1999

1. Rijndael	**+76**	
2. RC6	**+73**	
3. Twofish	**+61**	**Overwhelming YES**
4. Mars	**+52**	
5. Serpent	**+45**	
6. E2	**+14**	**Mild YES**
7. CAST-256	-2	
8. Safer+	-4	Middle-of-the-Road
9. DFC	-5	
10. Crypton	-15	Mild NO
11. DEAL	-70	
12. HPC	-77	
13. Magenta	-83	Overwhelming NO
14. Loki97	-85	
15. Frog	-85	

AES Finalists (1)

USA

Mars - IBM
C. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, S. M. Matyas, L. O'Connor, M. Peyravian, D. Safford, N. Zunic

RC6 - RSA Data Security, Inc.
R. Rivest - MIT
M. Robshaw, R. Sidney, Y. L. Yin - RSA

Twofish - Counterpane Systems
B. Schneier, J. Kelsey, C. Hall, N. Ferguson - Counterpane, D.Whiting - Hi/fn, D. Wagner - Berkeley

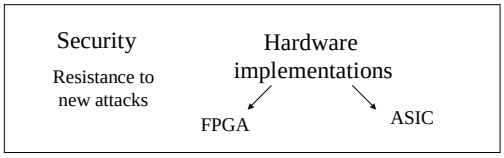
AES Finalists (2)

Europe

Rijndael - J. Daemen, V. Rijmen
Katholieke Universiteit Leuven
Belgium

Serpent - R. Anderson, **Cambridge**, **England**
E. Biham - **Technion**, **Israel**
L. Knudsen, **University of Bergen**, **Norway**

Second round August 1999-August 2000



AES contest: Second Round

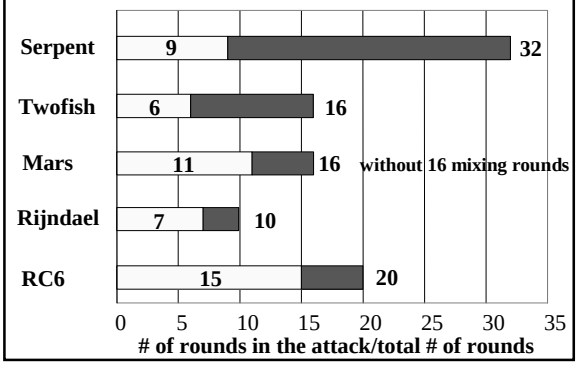
- 13-14 April 2000
  3rd AES Conference in New York
- 15 May 2000
  End of the comment period for Round II
- **2 October 2000 Winner announced**
- November 2001 FIPS-197: AES announced
- May 2002 Standard becomes effective

How NIST has made a final decision?

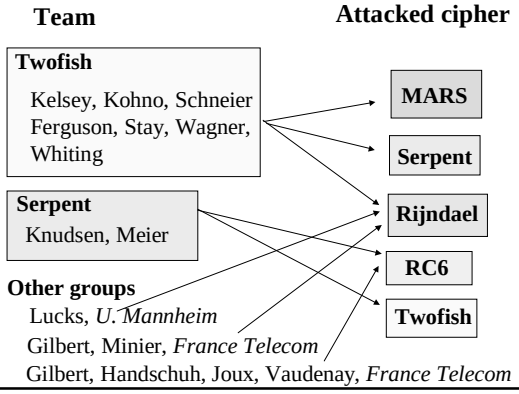
BASIC CRITERIA =

security
software efficiency
hardware efficiency
flexibility

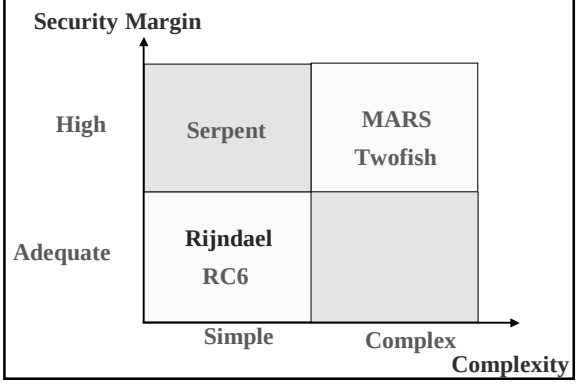
Security: Theoretical attacks better than exhaustive key search



Security: Authors of attacks

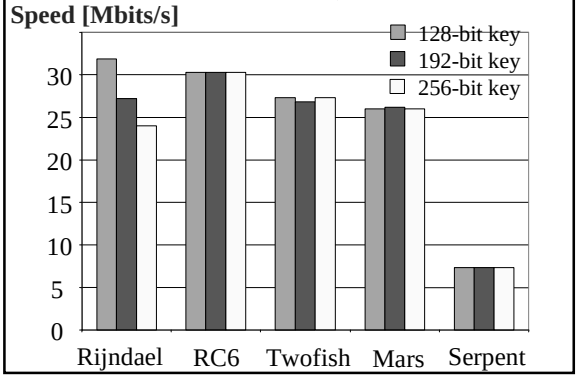


NIST Report: Security



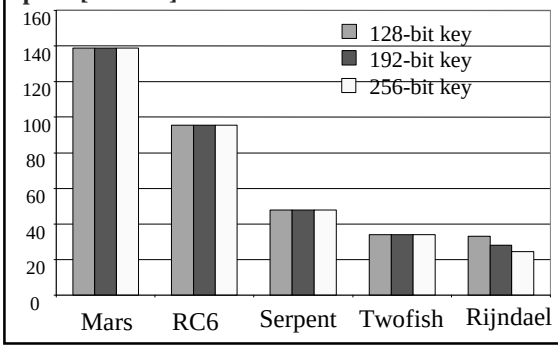
Efficiency in software: NIST-specified platform

