Lecture Notes

Advanced Discrete Structures

COT 4115.001 S15

2015-02-19

Recap

- Groups
- Fields
  - Arithmetic over $F[X]$
- Finite (Galois) Fields
  - p^n elements where p is prime
  - $\mathbb{Z}_p[X] \pmod{P(X)}$ where $P(X)$ is irreducible over the field

Inverse

Consider the finite field:

$$GF(2^8) = \mathbb{Z}_2[X] \ (\mathrm{mod}\ X^8 + X^4 + X^3 + X + 1)$$

- Since $X^7 + X^5 + X^2 + 1$ is not 0, it should have an inverse.
- Use the extended Euclidean algorithm to compute this

Inverse

$$X^8 + X^4 + X^3 + X + 1 \equiv (X)\,(X^7 + X^5 + X^2 + 1) + (X^6 + X^4 + 1)$$

$$X^7 + X^5 + X^2 + 1 \equiv (X)\,(X^6 + X^4 + 1) + (X^2 + X + 1)$$

$$X^6 + X^4 + 1 \equiv (X^4 + X^3 + X^2 + 1)\,(X^2 + X + 1) + (X)$$

$$X^2 + X + 1 \equiv (X + 1)\,(X) + (1)$$

$$X \equiv (X)\,(1) + 0$$

$$\gcd(X^8 + X^4 + X^3 + X + 1, X^7 + X^5 + X^2 + 1) = 1 \text{ in } \mathbb{Z}_2[X]$$

- Reverse the process to find the inverse.

Inverse

$$1 \equiv (1)(X^2 + X + 1) + (X + 1)(X)$$

$$\equiv (1)(X^2 + X + 1) + (X + 1)[(X^6 + X^4 + 1) + (X^4 + X^3 + X^2 + 1)\,(X^2 + X + 1)]$$

$$\equiv (X^5 + X^2 + X)(X^2 + X + 1) + (X + 1)(X^6 + X^4 + 1)$$

$$\equiv (X^5 + X^2 + X)\,[(X^7 + X^5 + X^2 + 1) + (X)\,(X^6 + X^4 + 1)] + (X + 1)(X^6 + X^4 + 1)$$

$$\equiv (X^5 + X^2 + X)(X^7 + X^5 + X^2 + 1) + (X^6 + X^3 + X^2 + X + 1)(X^6 + X^4 + 1)$$

$$\equiv (X^5 + X^2 + X)(X^7 + X^5 + X^2 + 1) + (X^6 + X^3 + X^2 + X + 1)[(X^8 + X^4 + X^3 + X + 1) + (X)\,(X^7 + X^5 + X^2 + 1)]$$

$$\equiv (X^7 + X^5 + X^4 + X^3)(X^7 + X^5 + X^2 + 1) + (X^6 + X^3 + X^2 + X + 1)(X^8 + X^4 + X^3 + X + 1)$$

Inverse

$$1 \equiv (X^7 + X^5 + X^4 + X^3)(X^7 + X^5 + X^2 + 1) + (X^6 + X^3 + X^2 + X + 1)(X^8 + X^4 + X^3 + X + 1)$$

which means

$$1 \equiv (X^7 + X^5 + X^4 + X^3)(X^7 + X^5 + X^2 + 1) \pmod{X^8 + X^4 + X^3 + X + 1}$$

Hence,

$$\left(X^7 + X^5 + X^2 + 1\right)^{-1} \equiv X^7 + X^5 + X^4 + X^3 \left(\text{mod } X^8 + X^4 + X^3 + X + 1\right)$$

Basic Number Theory - Section 3.11.2

GF(2^8)

$GF(2^8)$

- We've shown that the finite field is given by

$$\mathbb{Z}_2[X] \pmod{X^8 + X^4 + X^3 + X + 1}$$

- Every element can be represented uniquely as a polynomial

$$b_7X^7 + b_6X^6 + b_5X^5 + b_4X^4 + b_3X^3 + b_2X^2 + b_1X + b_0$$

where each b_i is 0 or 1.

- The 8 bits $b_7b_6b_5b_4b_3b_2b_1b_0$ represent a byte, so elements of $GF(2^8)$ may be represented as a byte

