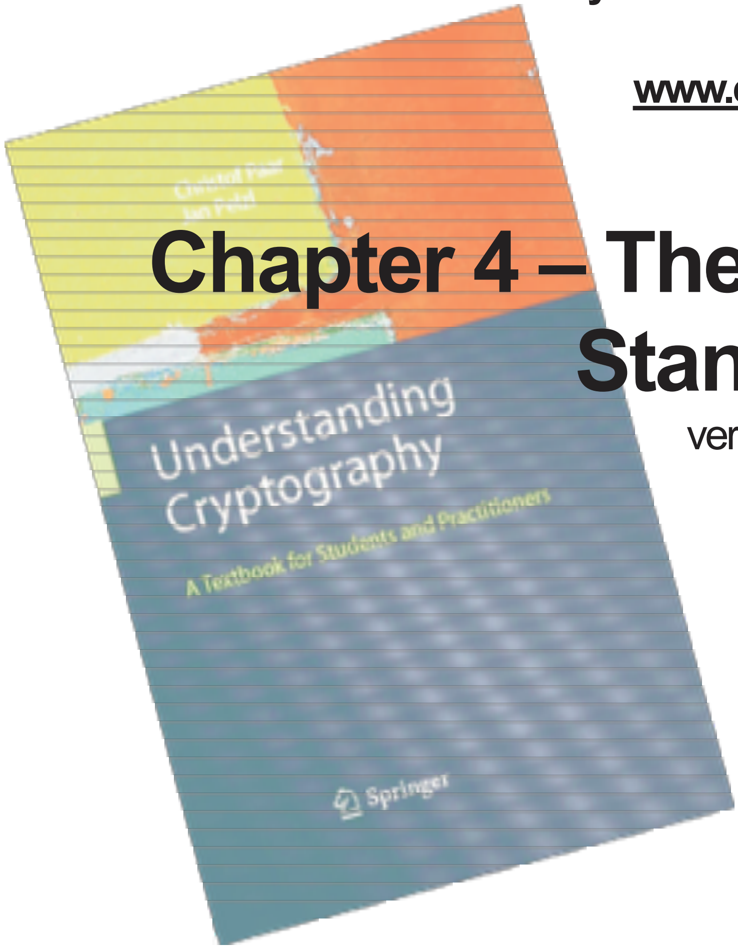
Understanding Cryptography

by Christof Paar and Jan Pelzl

www.crypto-textbook.com



Chapter 4 – The Advanced Encryption Standard (AES)

ver. October 28, 2009

These slides were prepared by Daehyun Strobel, Christof Paar and Jan Pelzl
Modified by Sam Bowne

Some legal stuff (sorry): Terms of Use

Contents of this Chapter

4.1 Introduction

Some Basic Facts

- AES is the most widely used symmetric cipher today
- The algorithm for AES was chosen by the US *National Institute of Standards and Technology* (NIST) in a multi-year selection process
- The requirements for all AES candidate submissions were:
  - Block cipher with **128-bit block size**
  - **Three supported key lengths**: 128, 192 and 256 bit
  - Security relative to other submitted algorithms
  - **Efficiency** in software and hardware