200 MHz Pentium Pro, Borland C++



Efficiency in software: NIST tests

450 MHz Pentium II, DJGPP gcc



Efficiency in software: Ranking of encryption speeds for various platforms

	Intel						Alpha			Sun-Sparc	H-P
Mars	4	4	2	4	3	2	3	4	4	3	3
RC6	**1**	3	**1**	**1**	4	**1**	4	3	3	5	4
Twofish	2	**1**	3	2	**1**	4	2	2	2	2	2
Rijndael	3	2	4	3	2	3	**1**	**1**	**1**	**1**	**1**
Serpent	5	5	5	5	5	5	5	5	5	4	5

NIST Report: Software Efficiency

Encryption and Decryption Speed

	32-bit processors	64-bit processors	DSPs
high	RC6	Rijndael Twofish	Rijndael Twofish
medium	Rijndael Mars Twofish	Mars RC6	Mars RC6
low	Serpent	Serpent	Serpent

NIST Report: Software Efficiency

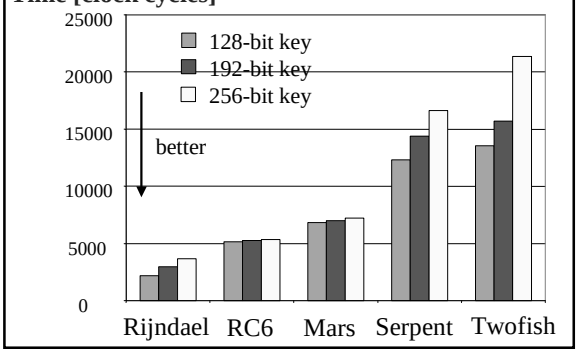
Encryption and decryption speed in software on smart cards

	8-bit processors	32-bit processors
high	Rijndael	Rijndael RC6
medium	RC6 Mars Twofish	Mars
low	Serpent	Twofish Serpent

Efficiency in software: Key setup

200 MHz Pentium Pro, Borland C++

Time [clock cycles]



NIST Report: Software Efficiency

Key scheduling

	32-bit processors	64-bit processors	DSPs
high	Rijndael	Rijndael	Rijndael Serpent
medium	Mars RC6	RC6 Serpent	Mars RC6
low	Serpent Twofish	Mars Twofish	Twofish

NIST Report: Software Efficiency

Key scheduling on smart cards

	8-bit processors
high	Rijndael
medium	Mars Twofish
low	RC6 Serpent

Efficiency in software

Strong dependence on:

1. Instruction set architecture (e.g., variable rotations)
2. Programming language (assembler, C, Java)
3. Compiler
4. Programming style

Efficiency in software: Conclusions

Encryption/decryption

- **Strong variation of results**
- **Serpent the worst for majority of platforms**

Key setup

- **Moderate variation of results**
- **Rijndael and RC6 the best for majority of platforms**
- **Twofish and Serpent the worst for majority of platforms**

Primary ways of implementing cryptography in hardware

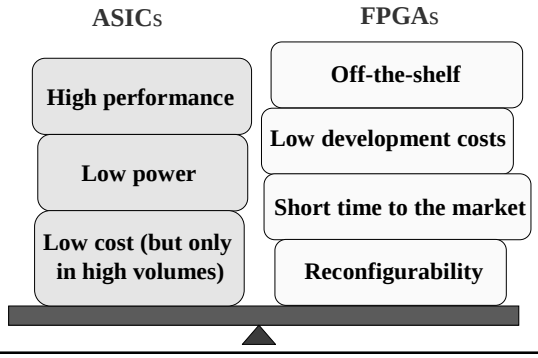
ASIC
Application **S**pecific
Integrated **C**ircuit

- designs must be sent for expensive and time consuming fabrication in semiconductor foundry
- designed all the way from behavioral description to physical layout

FPGA
Field **P**rogrammable
Gate **A**rray

- bought off the shelf and reconfigured by designers themselves
- no physical layout design; design ends with a bitstream used to configure a device

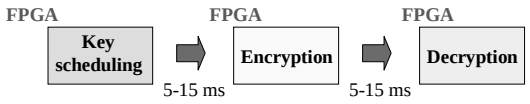
Which way to go?



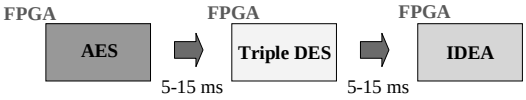
Reconfigurability

External ROM and microprocessor enables changing an FPGA function in several milliseconds

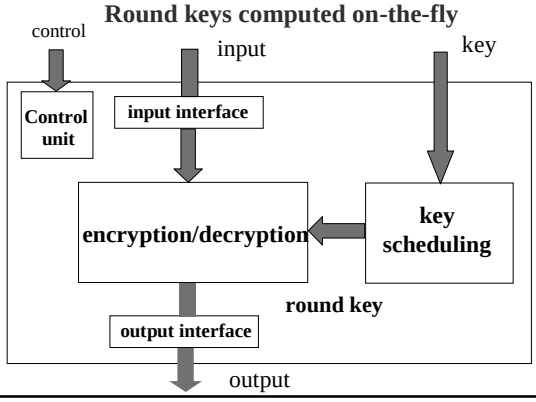
Encryption vs. decryption vs. key scheduling



