Symmetric cryptography

Cryptography: course for master's degree in EDGE COMPUTING

Michał Melosik, PhD

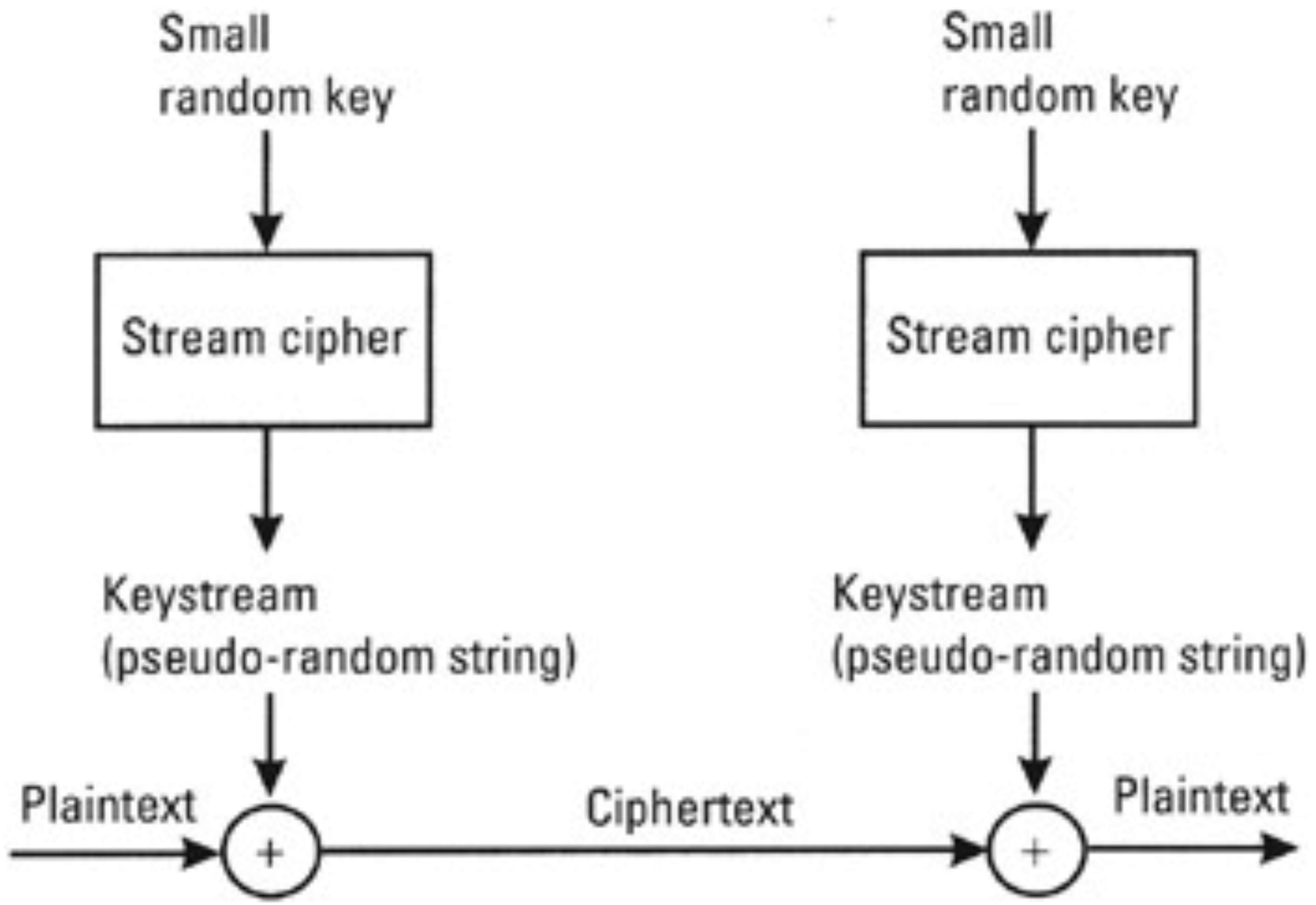
Lecture outline

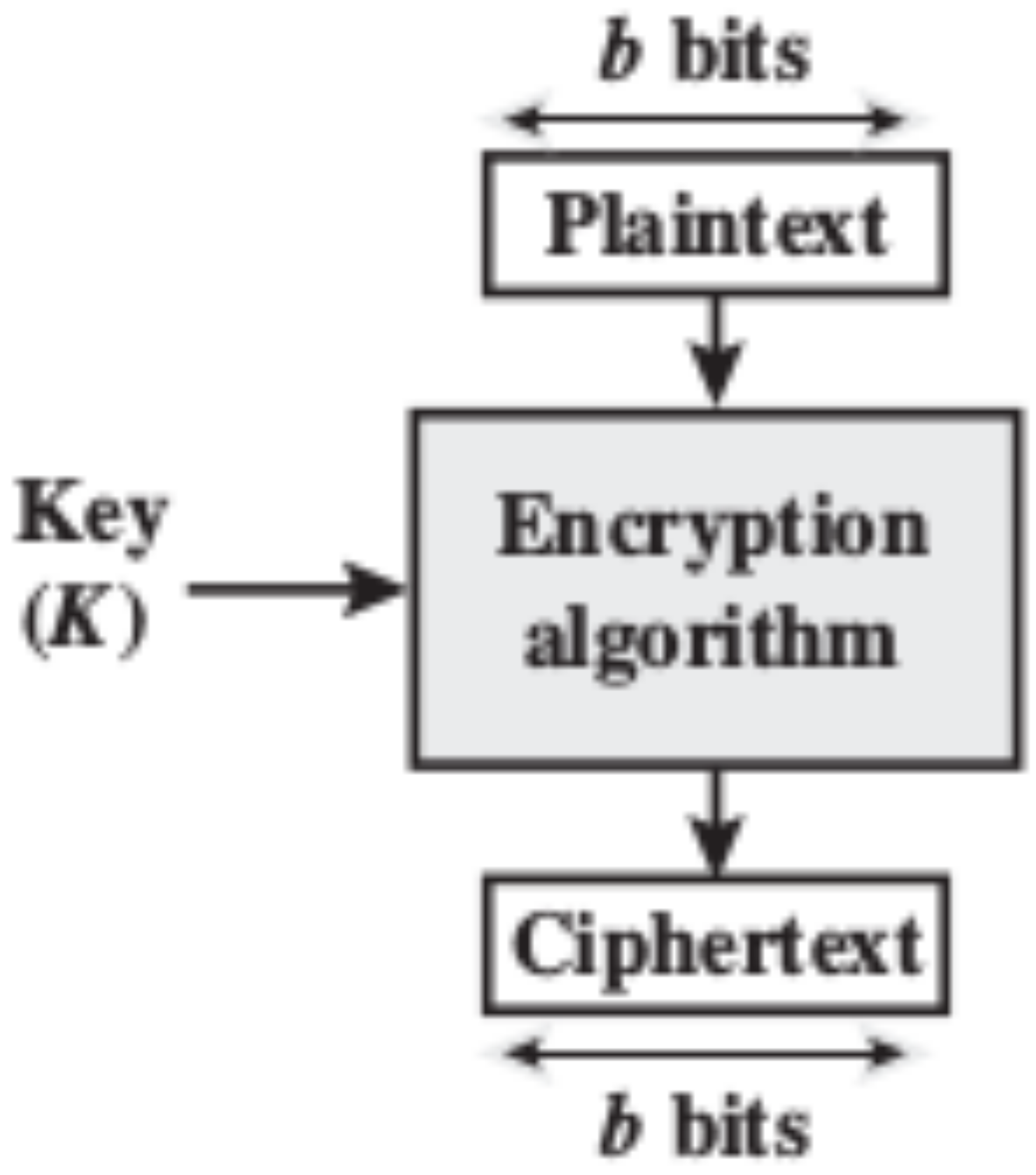
1. **Block cipher vs stream cipher**
2. **AES history**
3. **AES algorithm**
4. **Encryption modes**
5. **Implementation issues**
6. **Discussion**

Block cipher vs stream cipher

Stream encryption and Block encryption

Is it possible to determine which approach is the most secure?





Source (11.10.22) https://crypto.stackexchange.com/questions/33100/does-a-stream-cipher-provide-perfect-secrecy

Source (11.10.22): https://techblogmu.blogspot.com/2018/05/what-are-block-ciphers-explain-with-examples-the-ecb-and-cbc-modes-of-block-ciphers.html

AES history

AES as successor to DES algorithm

NIST competition

In 2001, the NIST calls for a successor to the DES algorithm.

Five algorithms were proposed MARS, Rivest Cipher 6, Serpent, Twofish and Rijndael.

NIST selected AES after an analysis that took into account factors such as security, performance, flexibility and implementation approaches.

A summary was published in Federal Information Processing Standards 197.

AES vs Rijndael

When Rijndael became the AES?

Strictly speaking, the Advanced Encryption Standard (AES) is not an algorithm. **AES is a specification** defined by the National Institute of Standards & Technology of the United States (NIST). AES is the successor of the Data Encryption Standard (DES).

AES has been announced in FIPS PUB 197 on November 26, 2001. Federal Information Processing Standards Publications (FIPS PUB) are issued by NIST after approval by the US Secretary of Commerce.

Rijndael is a symmetric key encryption algorithm created by Joan Daemen and Vincent Rijmen. It is a block cipher, with variable block size, variable key length & variable round number. Block length and key length can be independently specified to any multiple of 32 bits from 128 bits to 256 bits.

The Rijndael cipher as been selected as the Advanced Encryption Standard (AES).

In the Rijndael AES variant the block size is restricted to 128 bits and key length to 128, 192 or 256 bits only.

Source (15.10.22): https://stackoverflow.com/questions/748622/differences-between-rijndael-and-aes

Note on naming

1. Introduction

After the selection of Rijndael as the AES, it was decided to change the names of some of its component functions in order to improve the readability of the standard. However, we see that many recent publications on Rijndael and the AES still use the old names, mainly because the original submission documents using the old names, are still available on the Internet. In this note we repeat quickly the new names for the component functions. Additionally, we remind the reader on the difference between AES and Rijndael and present an overview of the most important references for Rijndael and the AES.

AES vs Rijndael

When Rijndael became the AES?

3. Naming

The names of the component functions of Rijndael have been modified between the publication of [2] and that of [3]. Table 1 lists the two versions of names. **We recommend using the new names**.

Old naming	New naming
`ByteSub`	`SubBytes`
`ShiftRow`	`ShiftRows`
`MixColumn`	`MixColumns`
`AddRoundKey`	`AddRoundKey`

Table 1: Old and new names of the Rijndael component functions

Source (10.10.22): https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf

AES vs Rijndael

When Rijndael became the AES?

4. Range of key and block lengths in Rijndael and AES

Rijndael and AES differ only in the range of supported values for the block length and cipher key length.

For Rijndael, the block length and the key length can be independently specified to any multiple of 32 bits, with a minimum of 128 bits, and a maximum of 256 bits. The support for block and key lengths 160 and 224 bits was introduced in reference [2].

AES fixes the block length to 128 bits, and supports key lengths of 128, 192 or 256 bits only.

Date: 9/04/2003 Page: 1/2

Source (10.10.22): https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf

AES vs Rijndael

When Rijndael became the AES?

5. Referencing

Reference [3] is the US Federal Information Processing Standard defining AES and hence the **definitive reference on AES**.

Reference [4] is the definitive reference on Rijndael. It is a book we have written after the selection of Rijndael as AES and was published in February 2002. It describes all aspects of Rijndael and is only available on paper.

Reference [1] is the original Rijndael documentation submitted to AES and dates from June 11, 1998. Reference [2] is an improved version dating from September 3, 1999 that supersedes reference [1]. Both were made available electronically in PDF formats on several sites. Both references should be used only when referring to the actual historical documents. Technical or scientific references should be restricted to [3] and [4].

[1] Joan Daemen and Vincent Rijmen, AES submission document on Rijndael, June 1998.

[2] Joan Daemen and Vincent Rijmen, AES submission document on Rijndael, Version 2, September 1999. http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael.pdf

[3] FIPS PUB 197, Advanced Encryption Standard (AES), National Institute of Standards and Technology, U.S. Department of Commerce, November 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[4] Joan Daemen and Vincent Rijmen, The Design of Rijndael, AES - The Advanced Encryption Standard, Springer-Verlag 2002 (238 pp.)