$GF(2^8)$ Arithmetic

- **Addition**: XOR of the bits

$$(X^6 + X^5 + X^2 + X + 1) + (X^7 + X^2 + X)$$

$$01100111 \oplus 10000110 = 11100001$$

- **Multiplication**: Consider

$$(X^6 + X^5 + X^2 + X + 1)(X^2)$$

$$\equiv (X^8 + X^7 + X^4 + X^3 + X^2) + (X^8 + X^4 + X^3 + X + 1)$$

$$\equiv X^7 + X^2 + X + 1 \pmod{X^8 + X^4 + X^3 + X + 1}$$

$GF(2^8)$ Arithmetic

- **Multiplication**: Consider

$$(X^6 + X^5 + X^2 + X + 1)(X^2)$$

$$\equiv (X^8 + X^7 + X^4 + X^3 + X^2) + (X^8 + X^4 + X^3 + X + 1)$$

$$\equiv X^7 + X^2 + X + 1 \pmod{X^8 + X^4 + X^3 + X + 1}$$

In binary:

$$01100111 \rightarrow 0110011100 \oplus 0100011011$$

$$\rightarrow 0010000111 = 10000111$$

Multiplication by X^m

1. Shift left, i.e., append m 0s to the end of the byte
2. If the first m bits are 0, then truncate the first m bits and stop.
3. If any of the first m bits are 1, XOR the appropriate multiple of 100011011 to cancel the first 1.
   - Repeat until the first m bits are 0. Go to 2.

Multiplication

Recall:

$$(X^2 + X + 1)(X^5 + X^3 + X^2) =$$
$$X^2\,(X^5 + X^3 + X^2) + X\,(X^5 + X^3 + X^2) + (X^5 + X^3 + X^2)$$

- Arbitrary multiplication can be performed

Comparison

$\mathbb{Z}$	$\leftrightarrow$	$\mathbb{Z}_p[X]$
prime q	$\leftrightarrow$	irreducible $P(X)$ of degree n
$\mathbb{Z}_q$	$\leftrightarrow$	$\mathbb{Z}_p[X]$
field with q elements	$\leftrightarrow$	field with p^n elements

Chapter 5

THE ADVANCED ENCRYPTION STANDARD: RIJNDAEL

History

- In 1997, the NIST put out a call to replace DES
- Requirements:
  1. Allow key sizes of 128, 192, and 256 bits
  2. Operate on 128-bit input blocks
  3. Work on a variety of hardware
     - 8-bit and 32-bit architectures
- Five finalist:
  - MARS (IBM)
  - RC6 (RSA)
  - Rijndael (Daemen, Rijmen)
  - Serpent (Anderson, Biham, Knudsen)
  - Twofish (Schneier, Kelsey, Whiting, Wagner, Hall, Ferguson)

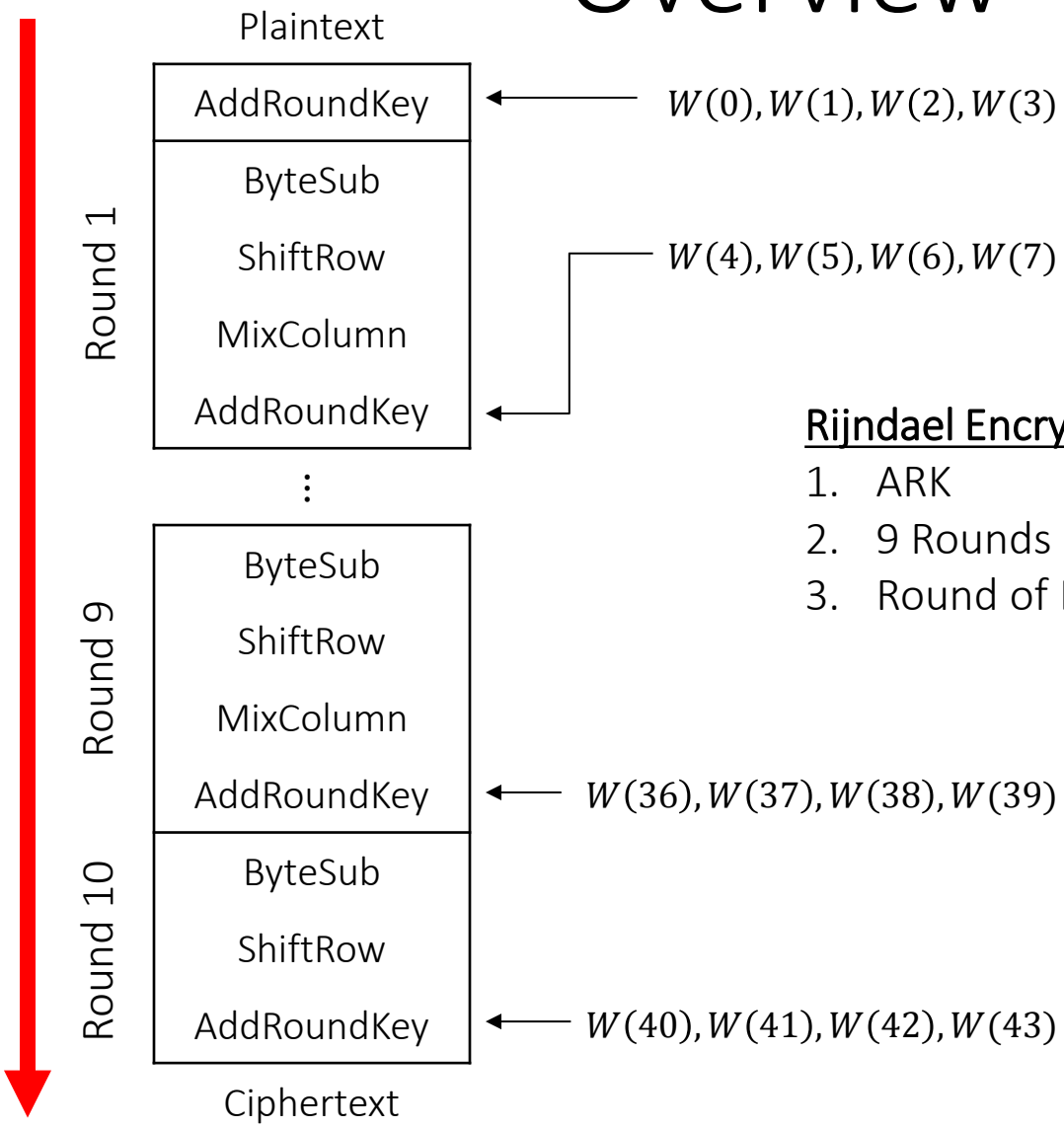
The Advanced Encryption Standard - Section 5.1