Chronology of AES Selection

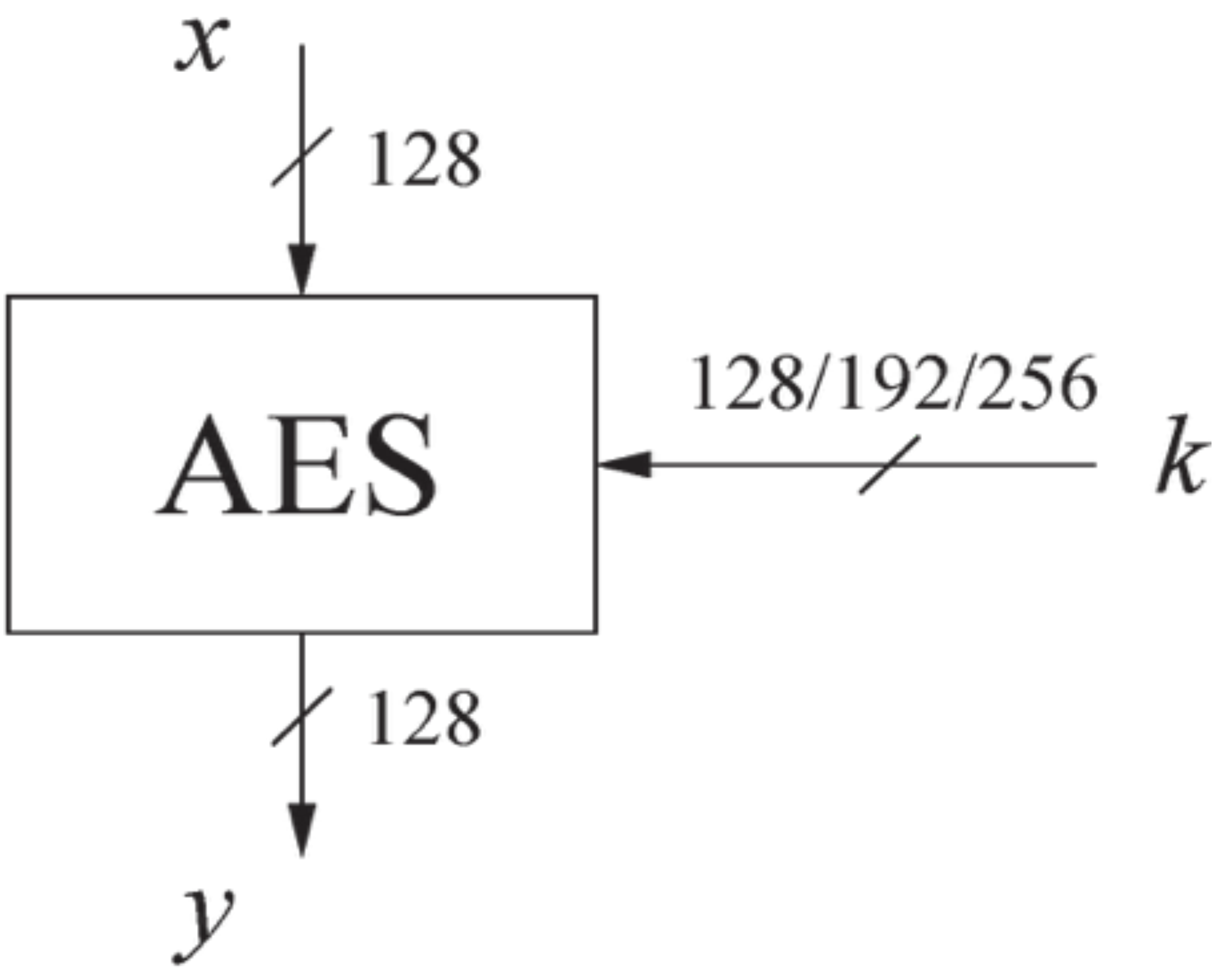
- The need for a new block cipher announced by NIST in January, 1997
- 15 candidates algorithms accepted in August, 1998
- 5 finalists announced in August, 1999:
    - *Mars* – IBM Corporation
    - *RC6* – *RSA* Laboratories
    - *Rijndael* – J. Daemen & V. Rijmen
    - *Serpent* – Eli Biham et al.
    - *Twofish* – B. Schneier et al.
- In October 2000, *Rijndael* was chosen as the AES

Chronology of AES Selection

- AES was formally approved as a US federal standard in November 2001
- In 1993, the NSA allows AES to encrypt classified documents
  - Up to SECRET for all key lengths
  - Up to TOP SECRET for 192 and 256-bit keys

4.2 Overview of the AES Algorithm

AES: Overview



The number of rounds depends on the chosen key length:

Key length (bits)	Number of rounds
128	10
192	12
256	14

AES: Overview

- Iterated cipher with 10/12/14 rounds
- Each round consists of “Layers”
- Unline DES, all 128 bits are encrypted in each round