Source (10.10.22): https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf

AES algorithm

Advanced Encryption Standard

FIPS Publication

Federal Information

Processing Standards Publication 197

November 26, 2001

Announcing the

ADVANCED ENCRYPTION STANDARD (AES)

Source (9.10.22): http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

Brute force attack

Key length and algorithm

Key Size	Possible Combinations
1 bit	2
2 bits	4
4 bits	16
8 bits	256
16 bits	65536
32 bits	4.2×10^{9}
56 bits (DES)	7.2×10^{16}
64 bits	1.8×10^{19}
128 bits (AES)	3.4×10^{38}
192 bits (AES)	6.2×10^{57}
256 bits (AES)	1.1×10^{77}

Table 1. Key sizes and corresponding possible combinations to crack by brute force attack.

Source (11.10.22): https://www.atpinc.com/blog/what-is-aes-256-encryption

DESIGNLINES | INTERNET OF THINGS DESIGNLINE

How Secure is AES 128 and 256 Encryption Against Brute Force Attacks?

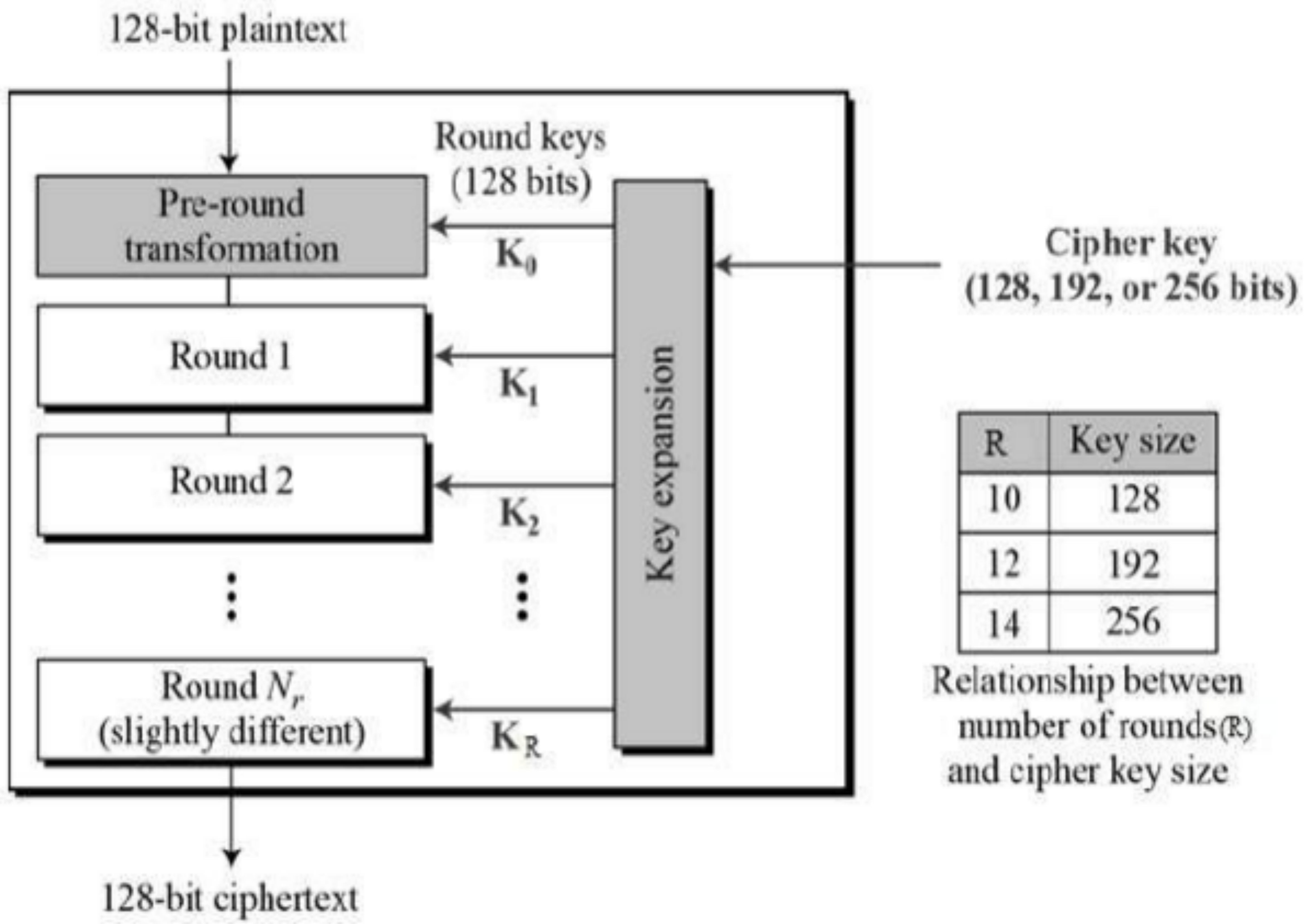
By Mohit Arora, Sr. Systems Engineer & Security Architect, Freescale Semiconductor 05.07.2012

Source (11.10.22) https://www.eetimes.com/how-secure-is-aes-against-brute-force-attacks/

Discussion with students about the article from https://www.eetimes.com/ as additional teaching content.

AES

What the algorithm contains?



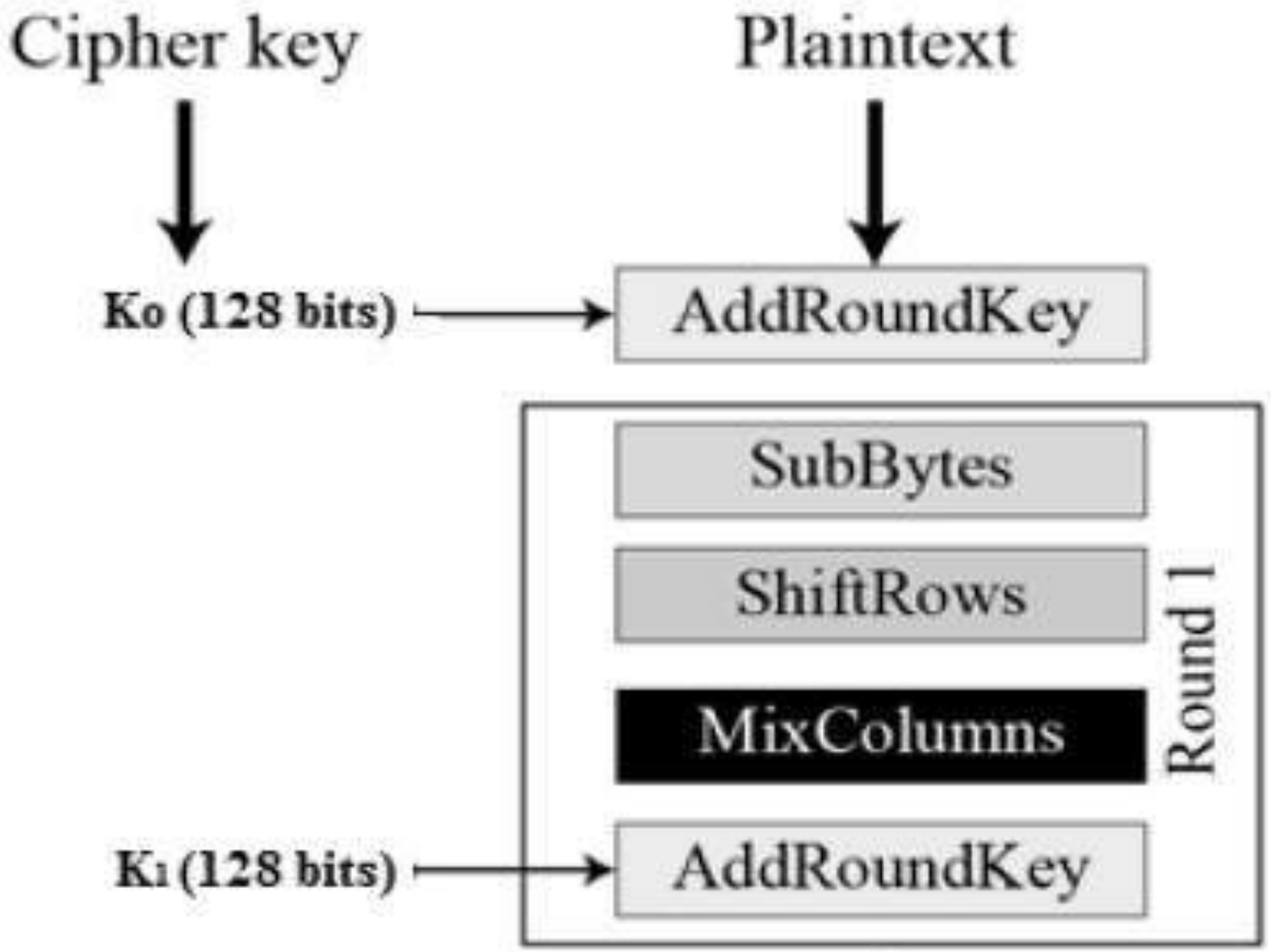
Key length vs. number of rounds:

- 128 bits = 9 rounds
- 192 bits = 11 rounds
- 256 bits = 13 rounds

Source (11.10.22): http://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm

AES

What the algorithm contains?



SubBytes

Transformation a block of data using S-Box.

ShiftRows

Bytes are shifted according to block sizes.

MixColumns

Each column is multiplied by the matrix. The bytes being multiplied are considered as polynomials, not as numbers. When results have more than 8 bits, the extra bits are cancelled out by XORing the binary 9-bit string 100011011 with the result.

