CS326 – Systems Security

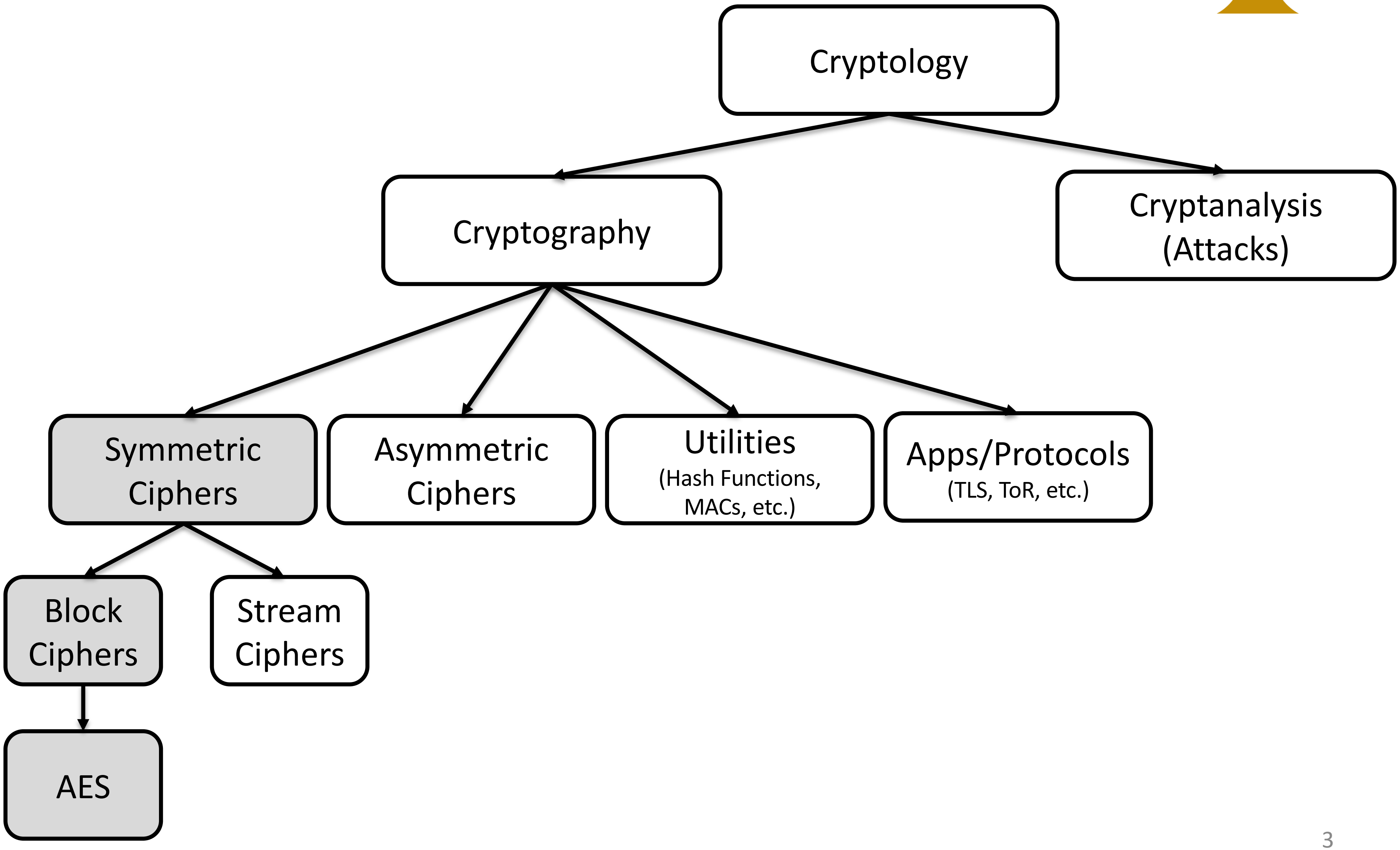
Lecture 5
Advanced Encryption Standard (AES)