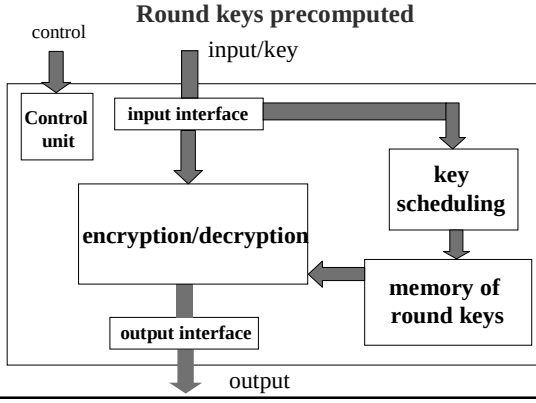
Various algorithms



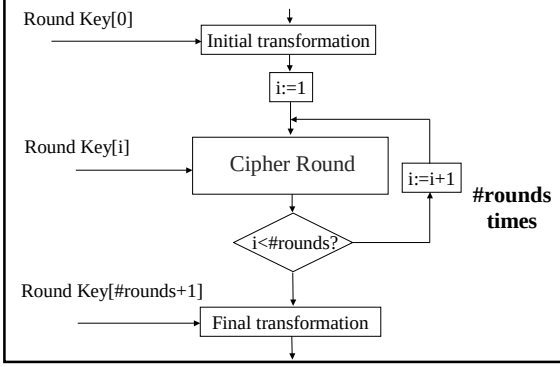
Implementation of a secret-key cipher
Round keys computed on-the-fly



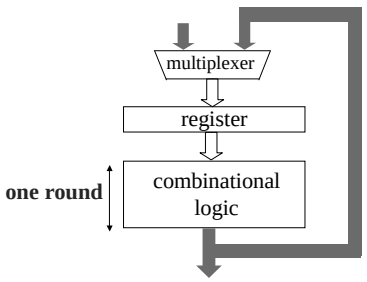
Implementation of a secret-key cipher
Round keys precomputed



Typical Flow Diagram of a Secret-Key Block Cipher



Basic iterative architecture

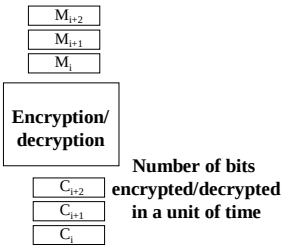


Primary parameters of hardware implementations for secret-key block ciphers

Latency

Time to encrypt/decrypt a single block of data

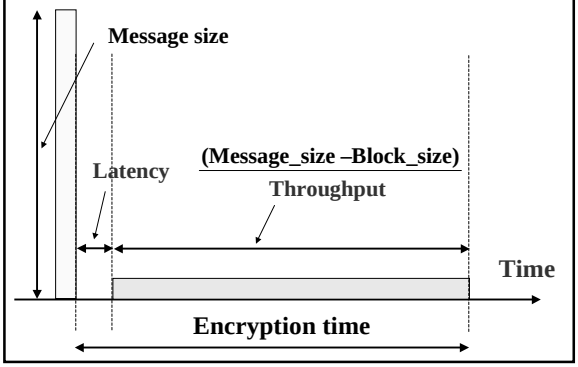
Throughput



Number of bits encrypted/decrypted in a unit of time

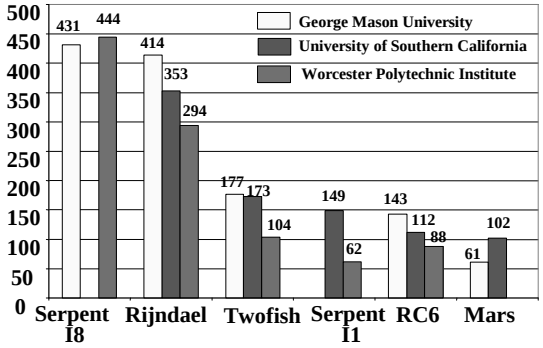
$$\text{Throughput} = \frac{\text{Block\_size} \cdot \text{Number\_of\_blocks\_processed\_simultaneously}}{\text{Latency}}$$

Dependence of the encryption time on latency and throughput



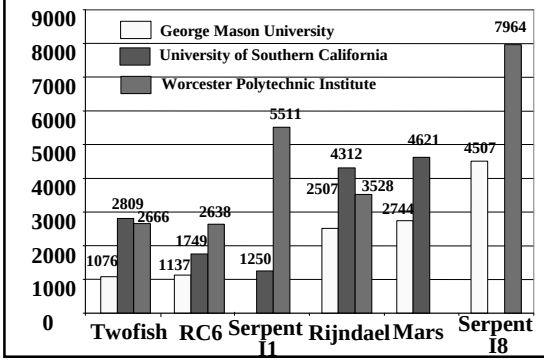
Efficiency in hardware: FPGA Virtex 1000: Speed

Throughput [Mbit/s]

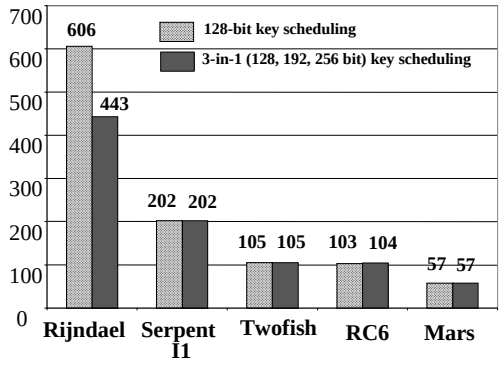


Efficiency in hardware: FPGA Virtex 1000: Area

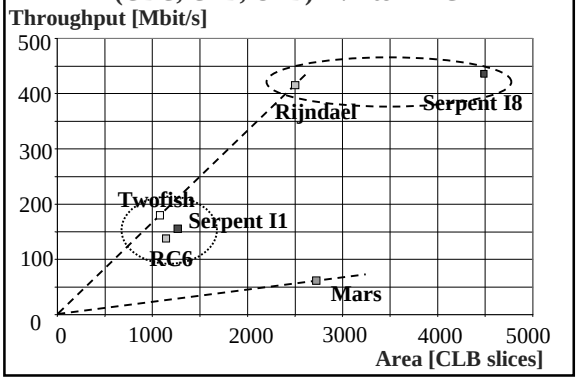
Area [CLB slices]