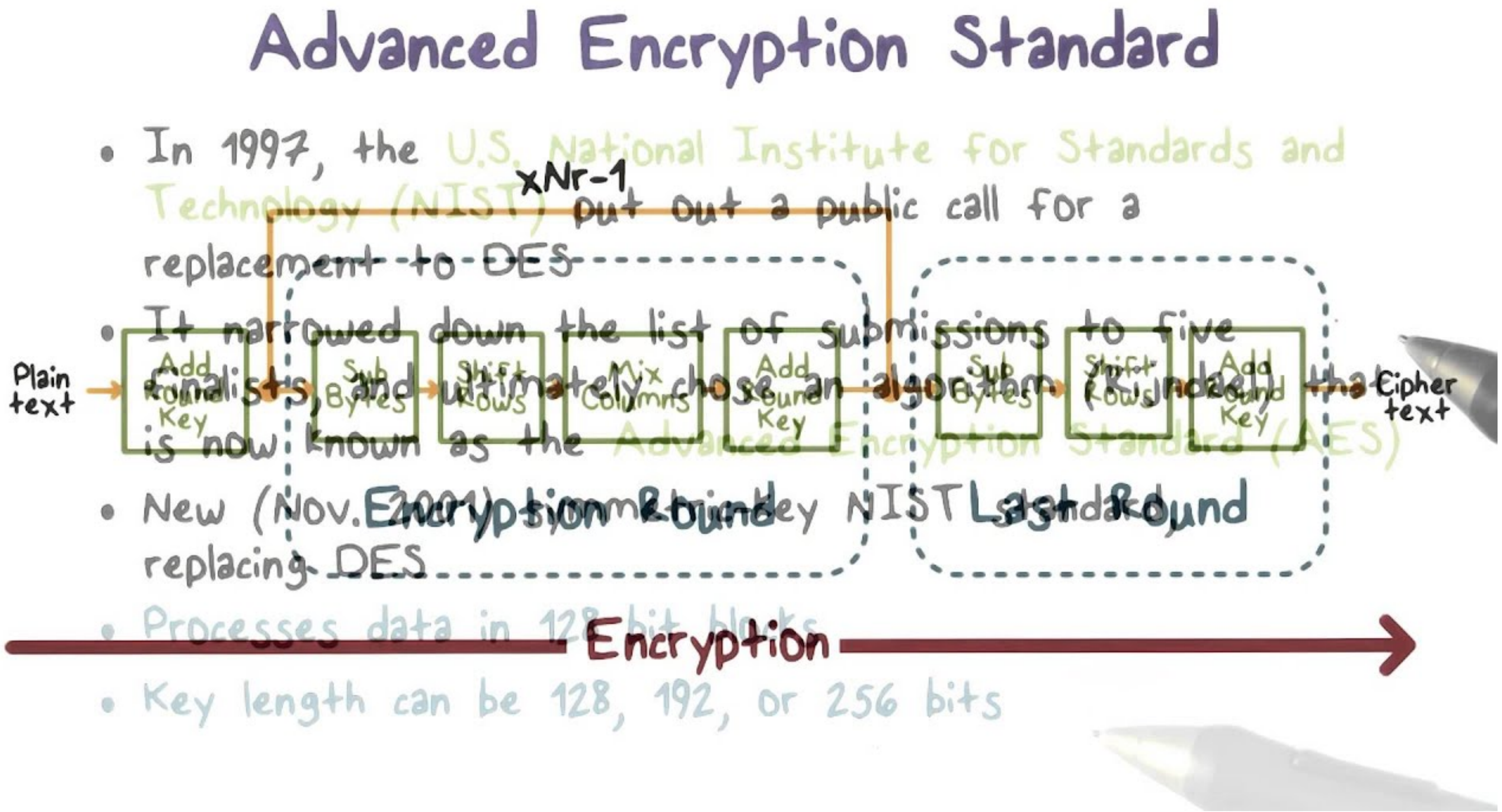
AddRoundKey

Opration with a new key for the round

Source (11.10.22): http://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm

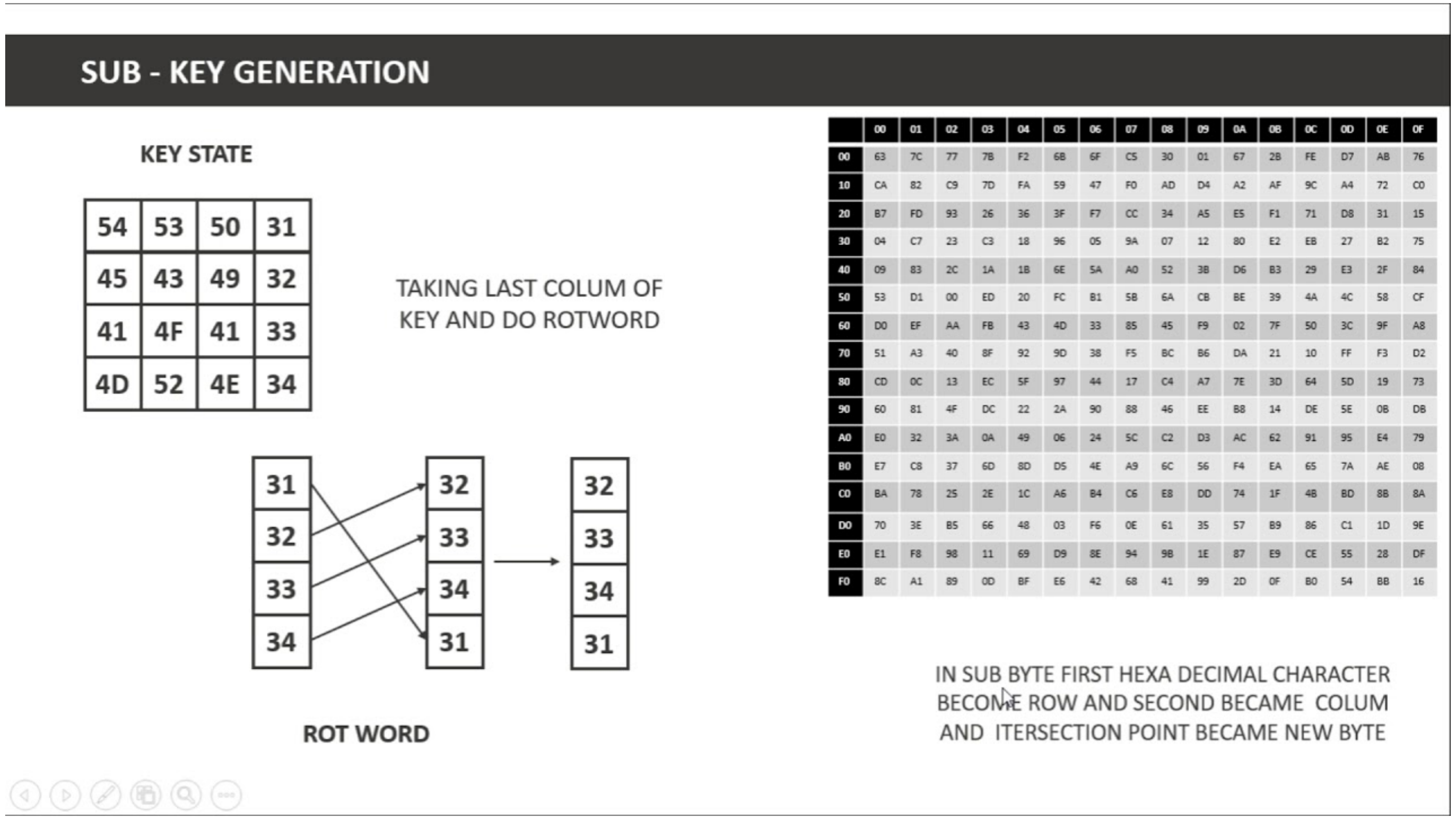
AES

External support material from YouTube for teaching uses

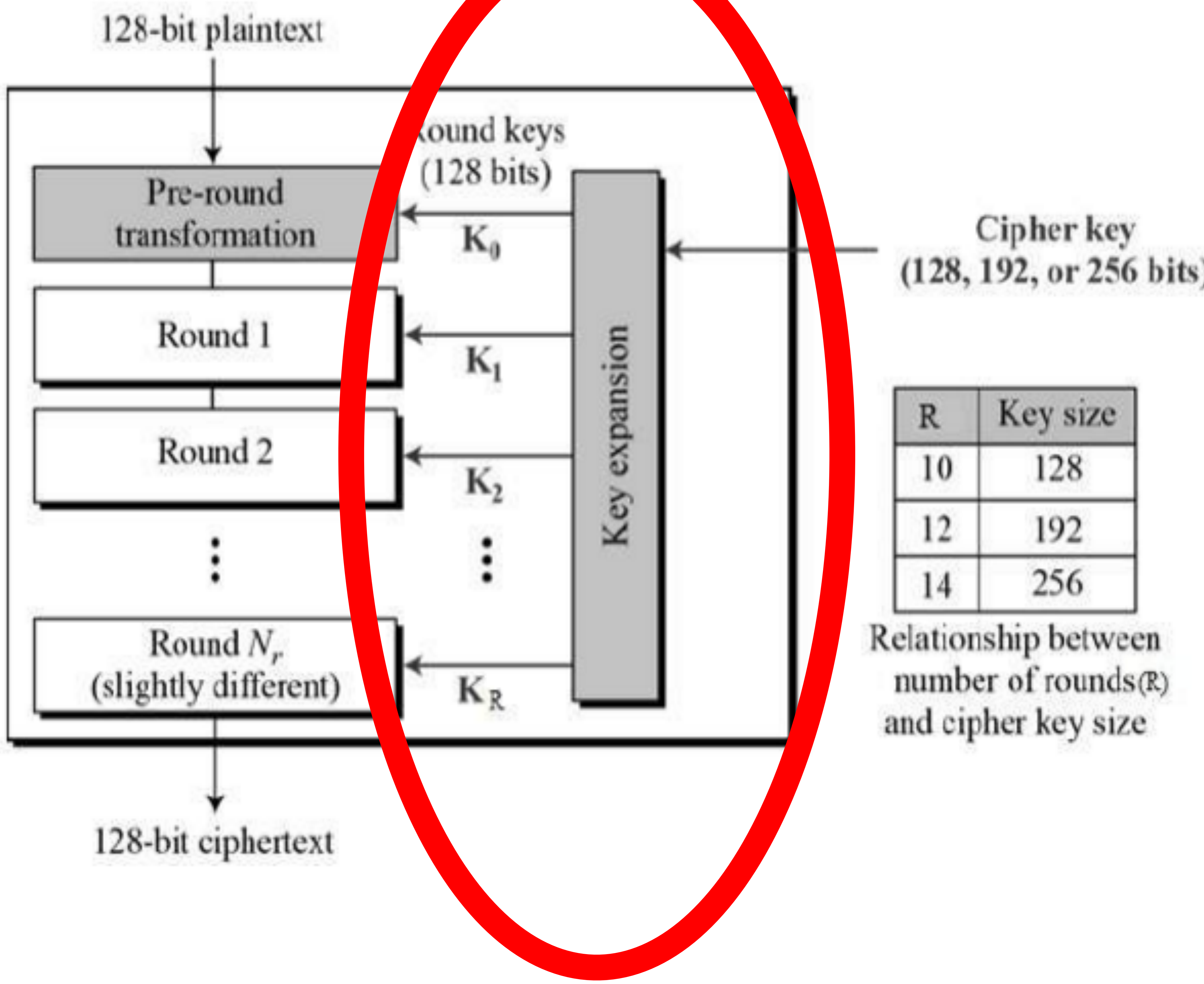


AES: subkey generation

External support material from YouTube for teaching uses



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



R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

AES: subBytes

External support material from YouTube for teaching uses