Chapter 4 of *Understanding Cryptography* by Christof Paar and Jan Pelzl

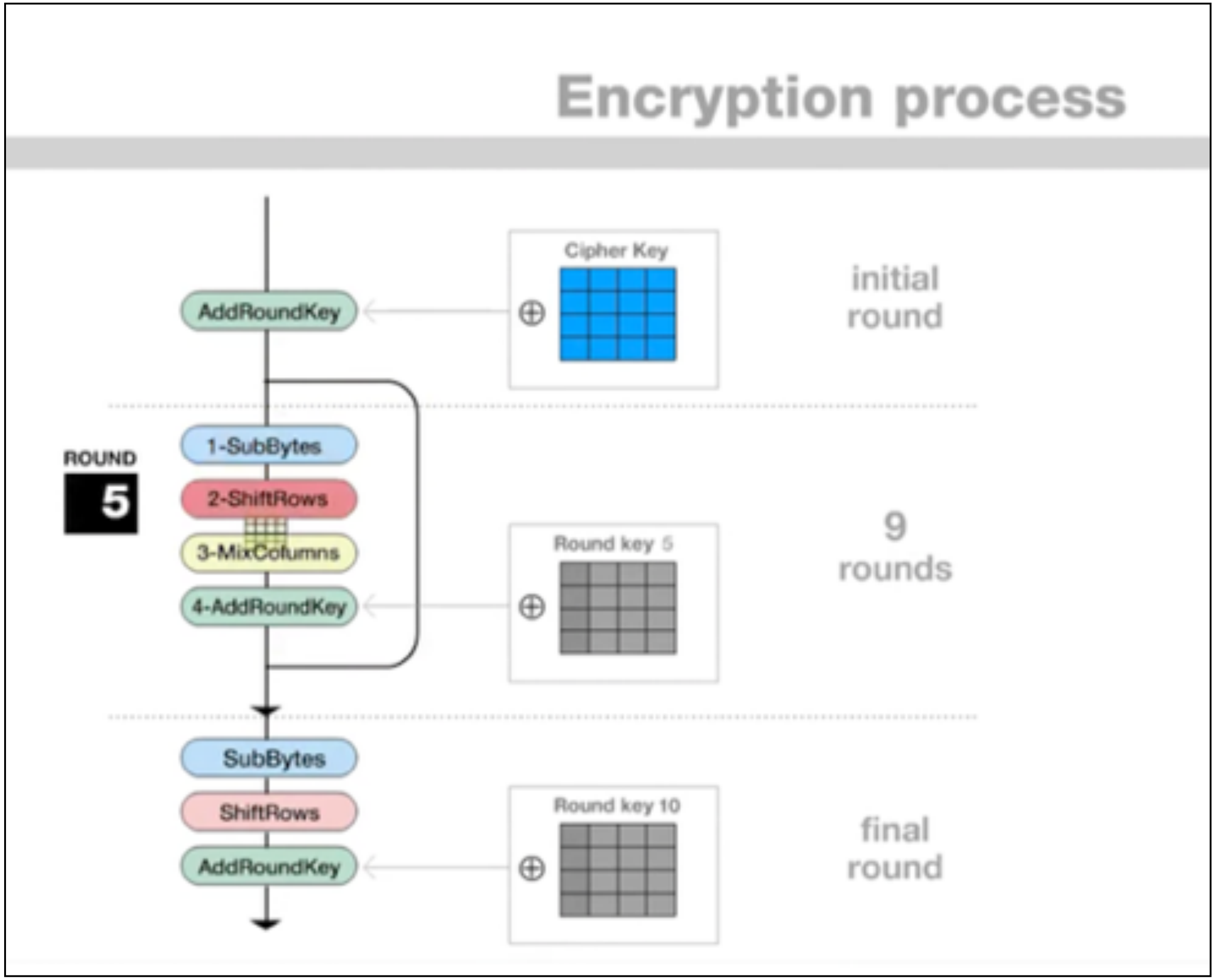
Three Layer Types

- Key Addition Layer
  - A 128-bit round key (or *subkey*)
  - Derived from the main key in the *key schedule*
  - XORed to the state
- Byte Substitution Layer (S-Box)
  - Nonlinear transformation using lookup tables
  - Introduces *confusion* to the data
    - (Obscures relationship between key and ciphertext)

Three Layer Types

- Diffusion Layer
  - Two sublayers: **ShiftRows** and **MixColumn**
    - (Makes sure that changing one plaintext bit affects many ciphertext bits)

Video: Link Ch 4a



Kahoot!