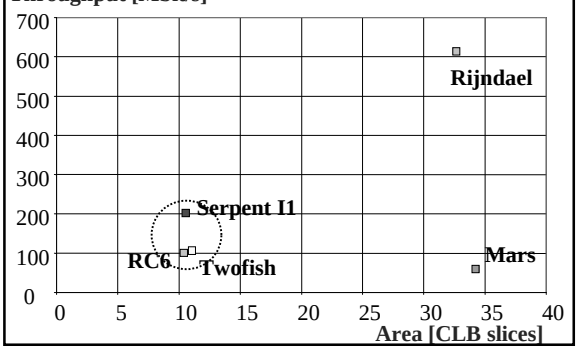
ASIC implementations: NSA group



GMU Results: Encryption in cipher feedback modes (CBC, CFB, OFB) - Virtex FPGA

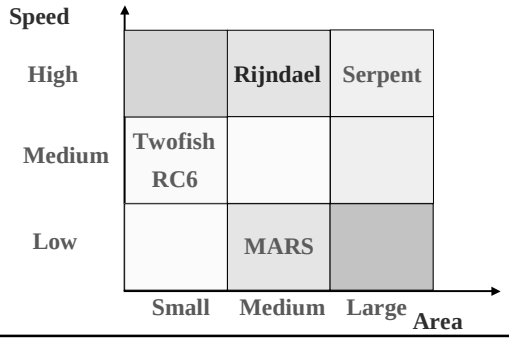


NSA Results: Encryption in cipher feedback modes (CBC, CFB, OFB) - ASIC, 0.5 µm CMOS



NIST Report + GMU Report: Hardware Efficiency

Feedback cipher modes: CBC, CFB

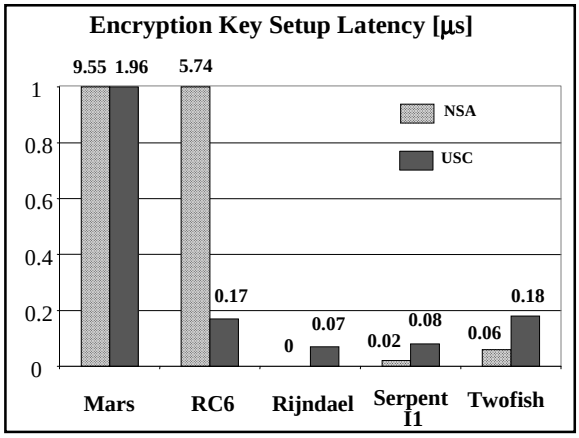


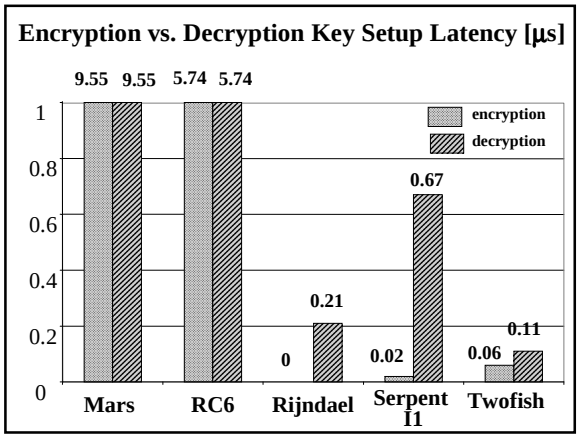
Conclusions for feedback cipher modes (1) (CBC, CFB, OFB)

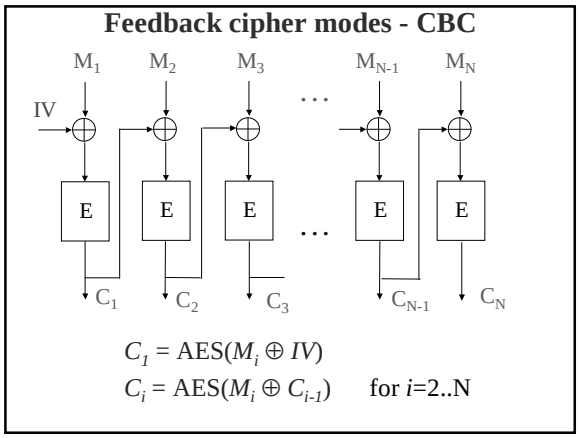
- Speed (throughput) should be the primary criteria of comparison
- Basic iterative architecture is the most appropriate for comparison and future implementations
- Serpent and Rijndael are over twice as fast as the next best candidate for all implementations

Conclusions for feedback cipher modes (2) (CBC, CFB, OFB)

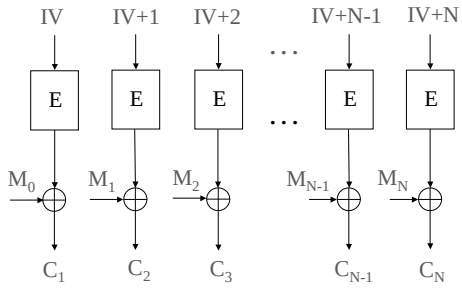
- Results confirmed by
  - three independent university groups for FPGAs, and
  - NSA group for ASICs
- Results of comparison independent of implementation technology (FPGAs vs. ASICs)


Encryption Key Setup Latency [µs]
9.55
1.96
5.74
0.17
0
0.07
0.02
0.08
0.06
0.18
NSA
USC
1
0.8
0.6
0.4
0.2
0
Mars
RC6
Rijndael
Serpent I1
Twofish


Encryption vs. Decryption Key Setup Latency [µs]
9.55
9.55
5.74
5.74
0
0.21
0.02
0.67
0.06
0.11
encryption
decryption
1
0.8
0.6
0.4
0.2
0
Mars
RC6
Rijndael
Serpent I1
Twofish