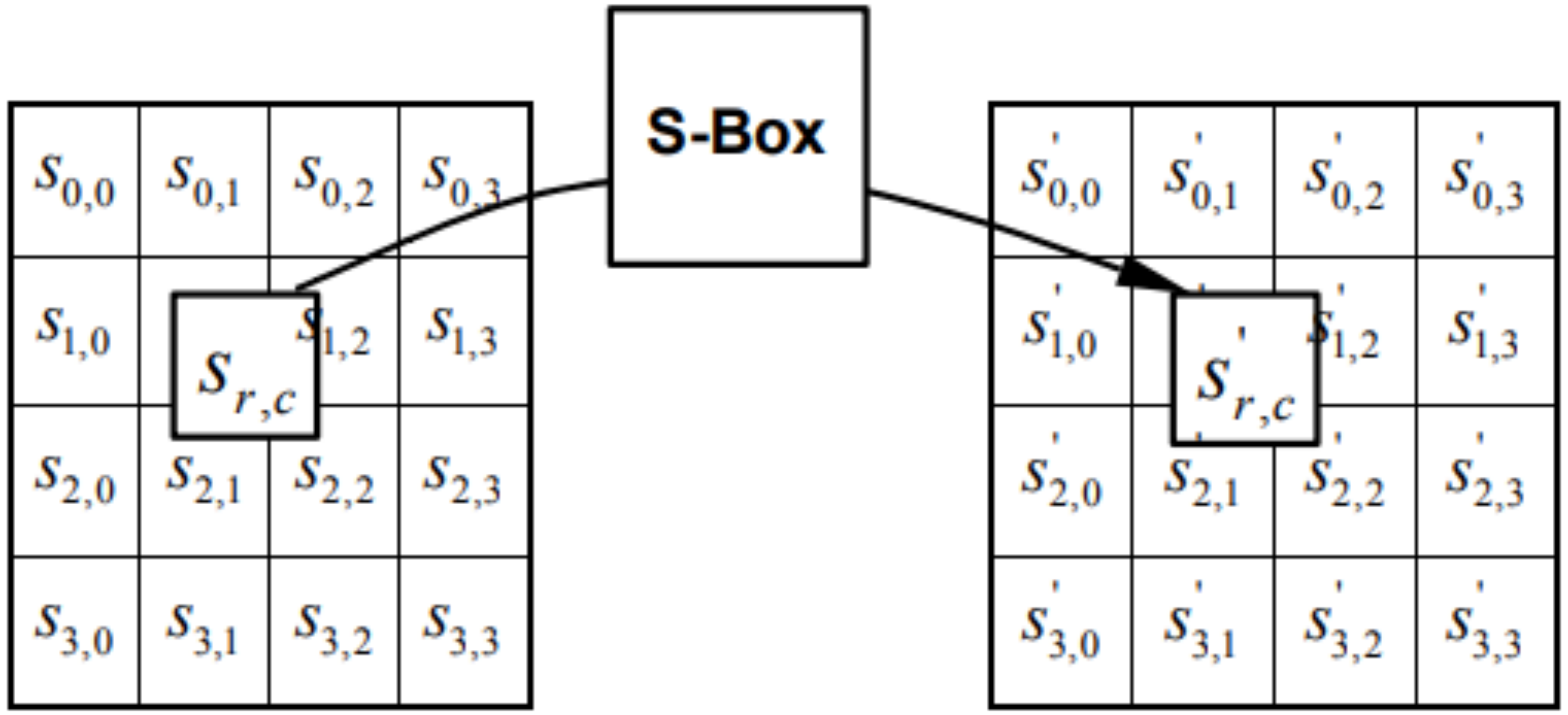
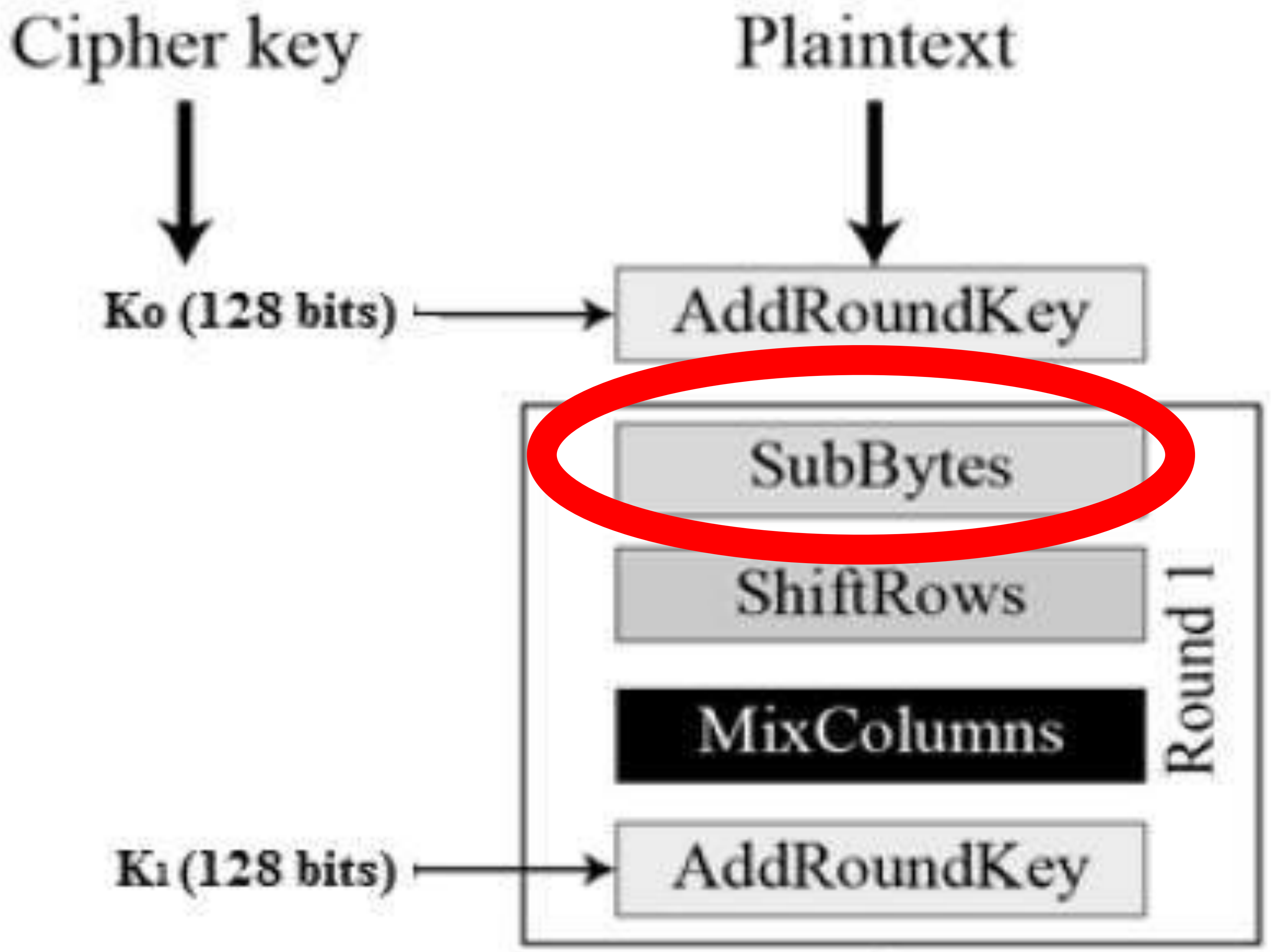


Figure 6. `SubBytes()` applies the S-box to each byte of the State.



Source (11.10.22): http://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm

AES: subBytes, S-BOX

External support material from YouTube for teaching uses

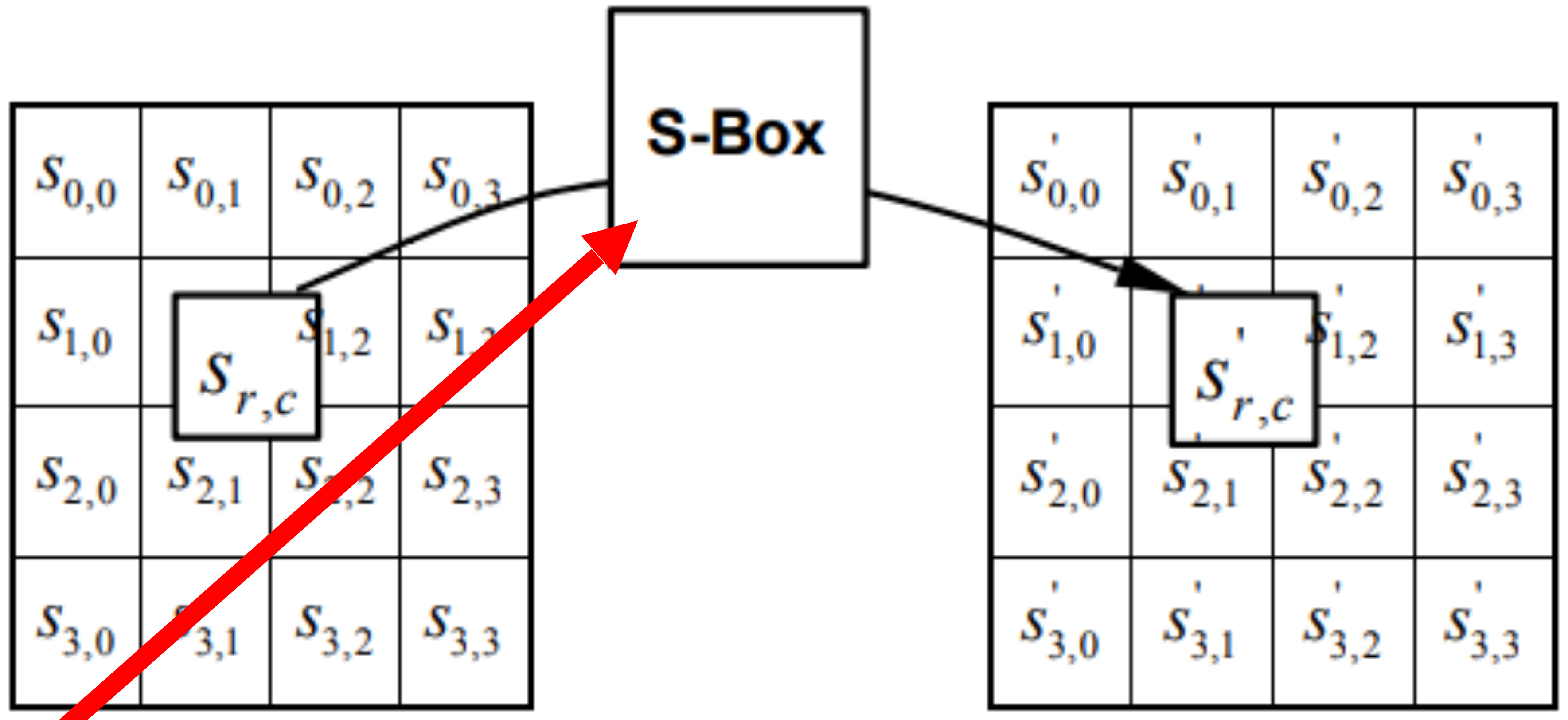


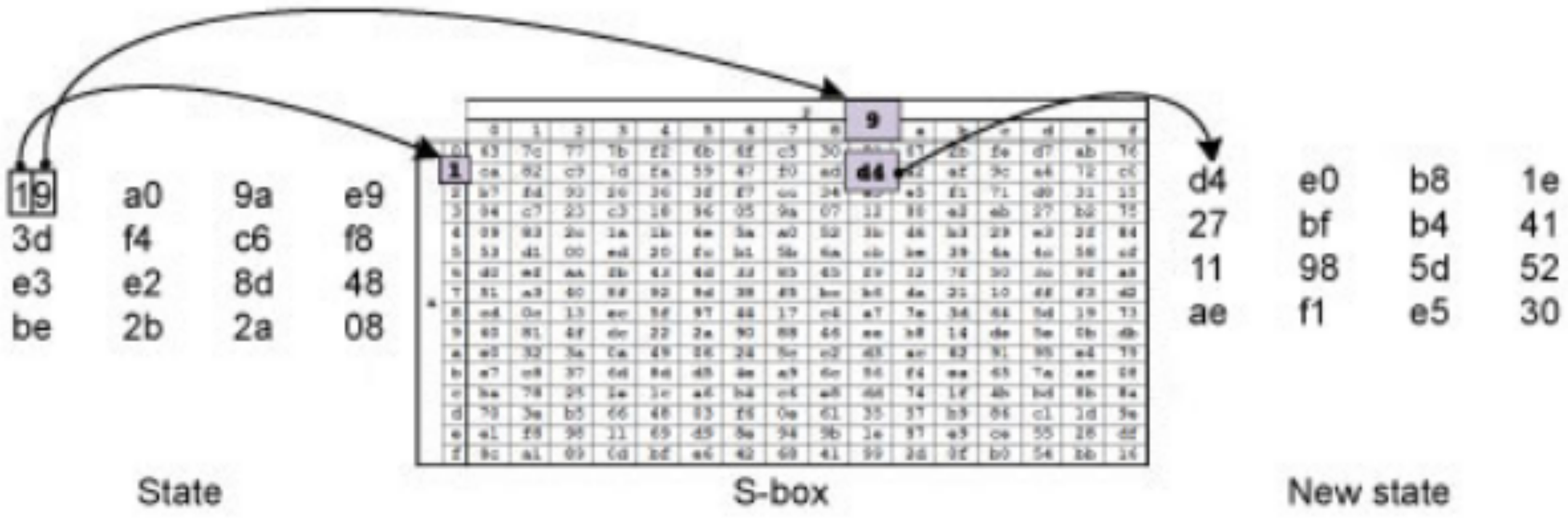
Figure 6. `SubBytes()` applies the S-box to each byte of the Sta

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

Figure 7. S-box: substitution values for the byte xy (in hexadecimal format).