4.4 Internal Structure of AES

Internal Structure of AES

- AES is a byte-oriented cipher
- The state *A* (i.e., the 128-bit data path) can be arranged in a 4x4 matrix:

A_0	A_4	A_8	A_{12}
A_1	A_5	A_9	A_{13}
A_2	A_6	A_{10}	A_{14}
A_3	A_7	A_{11}	A_{15}

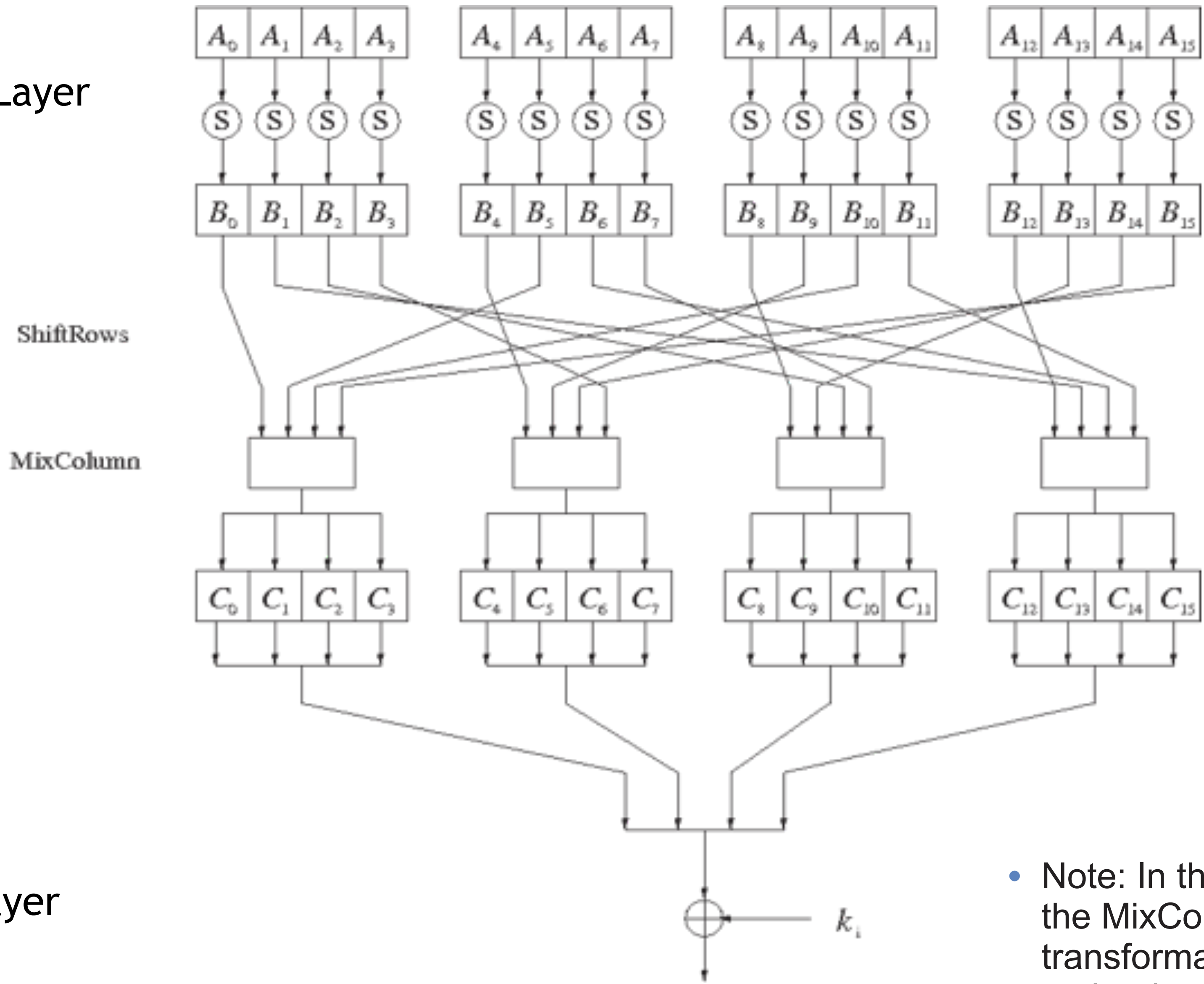
with $A_0, \ldots, A_{15}$ denoting the 16-byte input of AES