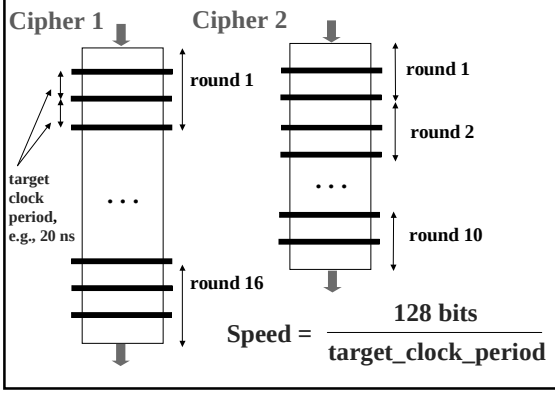

Feedback cipher modes - CBC
M1
M2
M3
MN-1
MN
IV
E
C1
C2
C3
CN-1
CN
C1 = AES(Mi ⊕ IV)
Ci = AES(Mi ⊕ Ci-1) for i=2..N

Non-feedback Counter Mode - CTR



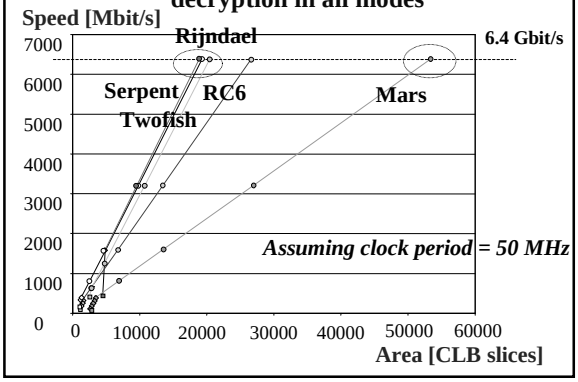
$C_i = M_i \oplus AES(IV+i)$ for i=0..N

Full Mixed Inner and Outer-Round Pipelining



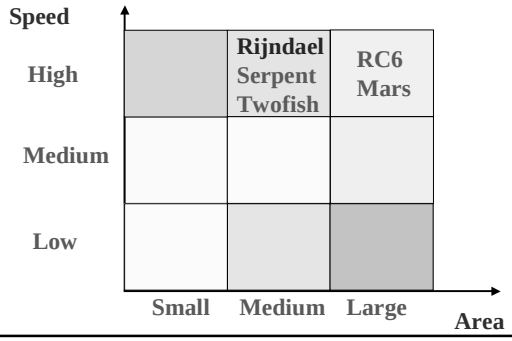
$$\text{Speed} = \frac{128 \text{ bits}}{\text{target\_clock\_period}}$$

Encryption in non-feedback modes (ECB, counter) decryption in all modes



NIST Report + GMU Report: Hardware Efficiency

Non-feedback cipher modes: ECB, CTR



Conclusions for non-feedback cipher modes (1) ECB, counter

- All ciphers can achieve approximately the same speed.
  Area should be the primary criteria of comparison.
- Serpent, Twofish and Rijndael are the most cost-efficient and take approximately the same amount of area

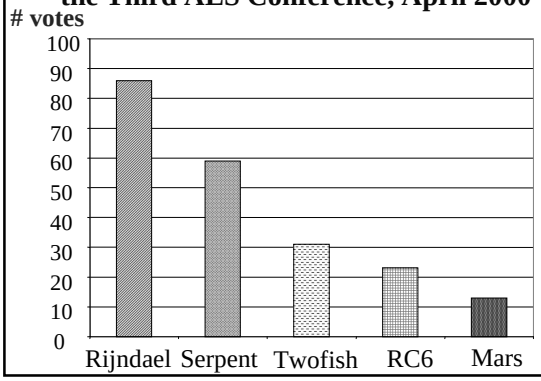
Importance of the AES candidate hardware efficiency comparison

- Important factor used to differentiate among final candidates
  - objective and commonly accepted measures
  - good agreement among results from various groups
  - large differences among final candidates
- Efficient architectures and methodologies developed for all algorithms

Flexibility: Criteria

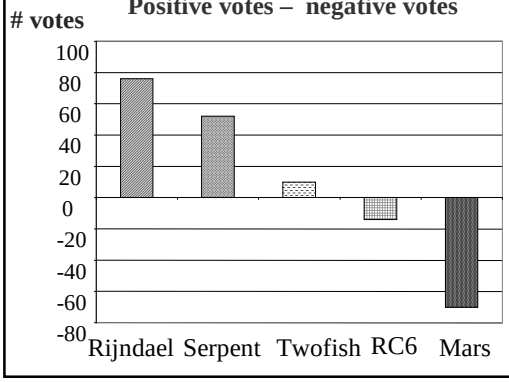
- Additional key-sizes and block-sizes
- Ability to function efficiently and securely in a wide variaty of platforms and applications
  low-end smartcards, wireless - memory requirements
  IPSec, ATM - key setup time in hardware
  B-ISDN, satellite communication - encryption speed

Survey filled by 167 participants of the Third AES Conference, April 2000



Ranking by participants of the AES3 Conference

Positive votes – negative votes



Most likely winner(s) (1)

Rijndael

+	–
• fastest in hardware	• security margin
• close to the fastest in software	
• very high flexibility	

novel ideas

Most likely winner(s) (2)

Serpent

+	–
• large security margin	• slow in software
• conservative construction	• moderate flexibility
• very fast in hardware	
• cryptanalytical reputation of authors	

Most likely winner(s) (3)

Twofish

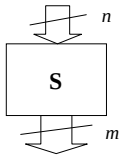
+	–
• good security margin	• moderately fast in hardware
• fast encryption/decryption in software	• slow key setup in software
• American	• moderate flexibility
• strongly advertized	