Elias Athanasopoulos
athanasopoulos.elias@ucy.ac.cy

Sections of this Lecture

- AES History
- AES Internals

Cryptography Roadmap



AES HISTORY

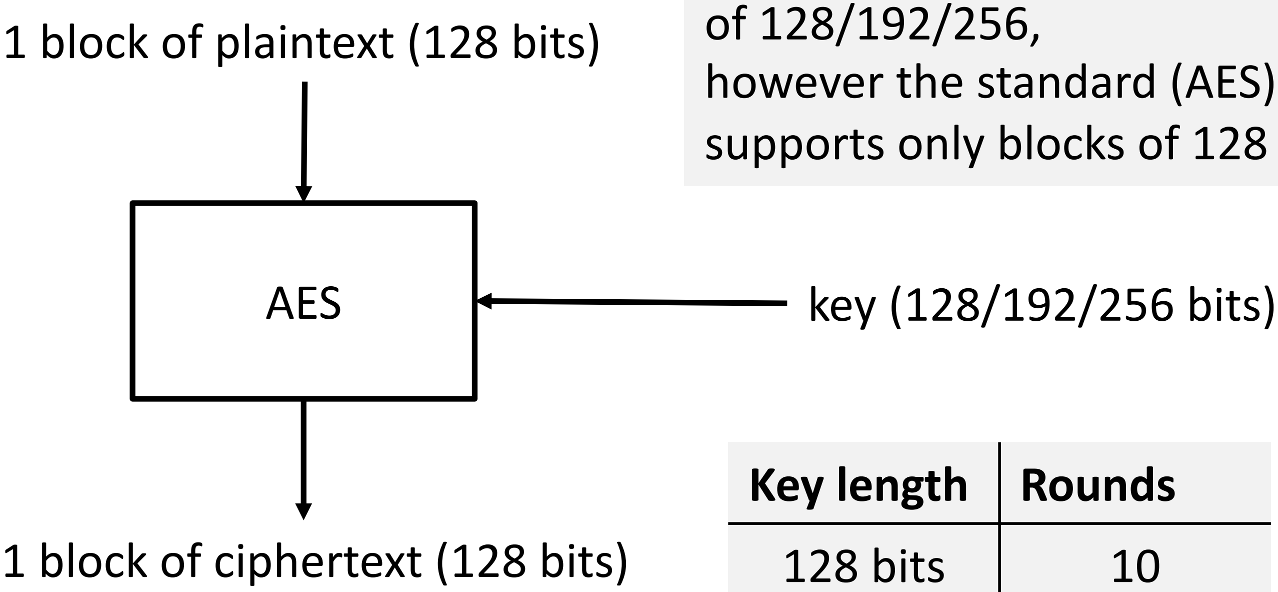
AES History

- 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
- AES was formally approved as a US federal standard in November 2001 6/28