THE BASIC ALGORITHM

Overview

- Takes 128-bit input and gives 128-bit output
- Number of rounds depends on key size:
  - 128-bit key → 10 round
  - 192-bit key → 12 round
  - 256-bit key → 14 round
- Each round (including 0) has a round key
- Complete rounds have 4 *layers*:
  - **ByteSub Transformation (BS)**: resists differential and linear cryptanalysis
  - **ShiftRow Transformation (SR)**: creates diffusion
  - **MixColumn Transformation (MC)**: similar to SR
  - **AddRoundKey (ARK)**: Round key is XORed with the result

Overview



Rijndael Encryption

1. ARK (0^{th} RK)
2. 9 Rounds BS,SR,MC,ARK (1-9^{th} RK)
3. Round of BS,SR,ARK (10^{th} RK)

The Advanced Encryption Standard - Section 5.2

THE LAYERS

Setup

- Input is 128 bits grouped into 16 bytes:

$$\begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

- Arithmetic ($+, \cdot$) on the bytes is assumed to be on the finite field $GF(2^8)$, constructed with the irreducible polynomial

$$P(X) = X^8 + X^4 + X^3 + X + 1$$

ByteSub Transformation (BS)

- $BS: \mathbb{Z}_2^8 \rightarrow \mathbb{Z}_2^8$ is a permutation
- Each byte

$$a_{i,j} = \underbrace{\beta_0\beta_1\beta_2\beta_3}_{\text{Row}}\ \underbrace{\beta_4\beta_5\beta_6\beta_7}_{\text{Column}}$$

is substituted by another byte $b_{i,j}$ according to the following S-box

ByteSub S-Box

99	124	119	123	242	107	111	197	45	1	103	43	254	215	171	118
202	130	201	125	250	89	71	240	173	212	162	175	156	164	114	192
183	253	147	38	54	63	247	204	52	165	229	241	113	216	49	21
4	199	35	195	24	150	5	154	7	18	128	226	235	39	178	117
9	131	44	26	27	110	90	160	82	59	214	179	41	227	47	132
83	209	0	237	32	252	177	91	106	203	190	57	74	76	88	207
208	239	170	251	67	77	51	133	69	249	2	127	80	60	159	168
81	163	64	143	146	157	56	245	188	182	218	33	16	255	243	210
205	12	19	236	95	151	68	23	196	167	126	61	100	93	25	115
96	129	79	220	34	42	144	136	70	238	184	20	222	94	11	219
224	50	58	10	73	6	36	92	194	211	172	98	145	149	228	121
231	200	55	109	141	213	78	169	108	86	244	234	101	122	174	8
186	120	37	46	28	166	180	198	232	221	116	31	75	189	139	138
112	62	181	102	72	3	246	14	97	53	87	185	134	193	29	158
225	248	152	17	105	217	142	148	155	30	135	233	206	85	40	223
140	161	137	13	191	230	66	104	65	153	45	15	176	84	187	22

ByteSub Transformation (BS)

Input (after ARK)

Output (after BS)

$$\begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix} \rightarrow \begin{pmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix}$$

Example

$$\begin{pmatrix} 01110100 & 10101110 & 10011110 & 01110101 \\ 00101011 & 00001010 & 10000111 & 01101101 \\ 01010110 & 11011110 & 01001110 & 10010011 \\ 10111001 & 00110100 & 10101011 & 11101010 \end{pmatrix}$$

$$\downarrow$$

$$\begin{pmatrix} 10010010 & 11100100 & 00001011 & 10011101 \\ 11110001 & 01100111 & 00010111 & 00111100 \\ 10110001 & 00011101 & 00101111 & 11011100 \\ 01010110 & 00011000 & 01100010 & 10000111 \end{pmatrix}$$

$$0111 = 7, \qquad 0100 = 4 \qquad \Longrightarrow \qquad 01110100 \rightarrow 146 = 10010010$$