Major operations of AES finalists

	Serpent	Rijndael	Twofish	RC6	Mars
S-boxes	✕	✕	✕		✕
Multiplication in GF(2^m)		✕	✕		
Integer multiplication				✕	✕
Variable rotation				✕	✕

Basic cipher operations (1) - S-box

S-box *n* x *m*



Software

C

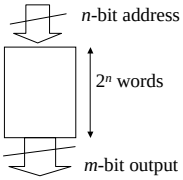
```
WORD S[1<<n]=
{ 0x23, 0x34, 0x56
  . . . . . .
}
```

ASM

```
S   DW  23H, 34H,
        56H
        .....
```

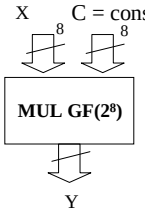
Hardware

ROM



direct logic

Basic cipher operations (2)
Multiplication in the Galois Field GF(2m)



Software

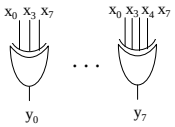
C

<<, ^, |, &

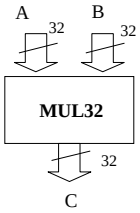
ASM

ROL, XOR, OR, AND

Hardware



Basic cipher operations (3) - Multiplication



Software

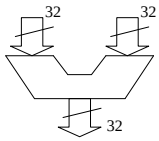
C

```
unsigned long  A, B, C;
C = A*B;
```

ASM

MUL

Hardware



Half-Multiplier

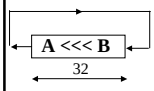
$C=A \cdot B \bmod 2^{32}$

Basic cipher operations (4) - Rotations

Software

C

```
C = (A << B) | (A >> (32-B));
```



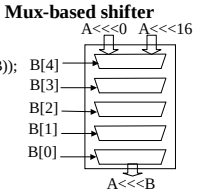
variable rotation ROL32

ASM

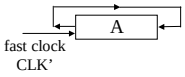
ROL A, B

Hardware

Mux-based shifter



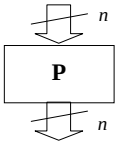
High-speed clock



min (B, 32-B) CLK' cycles

Auxiliary cipher operations - Permutation

Software



Permutation

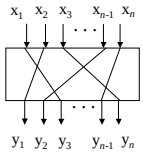
C

complex sequence of instructions <<, |, &

ASM

complex sequence of instructions ROL, OR, AND

Hardware



order of wires

AES: Types of candidate algorithms

Feistel Networks

Twofish
E2
DFC
Deal
LOKI97
Magenta

Modified Feistel Network

RC6
MARS
CAST-256

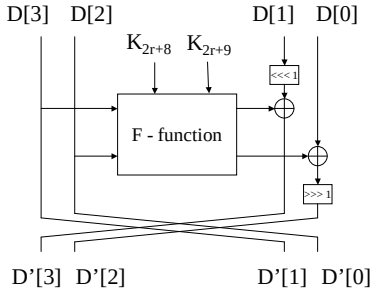
Substitution-Linear Transformation Networks

Rijndael
Serpent
Safer+
Crypton

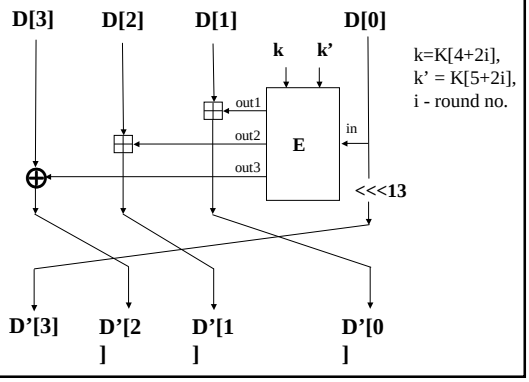
Others

Frog
HPC

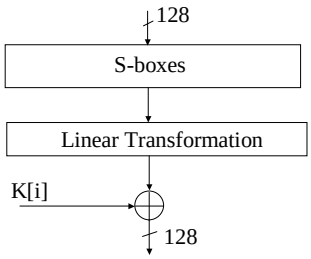
Feistel Network: Single Round of Twofish



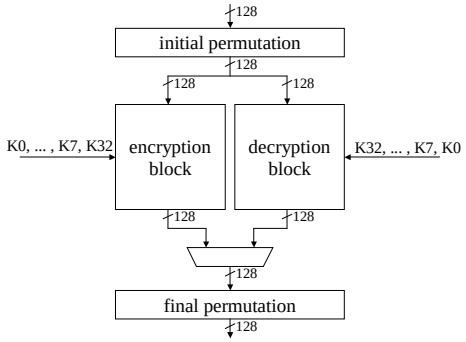
Modified Feistel Network: Single Round of MARS



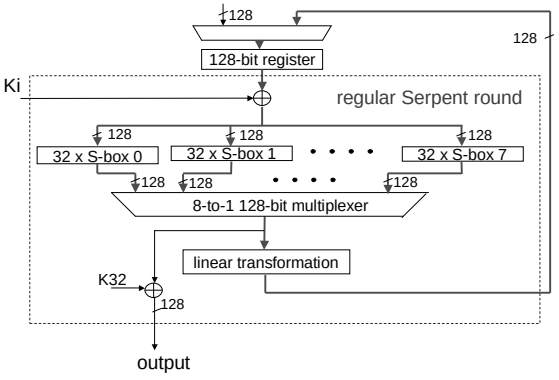
Substitution-Linear Transformation Network: Single Round of Serpent