Round function for rounds 1,2,…,n_{r-1}:

Byte Substution Layer
(S-Box)

Diffusion
Layer

Key Addition Layer



- Note: In the last round, the MixColumn transformation is omitted

Byte Substitution Layer

- The Byte Substitution layer consists of 16 **S-Boxes** with the following properties:

  The S-Boxes are

  - **identical**
  - the only **nonlinear** elements of AES, i.e.,
    ByteSub(A_i) + ByteSub(A_j) $\neq$ ByteSub($A_i + A_j$), for $i,j = 0,\ldots,15$
  - **bijective**, i.e., there exists a one-to-one mapping of input and output bytes

    $\Rightarrow$ S-Box can be uniquely reversed

- In software implementations, the S-Box is usually realized as a lookup table

S-Box

Table 4.3 AES S-Box: Substitution values in hexadecimal notation for input byte (xy)

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Diffusion Layer

- provides diffusion over all input state bits

- consists of two sublayers:
  - **ShiftRows Sublayer**: Permutation of the data on a byte level
  - **MixColumn Sublayer**: Matrix operation which combines ("mixes") blocks of four bytes

- performs a linear operation on state matrices A, B, i.e., $\mathrm{DIFF}(A) + \mathrm{DIFF}(B) = \mathrm{DIFF}(A + B)$

ShiftRows Sublayer

- Rows of the state matrix are shifted cyclically:

Input matrix

B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}

Output matrix

B_0	B_4	B_8	B_{12}	no shift
B_5	B_9	B_{13}	B_1	← one position left shift
B_{10}	B_{14}	B_2	B_6	← two positions left shift
B_{15}	B_3	B_7	B_{11}	← three positions left shift

MixColumn Sublayer

- Linear transformation which mixes each column of the state matrix

- Each 4-byte column is considered as a vector and multiplied by a fixed 4x4 matrix, e.g.,

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}.$$

Key Addition Layer

- Inputs:
  - 16-byte state matrix C
  - 16-byte subkey k_i
- Output: $C \oplus k_i$
  - Combined with XOR
- The subkeys are generated in the key schedule

Key Schedule

- Subkeys are derived recursively from the original 128/192/256-bit input key
- Each round has 1 subkey, plus 1 subkey at the beginning of AES

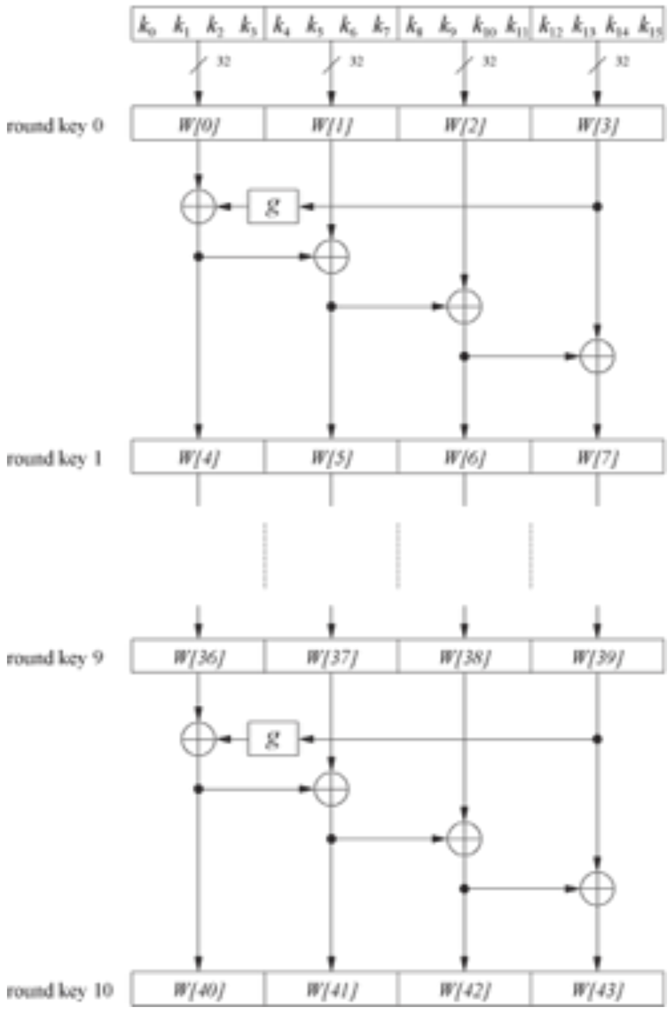
Key length (bits)	Number of subkeys
128	11
192	13
256	15