AES INTERNALS

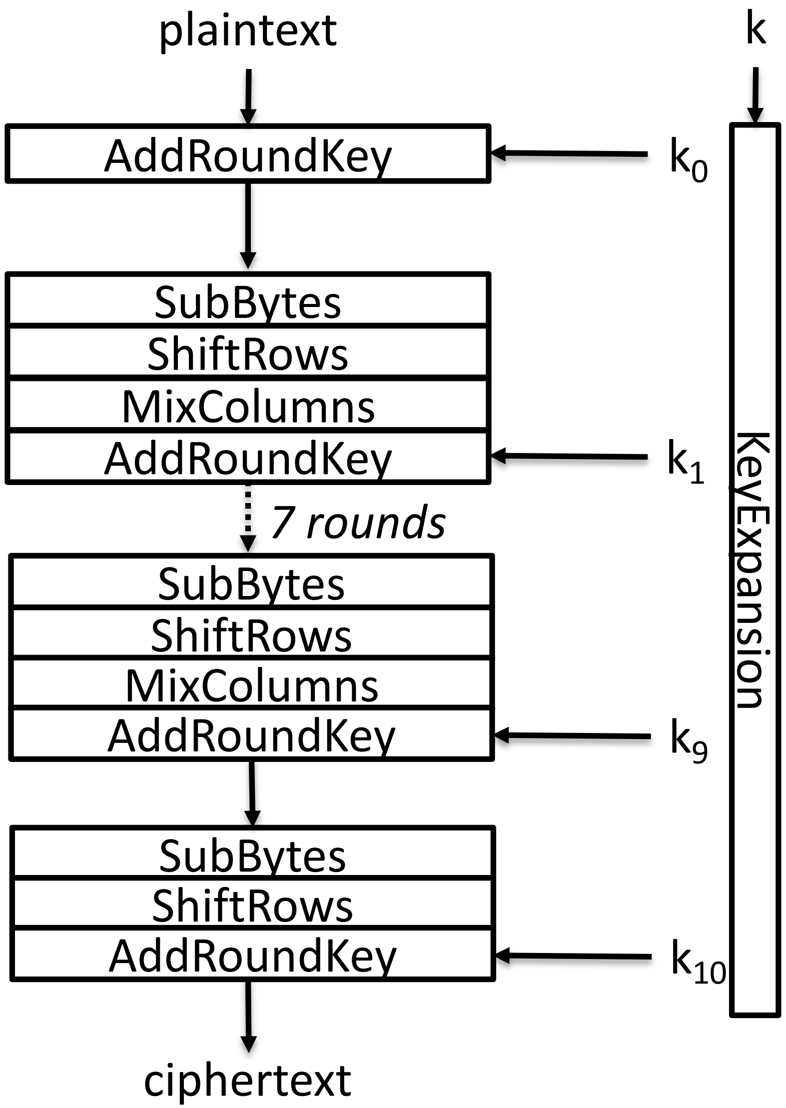
High-level View of AES

Rijndael can work with blocks of 128/192/256, however the standard (AES) supports only blocks of 128 bit



Key length	Rounds
128 bits	10
192 bits	12
256 bits	14

AES/128 Work Flow



Key Scheduling
(KeyExpansion)

For **10** rounds there are **11** round keys

AES Internal State

Block of 128 bits
or 16 bytes

S_0	S_4	S_8	S_{12}
S_1	S_5	S_9	S_{13}
S_2	S_6	S_{10}	S_{14}
S_3	S_7	S_{11}	S_{15}

AES Key State

Key of 192 bits

K_0	K_4	K_8	K_{12}	K_{16}	K_{20}
K_1	K_5	K_9	K_{13}	K_{17}	K_{21}
K_2	K_6	K_{10}	K_{14}	K_{18}	K_{22}
K_3	K_7	K_{11}	K_{15}	K_{19}	K_{23}