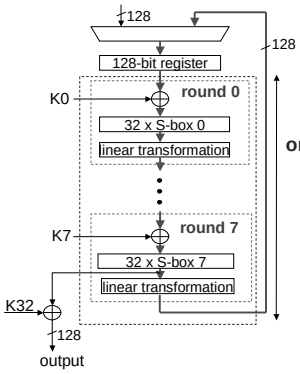
Substitution-Linear Transformation Network: Serpent in Hardware



First basic architecture of Serpent - Serpent I1

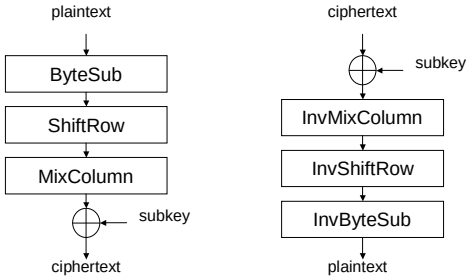


Alternative basic architecture of Serpent: Serpent I8

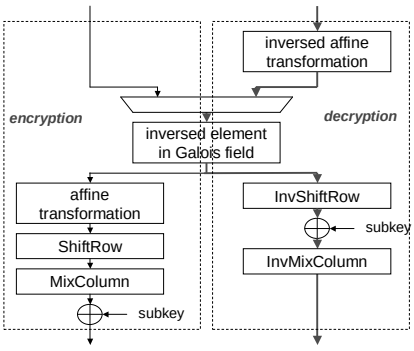


one implementation round of Serpent = 8 regular cipher rounds

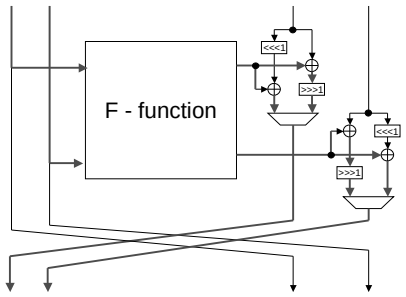
Rijndael – Different Circuits for Encryption and Decryption



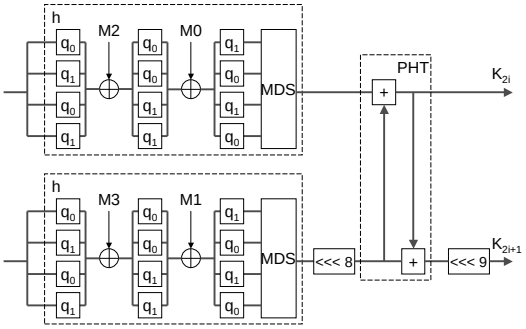
Rijndael – Resource Sharing Between Encryption and Decryption



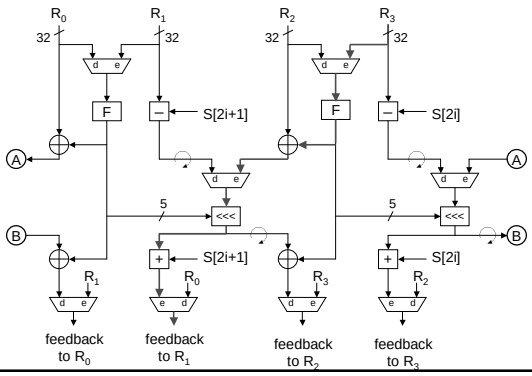
Twofish – Encryption/Decryption Round



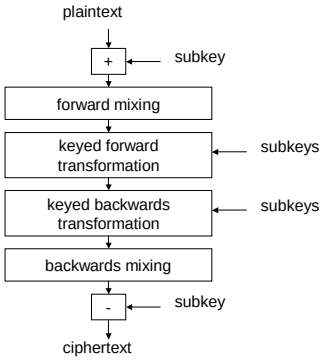
Twofish – F-function



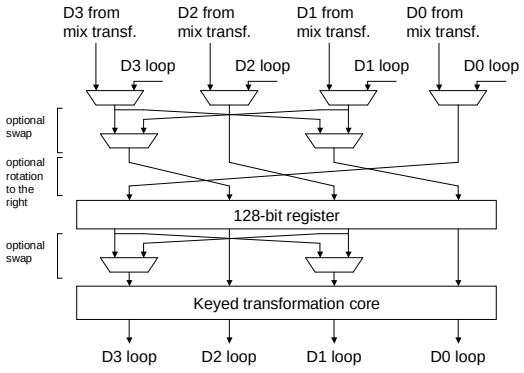
RC6 Round



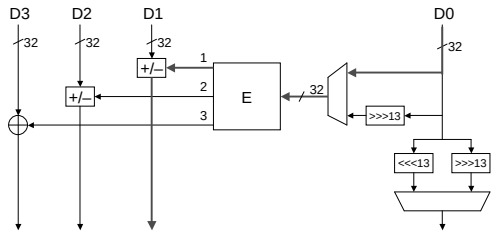
MARS - General Structure



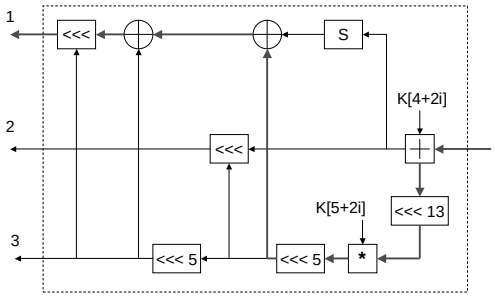
MARS - Keyed Transformation



MARS - Keyed Transformation Core



MARS – E-function



Secret-key cryptography standards

Federal standards	Banking standards	International standards
NIST	**ANSI**	**ISO**
FIPS 46-1 DES FIPS 46-2 DES	X3.92 DES	ISO 8732 Modes of operation of a 64-bit cipher
FIPS 81 Modes of operation	X3.106 DES modes of operation	
FIPS 46-3 Triple DES	**X9.52 Modes of operation of Triple DES**	**ISO 10116 Modes of operation of an n-bit cipher**
AES		