AES: subBytes, S-BOX

Example of transformation



AES: ShiftRows

Transformation on current matrix values

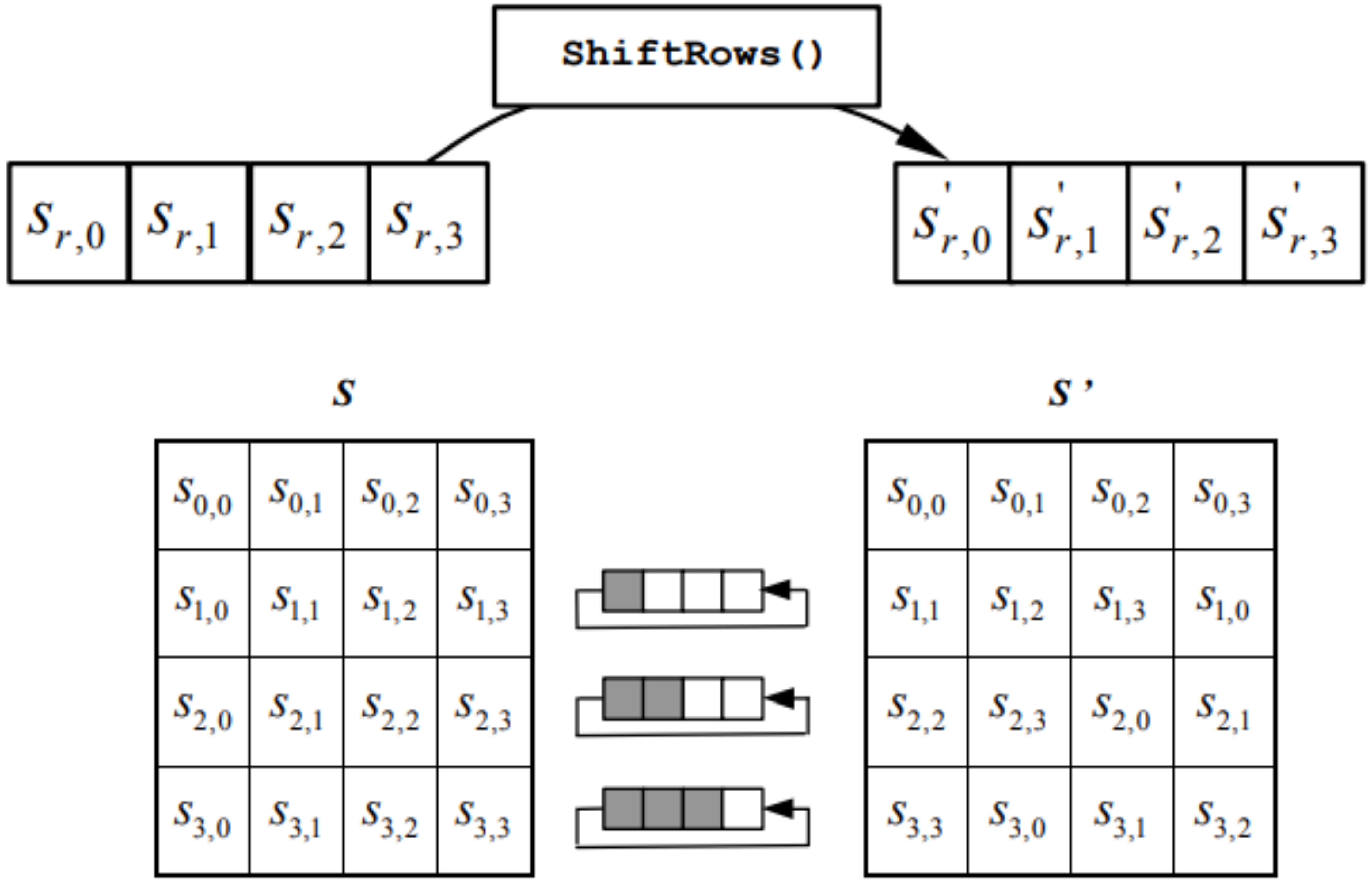
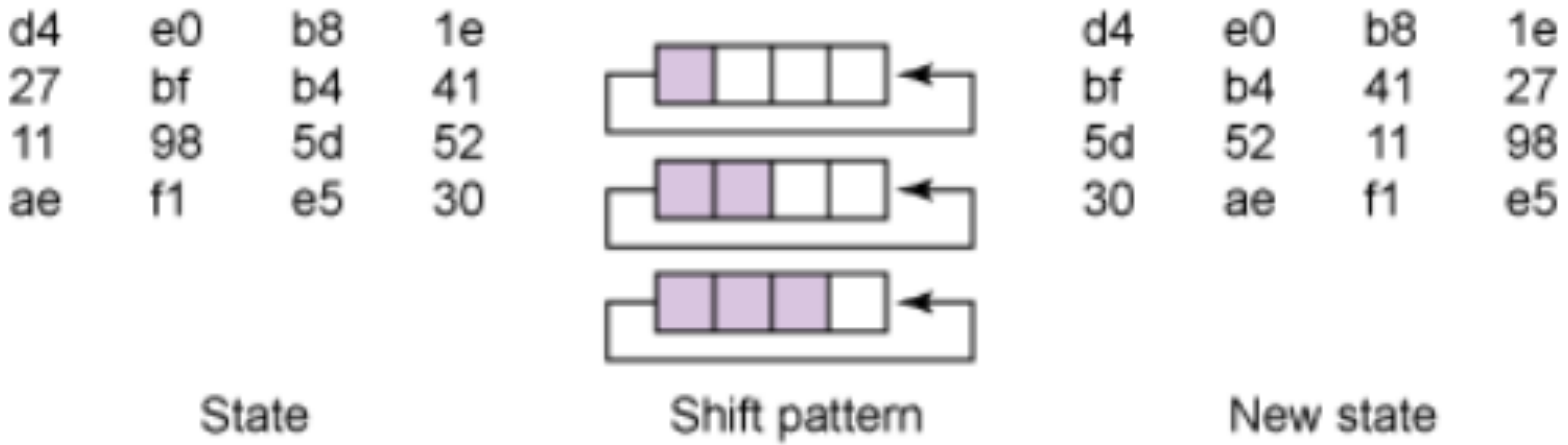


Figure 8. `ShiftRows()` cyclically shifts the last three rows in the State.

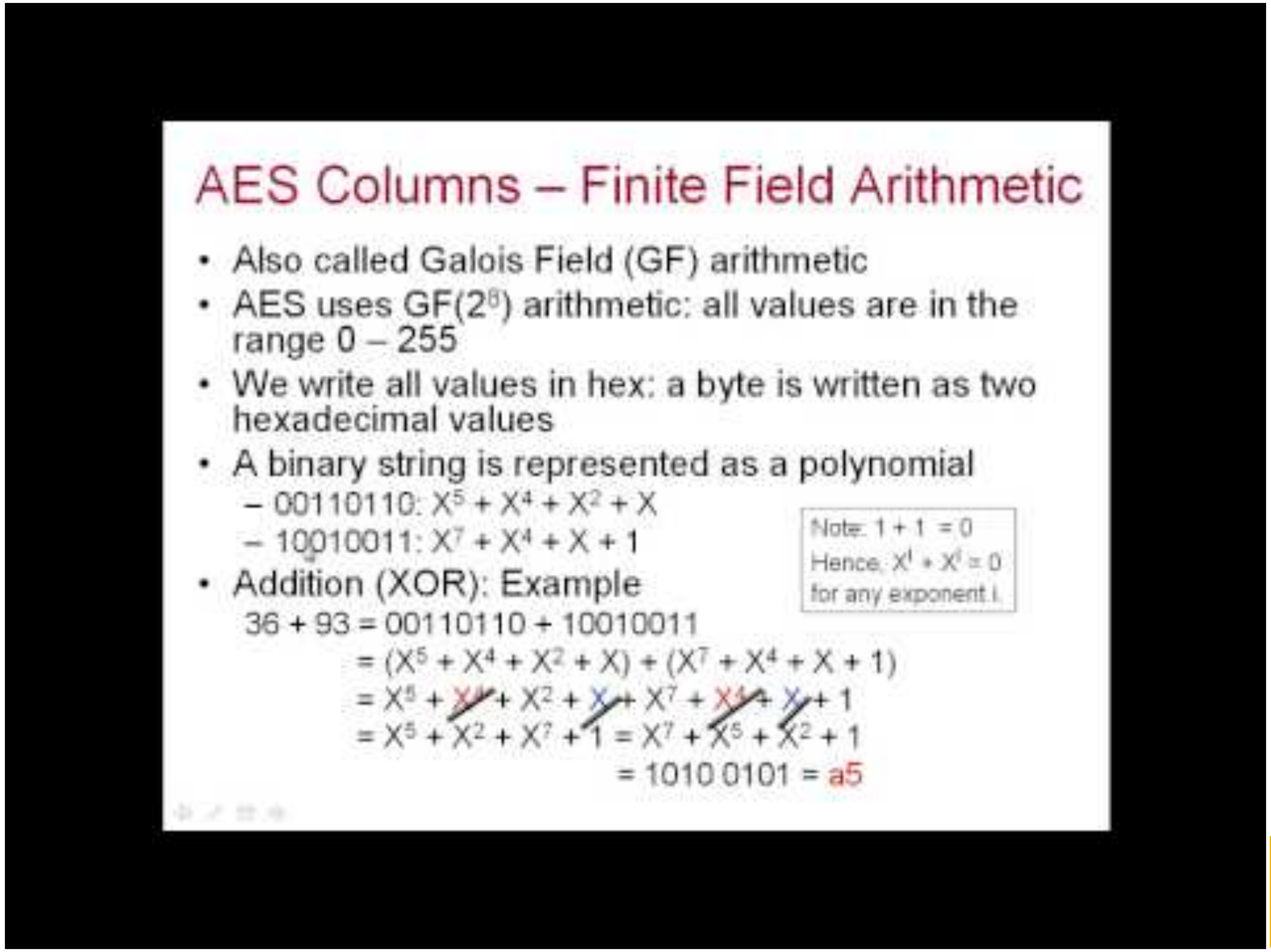
AES: subBytes, S-BOX

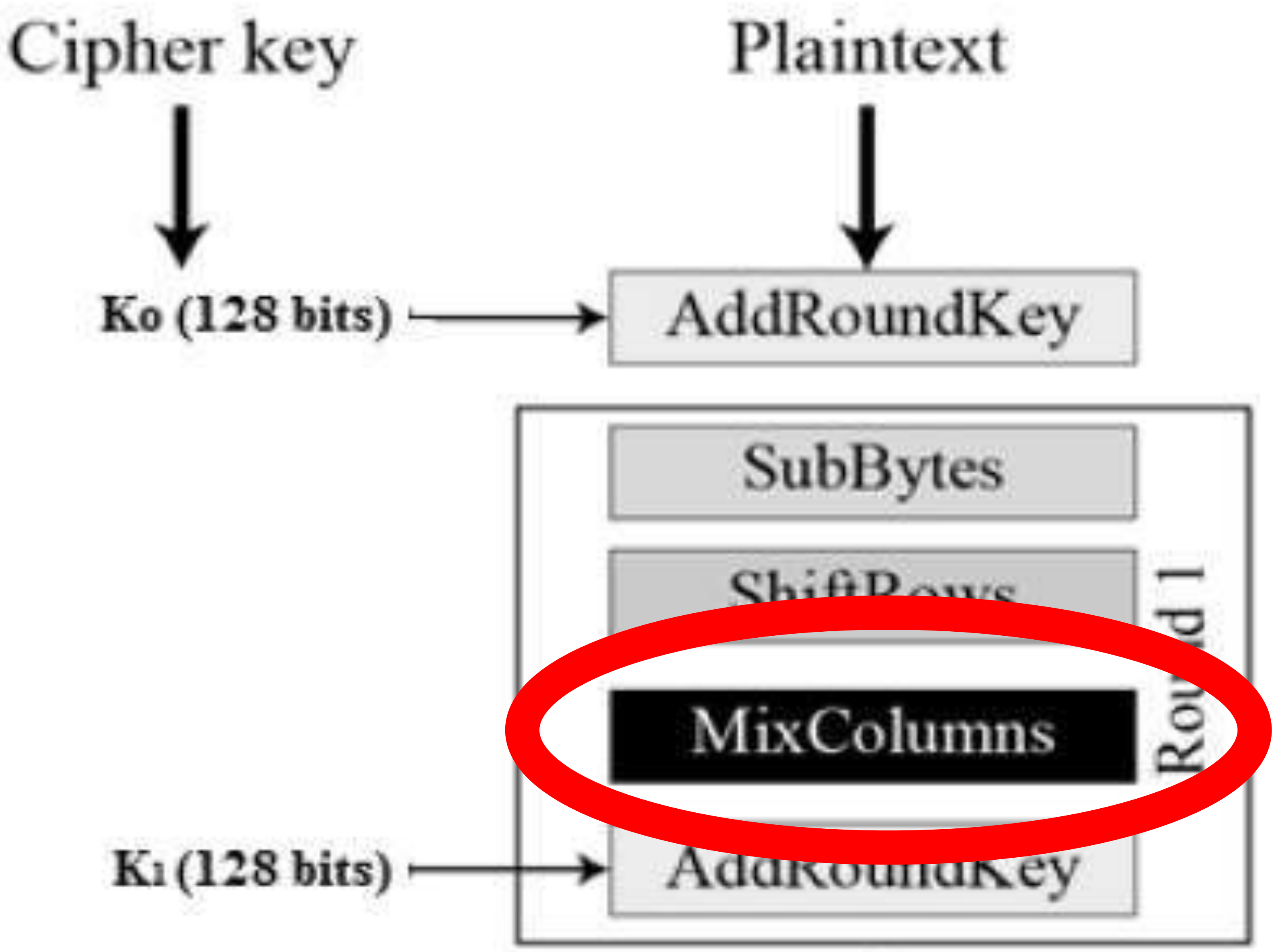
Example of transformation



AES: MixColumns

External support material from YouTube for teaching uses





key content on YouTube video: from 09:32 and 28:30

Encryption modes

Recommendations for encryption modes

NIST Special Publication 800-38A
2001 Edition

Recommendation for Block Cipher Modes of Operation

Methods and Techniques

Morris Dworkin

NIST
National Institute of Standards and Technology
Technology Administration
U.S. Department of Commerce