- Key whitening: Subkey is used both at the input and output of AES

  $\Rightarrow$ # subkeys = # rounds + 1
- There are different key schedules for the different key sizes

Key Schedule

Example: Key schedule for 128-bit key AES



- Word-oriented: 1 word = 32 bits
- 11 subkeys are stored in *W[0]…W[3], W[4]… W[7], … , W[40]…W[43]*
- First subkey *W[0]…W[3]* is the original AES key

Key Schedule

- Function g rotates its four input bytes and performs a bytewise
  S-Box substitution
  $\Rightarrow$ nonlinearity

- The round coefficient RC is only added to the leftmost byte and varies from round to round:

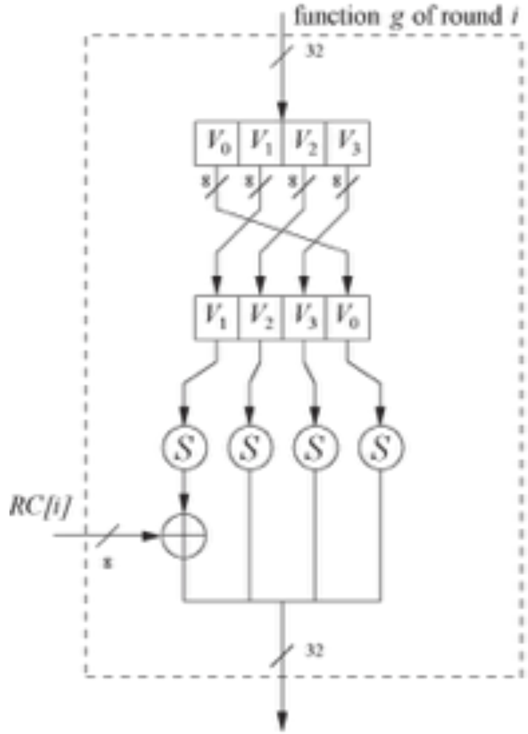
$RC[1] = x^0 = (00000001)_2$

$RC[2] = x^1 = (00000010)_2$

$RC[3] = x^2 = (00000100)_2$

...

$RC[10] = x^9 = (00110110)_2$



4.6 Implementation

Implementation in Software

- One requirement of AES was the possibility of an efficient software implementation
- Straightforward implementation is well suited for 8-bit processors (e.g., smart cards), but inefficient on 32-bit or 64-bit processors
- A more sophisticated approach: Merge all round functions (except the key addition) into one table look-up
  - This results in four tables with 256 entries, where each entry is 32 bits wide
  - One round can be computed with 16 table look-ups
- Typical SW speeds are more than 1.6 Gbit/s on modern 64-bit processors

Security

- **Brute-force attack:** Due to the key length of 128, 192 or 256 bits, a brute-force attack is not possible
- **Analytical attacks:** There is no analytical attack known that is better than brute-force
- **Side-channel attacks:**
  - Several side-channel attacks have been published
  - Note that side-channel attacks do not attack the underlying algorithm but the implementation of it

AES in Python

```
>>> from Crypto.Cipher import AES
>>> key        = "Sixteen byte key"
>>> plaintext = "secret: 16 bytes"
>>> ciphertext = cipher.encrypt(plaintext)
>>> print ciphertext.encode("hex")
1b853bed5a13d41147f03f1680a3aea3
>>>
>>>
>>> cipher.decrypt(ciphertext)
'secret: 16 bytes'
```

Kahoot!