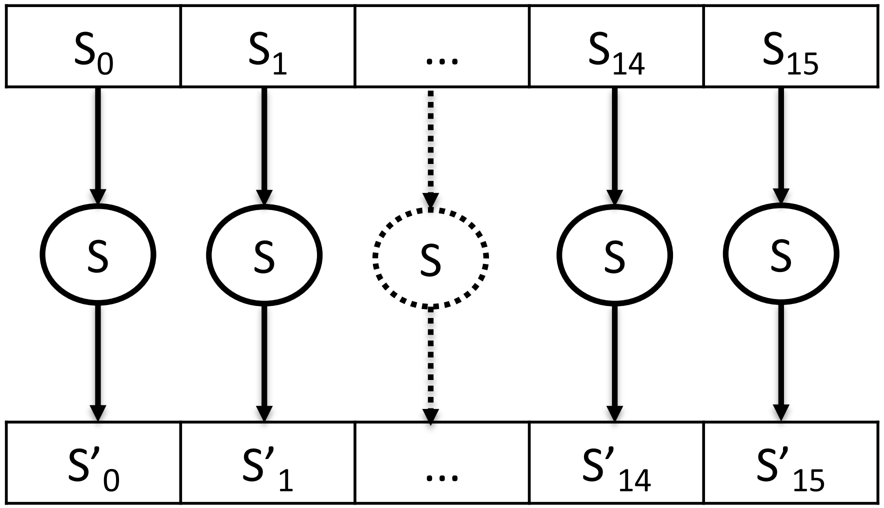
AddRoundKey

- First AddRoundKey applies a XOR operation with the key and the AES state before any round takes place
  - This is called *key whitening*
- All additional AddRoundKey apply a XOR operation with the current AES state

SubBytes

- One AES S-box
- Takes one byte as an input and produces one byte as an output
  - Like a lookup table
- All bytes of the AES state (S_0, S_1, …, S_{15}) pass through the S-Box

SubBytes



AES S-box

	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

The column is determined by the least significant 4 bits, and the row is determined by the other half (0x9a becomes 0xb8)

0x9a = 1001 1010
1001 = 9 (9th row)
1010 = 10 (10th column)

AES S-box

- Compared to DES, the S-box in AES has a very specific rational
- Assuming that each byte from the AES state is a member of the *GF(2^8)*
  - the S-box computes the inverse element and multiplies it by a constant value

Computing Inverses in $G(2^m)$

- Computing an inverse is equal with computing $A^{-1}(x)$ in:

  $A^{-1}(x) \cdot A(x) = 1 \bmod P(x)$,

  where $P(x)$ is the irreducible polynomial
- All elements of $G(2^m)$ have an inverse except 0
- We use pre-computed tables

	00	01	02	…	0F
00	00	01	8d	…	c7
10	74	b4	aa	…	b2
…	…	…	…	…	…
0F	5b	23	38	…	1C

ShiftRows

S'_0	S'_4	S'_8	S'_{12}
S'_1	S'_5	S'_9	S'_{13}
S'_2	S'_6	S'_{10}	S'_{14}
S'_3	S'_7	S'_{11}	S'_{15}

$\rightarrow$

S'_0	S'_4	S'_8	S'_{12}
S'_5	S'_9	S'_{13}	S'_1
S'_{10}	S'_{14}	S'_2	S'_6
S'_{15}	S'_3	S'_7	S'_{11}

MixColumns

- Applies a linear transformation to each of the four column of the state
- Cells with the same color are *equally* transformed
- Transformation is in GF(2^8)
- A change in a single byte affects several bytes

MixColumns Example

C_0	C_4	C_8	C_{12}
C_5	C_9	C_{13}	C_1
C_{10}	C_{14}	C_2	C_6
C_{15}	C_3	C_7	C_{11}

$$\begin{pmatrix} C_0 \\ C_5 \\ C_{10} \\ C_{15} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} S'_0 \\ S'_5 \\ S'_{10} \\ S'_{15} \end{pmatrix}$$

The matrix is fixed
All elements are from GF(2^8)
All operations are in the GF(2^8)

AES Decryption

- AES is not based on a Feistel network
- All steps should be reversed in decryption
- All round keys should be generated in advance and used in the reverse order
- All steps are reversible, since they involve particular operations in $GF(2^8)$

AES Security

- Key space is too large for brute force
- No analytical attack better than brute-force, so far
- Memory Corruption and Side Channels
  - Several attacks target the implementation of AES
  - For instance, a malicious process can *steal* the key of a benign process by just inferring computation (more on that in the software security part of the course)

Resources

- This lecture was built using material that can be found at
  - Chapter 4, Serious Cryptography, https://nostarch.com/seriouscrypto
  - Chapter 4 (*free chapter*), Understanding Cryptography, http://www.crypto-textbook.com
- A gift

  http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html