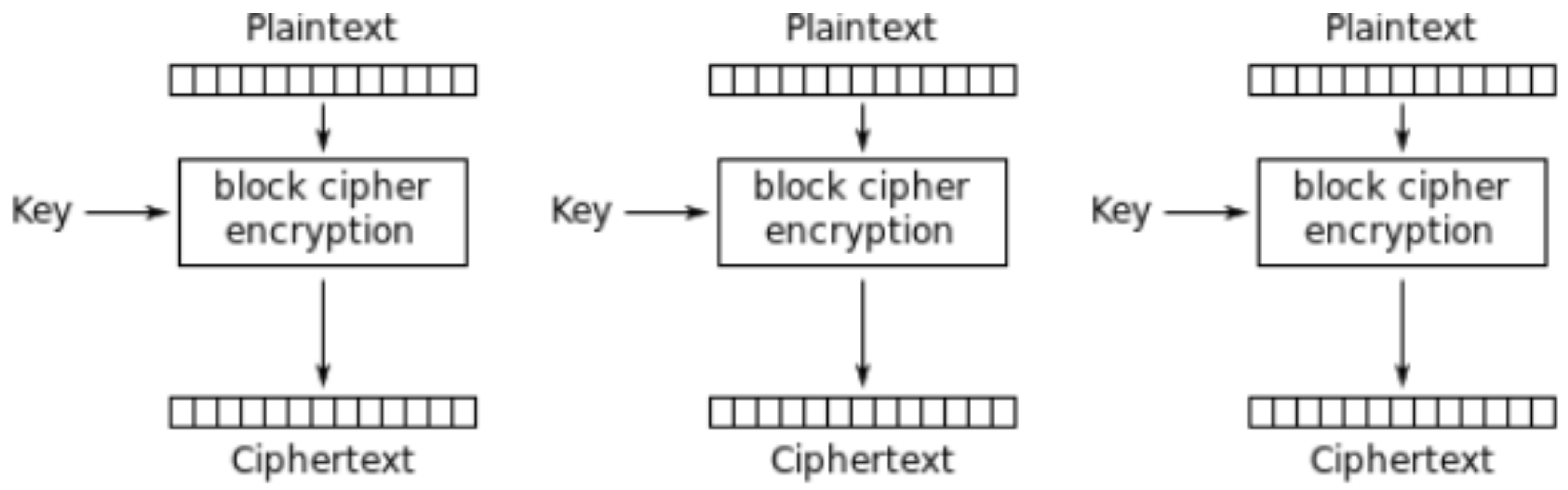
C O M P U T E R   S E C U R I T Y

Discussion with students

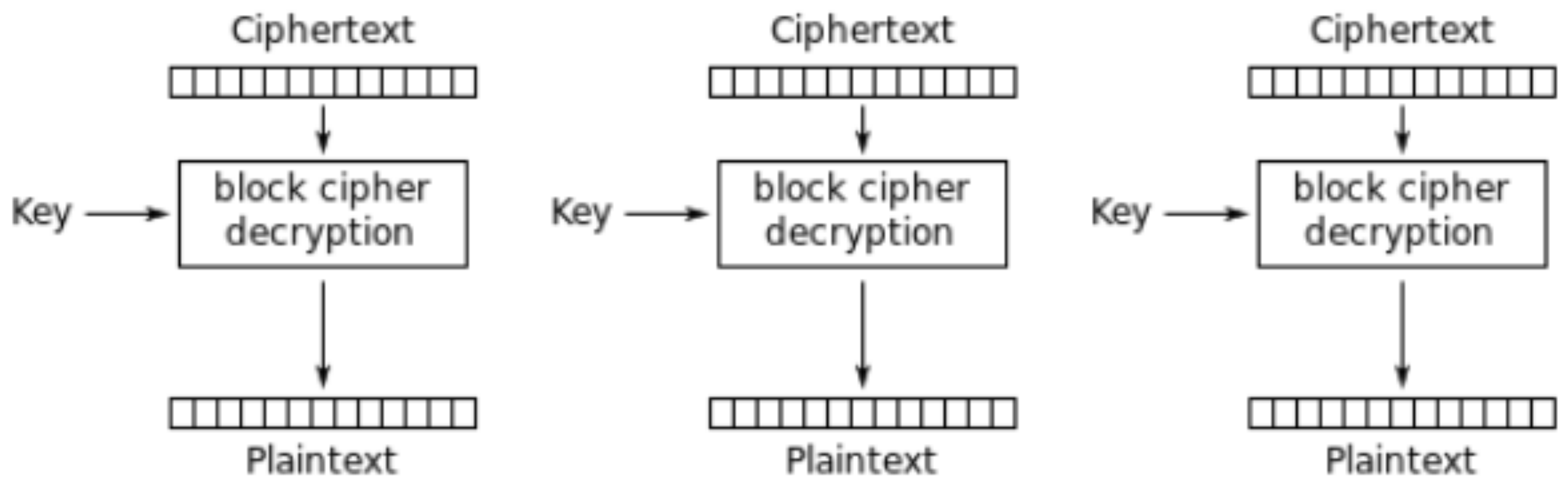
Why encryption modes are important for the security of using the AES algorithm?

Encryption modes

ECB - the weakest mode



Electronic Codebook (ECB) mode encryption

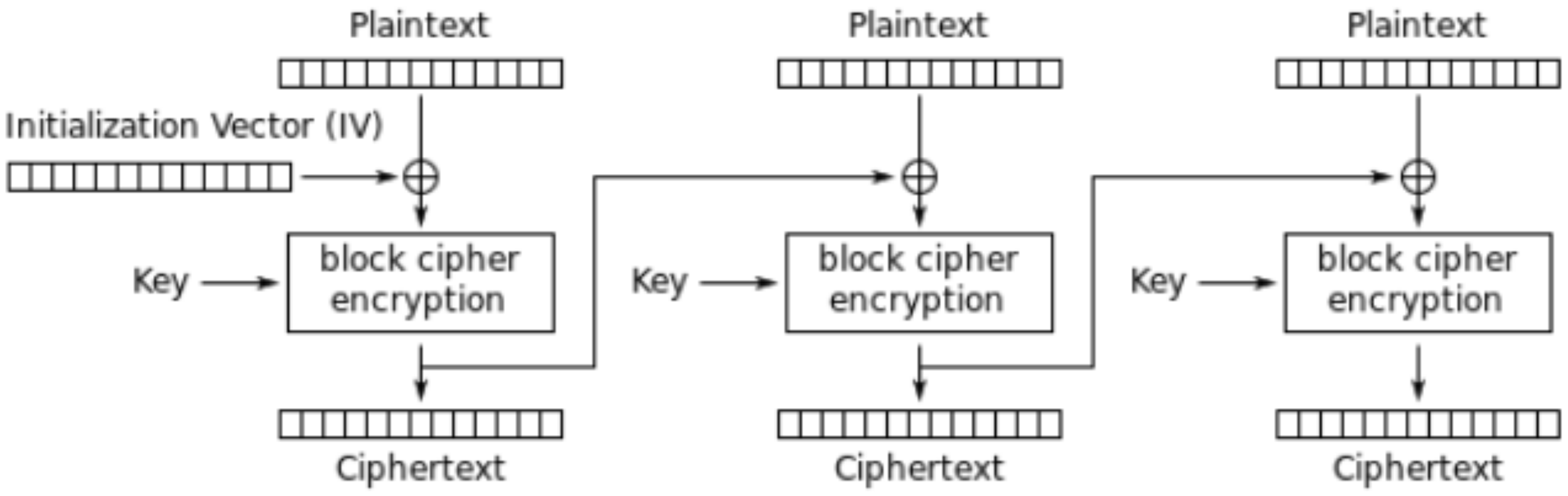


Electronic Codebook (ECB) mode decryption

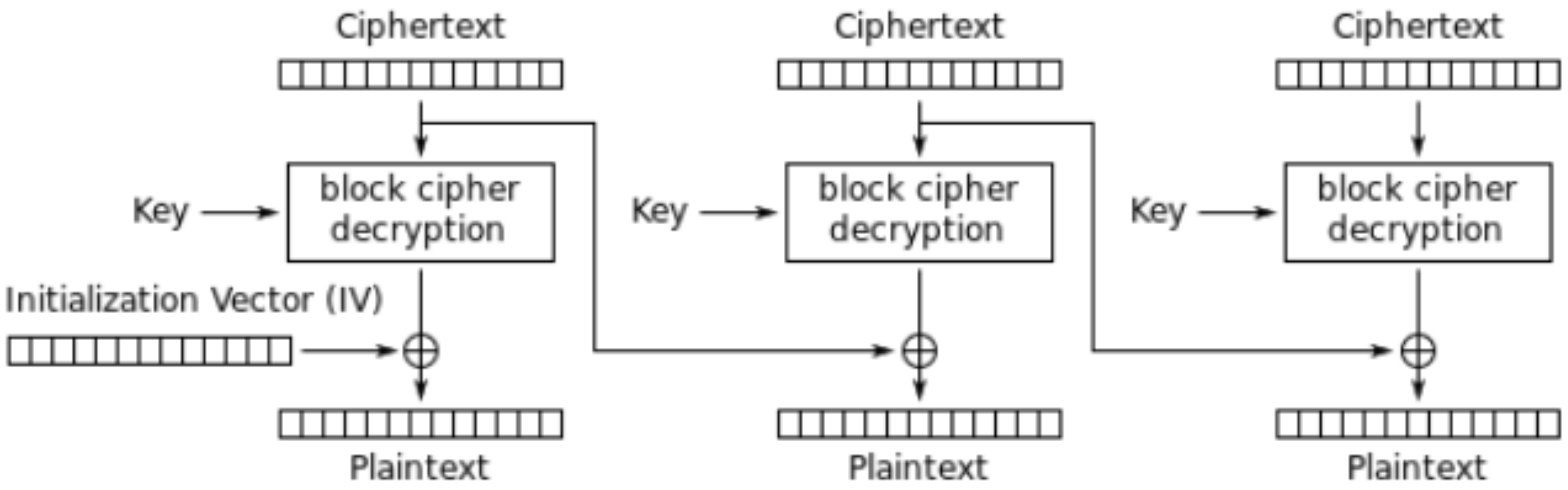
Source (17.10.22): https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