Operating Modes Contest

4 Old Modes
(CBC, CFB, OFB, ECB)

April 2001

10 New Candidates
from Egypt, Estonia, Norway, Sweden, Thailand, USA

Counter mode

Summer 2001

5 Standard Modes

2002

New Standard Modes

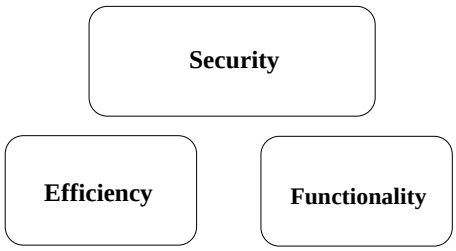
Modes submitted to the contest (1)

	Full name	Authors	Institution
2DEM	2D-Encryption Mode	A. A. Belal, M. A. Abdel-Gawad	Alexandria University, **Egypt**
ABC	Accumulated Block Chaining	L. Knudsen	U. of Bergen **Norway**
CTR	Counter Mode	H. Lipmaa, P. Rogaway, D. Wagner	**Finland, Estonia, USA, Thailand**
IACBC	Integrity Aware CBC	C. Jutla	IBM, **USA**
IAPM	Integrity Aware Parallalizable Mode	C. Jutla	IBM, **USA**

Modes submitted to the contest (2)

	Full name	Authors	Institution
IGE	Infinite Garble Extension	V. D. Gligor, P. Donescu	VDG, Inc., **USA**
KFB	Key Feedback Mode	J. Håstad, M. Naslund	NADA, Ericsson **Sweden**
OCB	Offset Codebook	P. Rogaway	**UCSD, USA, Thailand**
PCFB	Propagating Cipher Feedback	H. Hellström	StreamSec, **Sweden**
XCBC	eXtended CBC Encryption	V. D. Gligor, P. Donescu	VDG, Inc., **USA**

Evaluation Criteria for Modes of Operation



Evaluation criteria (1)

Security

- resistance to attacks
- **proof of security**
- random properties of the ciphertext

Efficiency

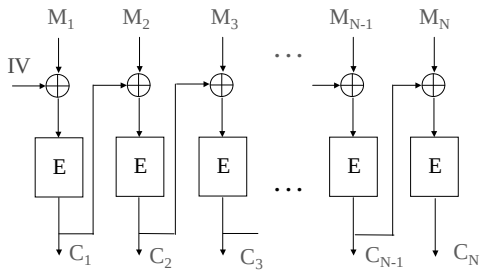
- number of calls of the block cipher
- **capability for parallel processing**
- memory/area requirements
- initialization time
- **capability for preprocessing**

Evaluation criteria (2)

Functionality

- **security services**
  - confidentiality, **integrity, authentication**
- flexibility
  - variable lengths of blocks and keys
  - different amount of precomputations
  - requirements on the length of the message
- **vulnerability to implementation errors**
- requirements on the amount of keys, initialization vectors, random numbers, etc.
- error propagation and the capability for resynchronization
- patent restrictions

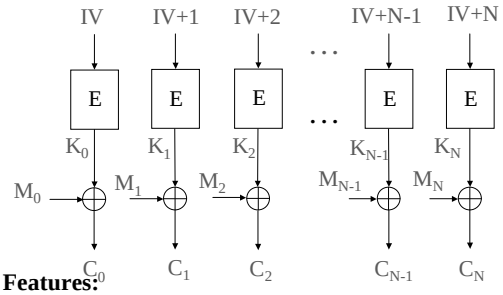
Modes of operation: Current standard - CBC



Problems:

- No parallel processing of blocks from the same packet
- No speed-up by preprocessing
- No integrity or authentication

Counter mode



Features:

+ Potential for parallel processing

+ Speed-up by preprocessing

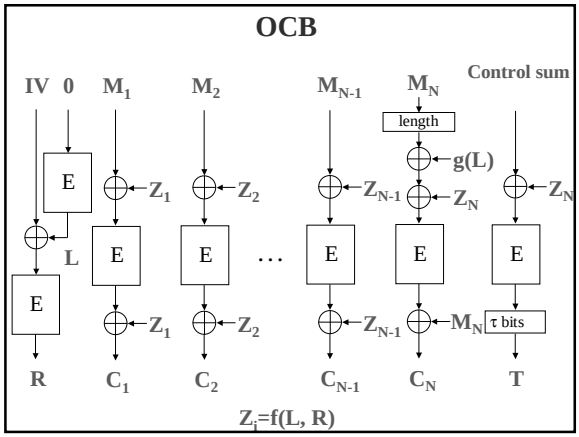
- No integrity or authentication

Properties of existing and new cipher modes

	CBC	CFB	OFB	New standard
Proof of security	✓	✓	✓	✓
Parallel processing	decryption only		–	✓
Preprocessing	–	–	✓	✓
Integrity and authentication	–	–	–	✓
Resistance to implementation errors	✓	✓	–	✓

Encryption with authentication

	Full name	Authors	Institutions
IACBC	Integrity Aware CBC	C. Jutla	**IBM (patent)**
IAPM	Integrity Aware Parallalizable Mode	C. Jutla	**IBM (patent)**
XCBC-XOR	eXtended CBC Encryption	V. D. Gligor, P. Donescu	**VDG, Inc., (patent)**
XECB-XOR	eXtended ECB Encryption	V. D. Gligor, P. Donescu	**VDG, Inc., (patent)**
OCB	Offset Codebook	P. Rogaway	**UCSD**, USA, Thailand


OCB
IV
0
M1
M2
MN-1
MN
Control sum
length
g(L)
E
Z1
Z2
ZN-1
ZN
L
…
τ bits
R
C1
C2
CN-1
CN
T
Zi=f(L, R)