ShiftRow Transformation (SR)

- Shift the four rows of the matrix cyclically to the left by offsets of 0, 1, 2, and 3 to obtain

$$\begin{pmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b_{0,0}} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,1} & b_{1,2} & b_{1,3} & \boldsymbol{b_{1,0}} \\ b_{2,2} & b_{2,3} & \boldsymbol{b_{2,0}} & b_{2,1} \\ b_{3,3} & \boldsymbol{b_{3,0}} & b_{3,1} & b_{3,2} \end{pmatrix}$$

Example

$$\begin{pmatrix} 10010010 & 11100100 & 00001011 & 10011101 \\ 11110001 & 01100111 & 00010111 & 00111100 \\ 10110001 & 00011101 & 00101111 & 11011100 \\ 01010110 & 00011000 & 01100010 & 10000111 \end{pmatrix}$$

$$\begin{pmatrix} 10010010 & 11100100 & 00001011 & 10011101 \\ 01100111 & 00010111 & 00111100 & 11110001 \\ 00101111 & 11011100 & 10110001 & 00011101 \\ 10000111 & 01010110 & 00011000 & 01100010 \end{pmatrix}$$

MixColumn Transformation (MC)

Multiply the byte matrices using the addition and multiplication rules for $GF(2^8)$:

$$\begin{pmatrix} 00000010 & 00000011 & 00000001 & 00000001 \\ 00000001 & 00000010 & 00000011 & 00000001 \\ 00000001 & 00000001 & 00000010 & 00000011 \\ 00000011 & 00000001 & 00000001 & 00000010 \end{pmatrix} \begin{pmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{pmatrix} =$$

$$\begin{pmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{pmatrix}$$

Example

$$\begin{pmatrix} \mathbf{10010010} & 11100100 & 00001011 & 10011101 \\ \mathbf{01100111} & 00010111 & 00111100 & 11110001 \\ \mathbf{00101111} & 11011100 & 10110001 & 00011101 \\ \mathbf{10000111} & 01010110 & 00011000 & 01100010 \end{pmatrix}$$

$$\downarrow$$

$$\begin{pmatrix} 01101000 & 00001010 & 11111011 & 01010110 \\ 10101010 & 10001001 & 11110101 & 01110111 \\ 01101111 & 10111000 & 00110000 & 11110000 \\ 11110000 & 01111110 & 10100000 & 11000010 \end{pmatrix}$$

$$\begin{aligned} & 0000010 \cdot \mathbf{10010010} + 00000011 \cdot \mathbf{01100111} + 00000001 \cdot \mathbf{00101111} + 00000001 \cdot \mathbf{10000111} \\ = \; & 01101001 + 10101001 + 00101111 + 10000111 \\ = \; & 01101000 \end{aligned}$$

Add Round Key (ARK)

Add the byte matrices using the addition rule for $GF(2^8)$, i.e., XOR:

$$\begin{pmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{pmatrix} \oplus \begin{pmatrix} k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} \end{pmatrix} = \begin{pmatrix} e_{0,0} & e_{0,1} & e_{0,2} & e_{0,3} \\ e_{1,0} & e_{1,1} & e_{1,2} & e_{1,3} \\ e_{2,0} & e_{2,1} & e_{2,2} & e_{2,3} \\ e_{3,0} & e_{3,1} & e_{3,2} & e_{3,3} \end{pmatrix}$$

The Key Schedule

- The original key consists of 128 bits
  - Arrange into 4×4 matrix with columns labeled $W(0), W(1), W(2), W(3)$
  - Add 40 more columns $W(4), W(5), \ldots, W(43)$ according to the recursive definition:
    - If $i \not\equiv 0 \pmod 4$,

$$W(i) = W(i-4) \oplus W(i-1)$$

    - If $i \equiv 0 \pmod 4$,

$$W(i) = W(i-4) \oplus T(W(i-1))$$

$T(W(i-1))$

$$W(i-1) = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \quad \overset{\text{Rotate Cyclically}}{\Longrightarrow} \quad \begin{pmatrix} b \\ c \\ d \\ a \end{pmatrix} \quad \overset{\text{S-box Substitution}}{\Longrightarrow} \quad \begin{pmatrix} e \\ f \\ g \\ h \end{pmatrix}$$

$r(i) = 00000010^{(i-4)/4}$ using multiplication in $GF(2^8)$

$$T(W(i-1)) = \begin{pmatrix} e \oplus r(i) \\ f \\ g \\ h \end{pmatrix}$$