Encryption modes

CBC



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

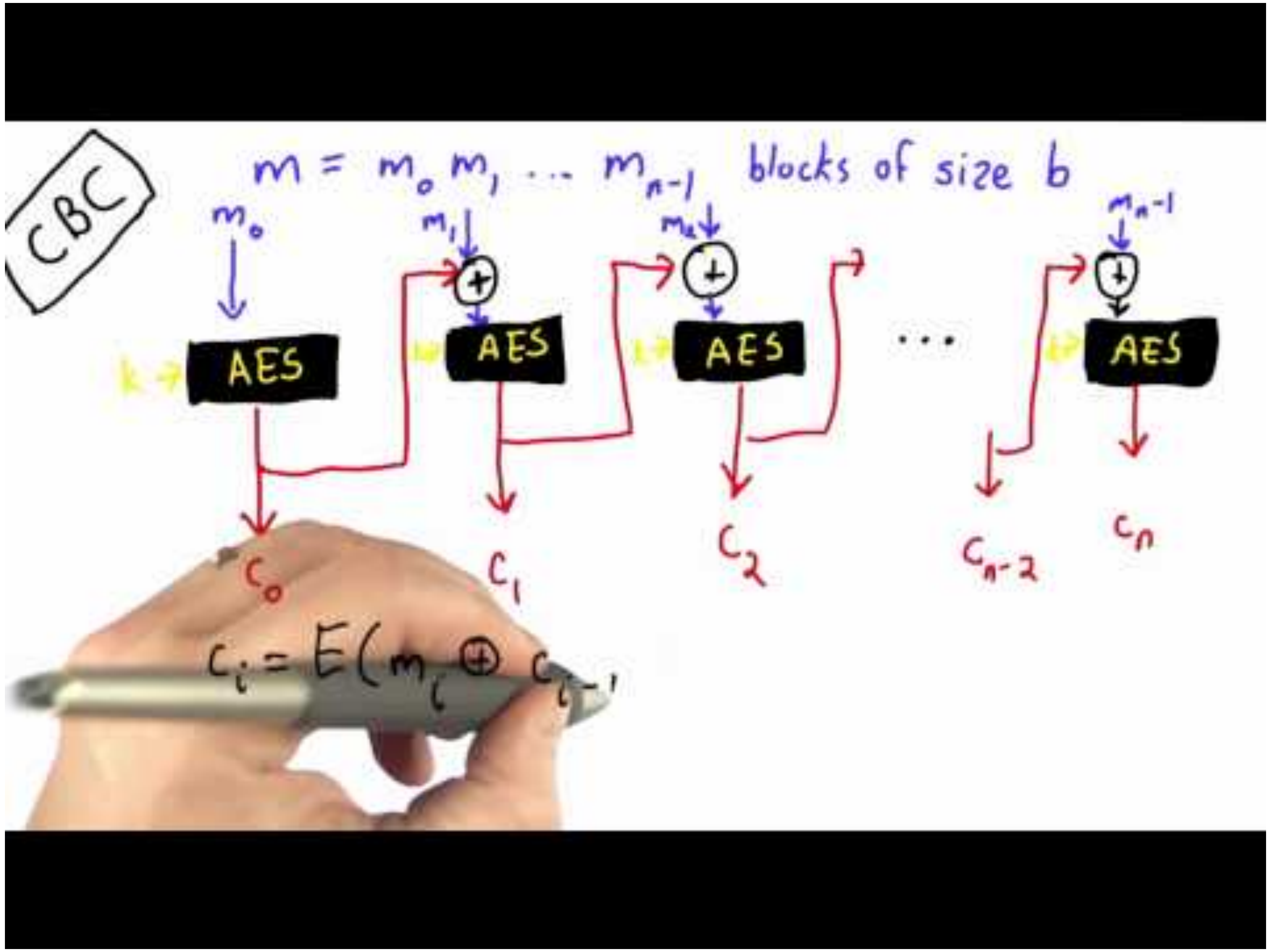
Source (17.10.22): https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

<u>Discussion with students</u>

What is the role of the IV?

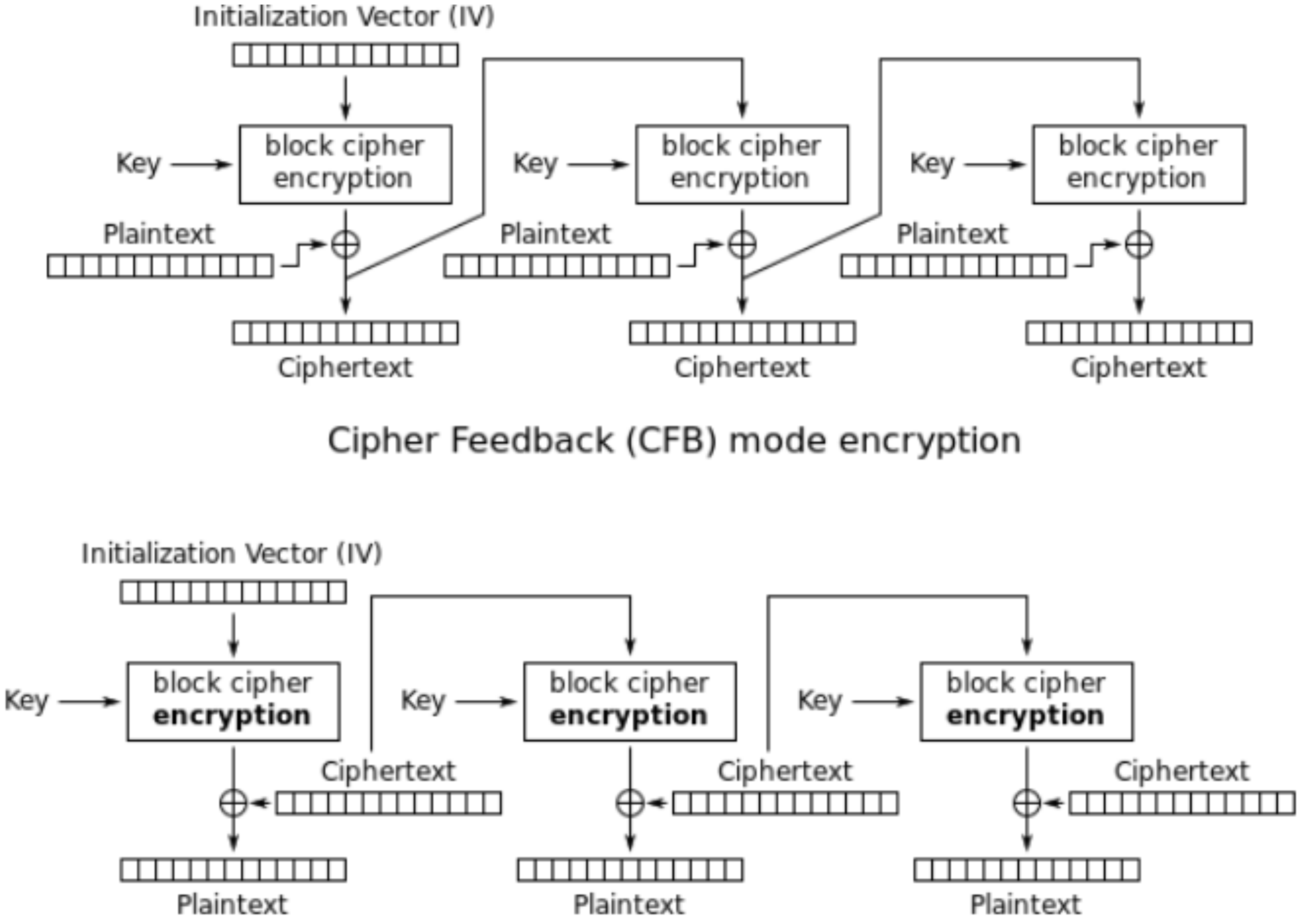
Encryption modes

External support material from YouTube for teaching uses



Encryption modes

CFB



Cipher Feedback (CFB) mode encryption

Cipher Feedback (CFB) mode decryption

Source (17.10.22): https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

Encryption modes

IV - initialization vector

Each mode of operation has its own IV requirements. Some need uniform, unpredictable randomness. Other are equally happy with just uniqueness.

CBC is well-known for its need of an IV chosen randomly and uniformly among the possible IV values, and such that an attacker who can choose the text to encrypt may not predict the IV value before submitting the said text.

PCBC is similar to CBC in that respect. When encrypting, PCBC is equivalent to applying a linear transformation on the input data, followed by CBC encryption. So PCBC also needs a uniform unpredictable IV.

CFB and OFB require only uniqueness: for a given key, each IV value shall be used at most once. The is no need for unpredictability or uniformness because the IV is first encrypted "as is" (before any operation with the plaintext) and encryption of a sequence of values with a good block cipher, using a key that the attacker does not know, *is* a good PRNG. This means that CFB and OFB somehow include what it takes to elevate a unique IV to appropriate uniform randomness.

Source(29.11.22): https://crypto.stackexchange.com/questions/1312/using-a-non-random-iv-with-modes-other-than-cbc

Implementation issues

AES: reference implementation

External support material from YouTube for teaching uses

Appendix A - Key Expansion Examples

This appendix shows the development of the key schedule for various key sizes. Note that mu byte values are presented using the notation described in Sec. 3. The intermediate values produced during the development of the key schedule (see Sec. 5.2) are given in the following table (all values are in hexadecimal format, with the exception of the index column (i)).

A.1 Expansion of a 128-bit Cipher Key

This section contains the key expansion of the following cipher key:

```
Cipher Key = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c
```

for $Nk = 4$, which results in

$w_0 =$ 2b7e1516 $w_1 =$ 28aed2a6 $w_2 =$ abf71588 $w_3 =$ 09cf4f3c

i (dec)	temp	After RotWord()	After SubWord()	Rcon[i/Nk]	After XOR with Rcon	w[i-Nk]	w[i]= temp XOR w[i-Nk]
4	09cf4f3c	cf4f3c09	8a84eb01	01000000	8b84eb01	2b7e1516	a0fafe17
5	a0fafe17					28aed2a6	88542cb1
6	88542cb1					abf71588	23a33939
7	23a33939					09cf4f3c	2a6c7605
8	2a6c7605	6c76052a	50386be5	02000000	52386be5	a0fafe17	f2c295f2
9	f2c295f2					88542cb1	7a96b943

AES - a bug-free implementation

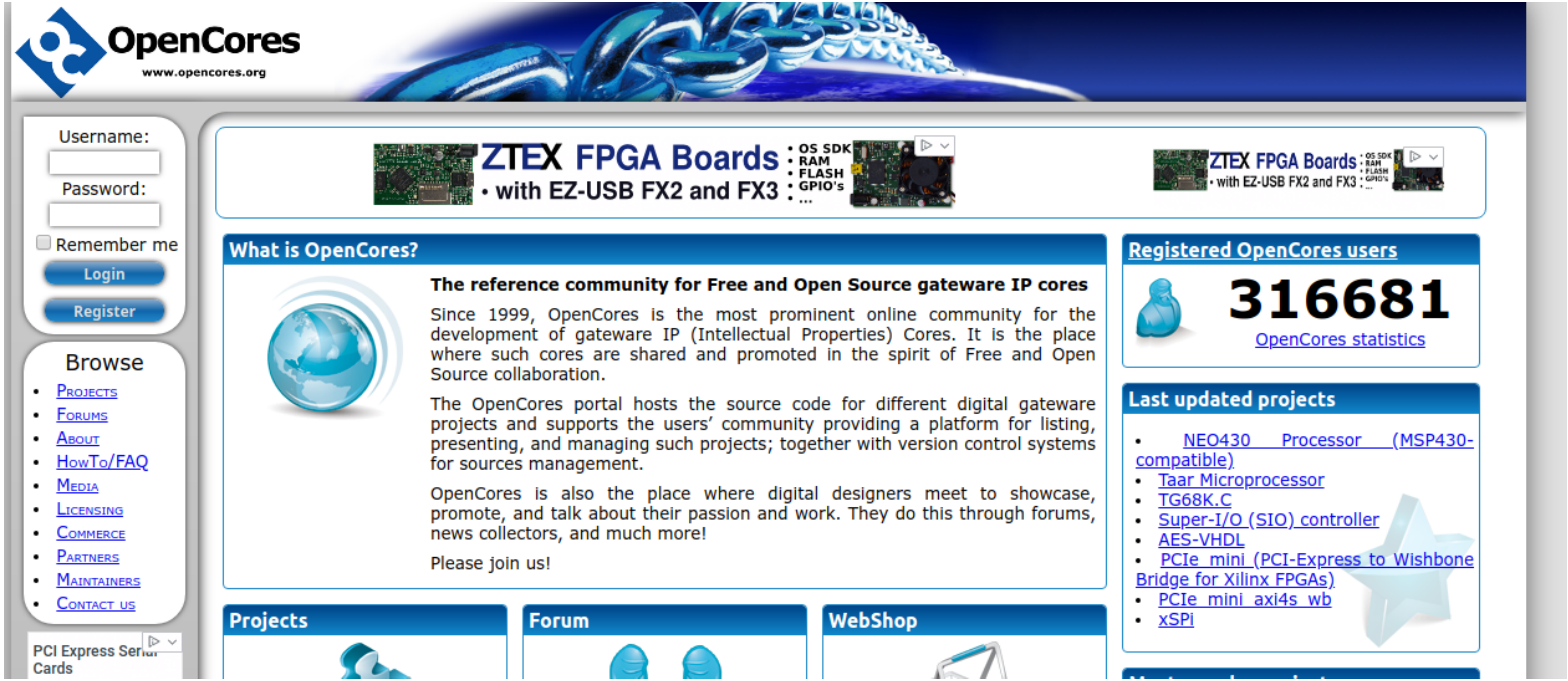
Independent approaches alone vs. industrial approaches

Discussion with students

Are student implementations of the AES algorithm secure?

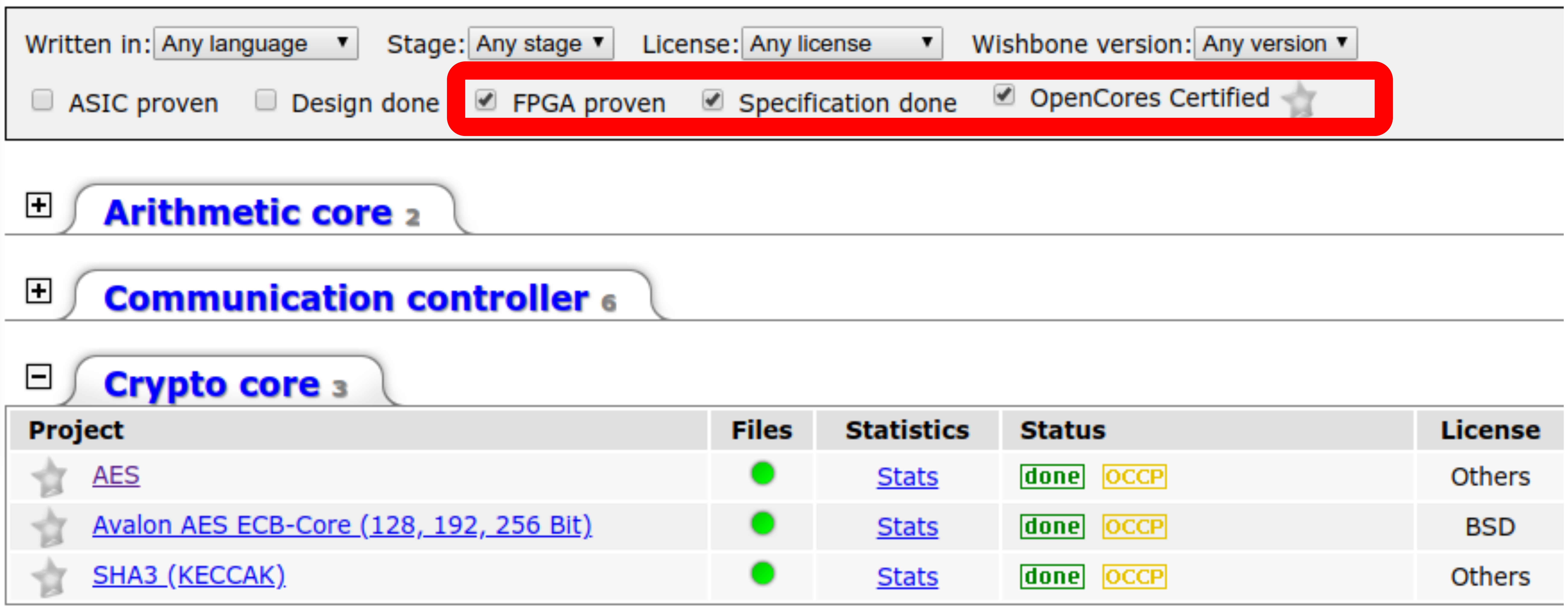
Cryptographic IPCores

Open Source Code



Cryptographic IPCores

Open Source Code



Project	Files	Statistics	Status	License
AES		Stats	done OCCP	Others
Avalon AES ECB-Core (128, 192, 256 Bit)		Stats	done OCCP	BSD
SHA3 (KECCAK)		Stats	done OCCP	Others

Cryptographic IPCores

Open Source Code

Crypto core 29

Project	Files	Statistics	Status	License
128/192 AES	●	Stats	wbc	
3DES (Triple DES) / DES (VHDL)	●	Stats		
AES	●	Stats	done OCCP	Others
AES Decryption Core for FPGA	●	Stats	done	LGPL
AES-128 Encryption	●	Stats	done	LGPL
Avalon AES ECB-Core (128, 192, 256 Bit)	●	Stats	done OCCP	BSD
Bluespec Cryptosorter	●	Stats		
Bluespec MD6	●	Stats		
Compact CLEFIA for FPGA	●	Stats	done	LGPL
Crypto-PAn	●	Stats	done	GPL
fast AES-128 Encryption only cores	●	Stats		Others
Flexible Design of a Modular Simultaneous Exponentiation Core	●	Stats		LGPL
Galois Counter Mode Advanced Encryption Standard GCM-AES	●	Stats		Others
GOST 28147-89	●	Stats		BSD
gost28147-89	●	Stats		BSD
high throughput and low area aes core	●	Stats		LGPL
IOTA PoW Pearl-Diver Curl-P81	●	Stats	done	Others
MD5 Pipelined	●	Stats	done	LGPL
Montgomery modular multiplier and exponentiator	●	Stats	done	LGPL
Present - a lightweight block cipher	●	Stats	done	LGPL
RC4 Pseudo-random stream generator	●	Stats	done	LGPL
rc6 cryptography	●	Stats		GPL

Cryptographic IPCores

Commercial

AES Core G2

Price: $5,000.00

add to cart

Easy to use Advanced Encryption Standard (AES) Core proving privacy modes. Product has NIST validation certificate and 32 bit internal datapath width. more info

AES Core G3

Price: $10,000.00

add to cart

Flexible Advanced Encryption Standard (AES) Core providing privacy modes. Product has a NIST validation certificate and parameterisable internal datapath width to allow a wide range of performance/area tradeoffs. more info

AES Keywrap Core

Price: $12,000.00

add to cart

The AES Keywrap algorithm (IETF RFC 3394) is used to protect cryptographic keys in transit, it is listed as an approved Key Establishment Technique in FIPS 140-2. This implementation is based on our G3 AES core configured with a 32 bit data path width. more info

AES Core CCM

Price: $12,000.00

add to cart

AES-CCM is the encryption algorithm used in the IEEE802.11 WiFi standards, it provides privacy and authentication of data. This IP Core implements the algorithm as specified in NIST SP800-38C including 128, 192, 256 bit keys. more info

AES Based Random Number Generator

Price: $14,000.00

add to cart

Implementation of the CTR-DRBG option using the AES cipher in Draft NIST SP-800-90A, Rev 1 (Nov 2014) "Recommendation for Random Number Generation Using Deterministic Random Bit Generators". more info

Źródło:http://www.algotronix-store.com/category_s/23.htm

Cryptographic IPCores

Commercial

CAST

Digital IP Cores and Subsystems

Company | News & Events | Sales & Support | Request Info

Controllers & Processors IP | Compression IP | Peripherals IP | Interconnect IP | Security & Encryption IP

AES

AES Encrypt/Decrypt Core

The AES encryption IP core implements Rijndael encoding and decoding in compliance with the NIST Advanced Encryption Standard. It processes 128-bit blocks, and is programmable for 128-, 192-, and 256-bit key lengths.

Two architectural versions are available to suit system requirements. The Standard version (AES-S) is more compact, using a 32-bit datapath and requiring 44/52/60 clock cycles for each data block (128/192/256-bit cipher key, respectively). The Fast version (AES-F) achieves higher throughput, using a 128-bit datapath and requiring 11/13/15 clock cycles for each data block.

Various cipher modes can be supported (ECB, CBC, OFB, CFB, CTR, CCM, GCM and LRW). The core works with a pre-expanded key, or with optional key expansion logic.

The AES core is a fully synchronous design and has been evaluated in a variety of technologies. It is available optimized for ASICs or FPGAs, with complete deliverables.

This core can be mapped to any any Intel, Lattice, MicroSemi, or Xilinx programmable device, or to any ASIC technology, provided sufficient silicon resources are available. Please contact CAST Sales to get accurate characterization data for your specific implementation requirements. Meanwhile, we provide the following representative results (each in a new pop-up window):

ASIC | intel FPGA | XILINX

Features | More Info

- Encrypts and decrypts using the AES Rijndael Block Cipher Algorithm
- Implemented according to the Federal Information Processing Standard (FIPS) Publication 197 from the US National Institute of Standards and Technology (NIST)
- NIST Certified
- Processes 128-bit data in 32-bit blocks
- Employs user-programmable key size of 128, 192, or 256 bits
- Smallest version supports a single block cipher mode, Electronic Codebook (ECB); these modes can be added as needed:
  - Cipher Block Chaining (CBC)
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
  - Counter (CTR)
  - Counter with CBC-MAC (CCM)
  - Galois/Counter (GCM) and
  - Liskov-Rivest-Wagner (LRW)
- Two architectural versions:

Cryptographic IPCores

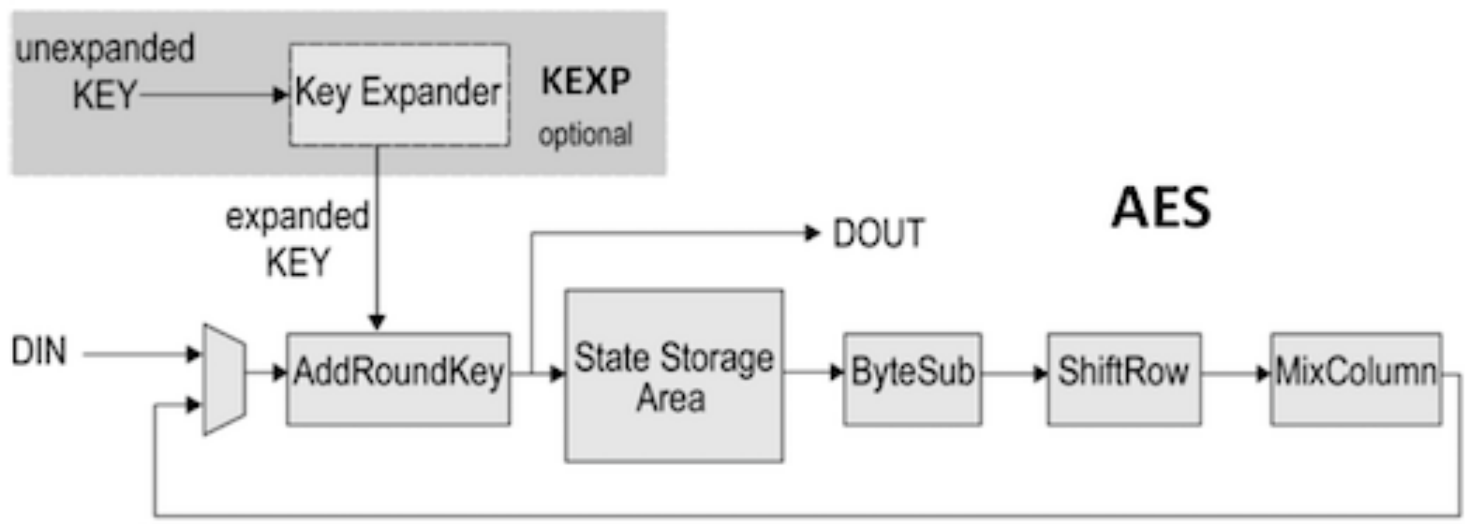
Commercial

Applications

The AES can be utilized for a variety of encryption applications including:

- Protected network routers
- Electronic financial transactions
- Secure wireless communications
- Secure video surveillance systems
- Encrypted data storage

Block Diagram



Support

The core as delivered is warranted against defects for ninety days from purchase. Thirty days of phone and email technical support are included, starting with the first interaction. Additional maintenance and support options are available.

Verification

The core has been verified through extensive synthesis, place and route and simulation runs. It has also been embedded in several products, and is proven in FPGA technologies.

Export Permits

This core implements encryption functions and as such it is subject to export control regulations. Export to your country may or may not require a special export license. Please contact CAST to determine what applies in your specific case.

Deliverables

The core is available in ASIC (synthesizable HDL) and FPGA (netlist) forms, and includes everything required for successful implementation:

32-bit data path size
Processes each 128-bit data block in 44/52/60 clock cycles for 128/192/256-bit cipher keys, respectively

- Fast yields higher transmission rates: 128-bit data path Processes each 128-bit block in 11/13/15 clock cycles for 128/192/256-bit cipher keys, respectively

- Works with a pre-expended key or can integrate the optional key expansion function
- Simple, fully synchronous, reusable design
- Available as fully functional and synthesizable VHDL or Verilog, or as a netlist for popular programmable devices
- Complete deliverables include test benches, C model and test vector generator

Discussion

Check questions

1. What the security of the AES algorithm is based on?
2. What is the role of encryption modes?
3. Does the security of the AES algorithm depend on how it is implemented? If so explain it with an example?