3.3 블록 암호 – AES

■ AES(Advanced Encryption Standard) 알고리즘

- DES의 한계: 56비트 키 -> 전수조사(Brute-force) 현실화
- 3DES의 한계: 보안은 ↑, 속도는 ↓(DES 3회)
- NIST 공모(1997) -> Rijndael 채택(2001): 안전성, 효율성, 단순 구현성에서 우수
- 핵심 차이:
  - AES : SPN(Substitution Permutation Network)
  - DES : Feistel

- 블록 암호지만 파이스텔 암호는 아님
- AES를 복호화하려면 AES의 연산 과정을 거꾸로 수행할 수 있어야 한다는 의미
- 고도의 수학적 구조를 가짐

3.3 블록 암호 – AES

■ AES와 관련된 몇 가지 사항

- 블록 크기 : 128비트 (고정)
- 키 길이 : 128비트, 192비트, 256비트의 세 개의 키 길이 사용
- 라운드 수(Nr)
  - AES-128:10
  - AES-192:12
  - AES-256:14
- 상태(State) : 128비트를 4x4 바이트 행렬로 표현
- 각 라운드은 3개 층, 4개 함수로 구성
  - ByteSub(비선형 층)
  - ShiftRow(선형 혼합 층)
  - MixColumn(비선형 층)
  - AddRoundKey(키 추가 층)

3.3 블록 암호 – AES

■ AES의 4가지 함수

- ByteSub, ShiftRow, MixColumn, AddRoundKey는 모두 역함수가 존재.
- 전체 알고리즘의 역이 당연히 성립되므로 AES는 암호화뿐만 아니라 복호화도 가능함

3.3 블록 암호 – AES

AES의 4가지 함수

- ByteSub 연산

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \rightarrow \text{ByteSub} \rightarrow \begin{bmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{bmatrix}$$

3.3 블록 암호 – AES

■ AES의 4가지 함수

- ByteSub 연산
  - 각 바이트 x에 대해 $GF(2^8)$에서 곱셈 역원 x^{-1} 계산(0은 0으로), 이후 아핀 변환 적용

  - 아핀 변환(Affine Transformation)
    - 선형 변환
      - » 어떤 벡터 x에 대해 행렬 A를 곱하는 것 $(y = A \cdot x)$
    - 아핀변환
      - » 선형 변환 뒤에 상수 벡터 b를 더한 것 $(y = A \cdot x \oplus b)$

    즉 선형변환 + 평행 이동이 합쳐진 상태

3.3 블록 암호 – AES

■ AES의 4가지 함수

▪ ByteSub 연산

- 아핀 변환(Affine Transformation)
  - 선형 변환만 있으면, 입력과 출력이 원점(0)을 기준으로 항상 일정하게 대응
    -> $y = 2x$는 언제나 0을 통과
  - 암호학에서는 0도 특별한 의미를 갖지 않게 만들고자 함
    -> 그래서 단순한 선형 변환이 아니라, 상수 벡터를 더해 원점을 이동시킴
  - 선형 변환은 "늘리고, 돌리고, 비트 섞기"만 하는 것
  - 아핀 변환은 "위치를 옮기는것 " 까지 추가

3.3 블록 암호 – AES

■ AES의 4가지 함수

- ByteSub 연산
  - 각 바이트 x에 대해 $GF(2^8)$에서 곱셈 역원 x^{-1} 계산(0은 0으로), 이후 아핀 변환 적용
    - $S(x) = A \cdot x^{-1} \oplus b$
    - 기약다항식 $m(x) = x^8 + x^4 + x^3 + x + 1$ 로 정의된 $GF(2^8)$ 사용
    - A: 고정 8x8비트 행렬 $b = 0x63$
  - 비선형성(Confusion)제공 / 선형/차분 공격 저항성의 핵심.

- 아핀 변환(Affine Transformation)
  - 선형 변환
    - 어떤 벡터 x에 대해 행렬 A를 곱하는 것 ($y = A \cdot x$)
  - 아핀변환
    - 선형 변환 뒤에 상수 벡터 b를 더한 것 ($y = A \cdot x \oplus b$)

  즉 선형변환 + 평행 이동이 합쳐진 상태

3.3 블록 암호 – AES

■ AES의 4가지 함수

▪ ByteSub 연산

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \rightarrow \text{ByteSub} \rightarrow \begin{bmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{bmatrix}$$

표 3-5 AES ByteSub

	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

3C -> ByteSub -> ?

3.3 블록 암호 – AES

■ AES의 4가지 함수

- ShiftRows 연산
  - 0행:0칸, 1행 1칸, 2행 2칸, 3행 3칸 왼쪽 순환 이동
  - 역연산은 같은 횟수만큼 오른쪽 순환

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \\ c_0 & c_1 & c_2 & c_3 \\ d_0 & d_1 & d_2 & d_3 \end{bmatrix} \Rightarrow \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 & b_0 \\ c_2 & c_3 & c_0 & c_1 \\ d_3 & d_0 & d_1 & d_2 \end{bmatrix}$$

3.3 블록 암호 – AES

■ AES의 4가지 함수

▪ MixColumns – 열 단위 선형 연산

- 각 열 벡터 $[a_0, a_1, a_2, a_3]^T$에 대해

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (\mathrm{GF}(2^8))$$

- 역연산 InvMixColumns

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}$$

3.3 블록 암호 – AES

AES의 4가지 함수

- MixColumn 연산

$$\begin{bmatrix} a_{0i} \\ a_{1i} \\ a_{2i} \\ a_{3i} \end{bmatrix} \rightarrow \text{MixColumn} \rightarrow \begin{bmatrix} b_{0i} \\ b_{1i} \\ b_{2i} \\ b_{3i} \end{bmatrix}, for\ i = 0,1,2,3$$

3.3 블록 암호 – AES

AES의 4가지 함수

- AddRoundKey 연산

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \oplus \begin{bmatrix} k_{00} & k_{01} & k_{02} & k_{03} \\ k_{10} & k_{11} & k_{12} & k_{13} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{30} & k_{31} & k_{32} & k_{33} \end{bmatrix} = \begin{bmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{bmatrix}$$

3.3 블록 암호 – AES

■ 키 스케줄(Key Expansion)

- 공통 파라미터
  - $N_b = 4$ (블록 4워드), $N_k \in \{4,6,8\}$(키워드 수)
  - 라운드 수 $N_r \in \{10,12,14\}$
- 생성규칙(AES-128 example)
  - 입력 16바이트 $\rightarrow$ 4워드 $W[0 \ldots 3]$
  - 총 $(N_r + 1) \times N_b = 44$개의 워드 생성 $W[0 \ldots 43]$
  - 규칙
    - $W[i] = W[i-4] \oplus W[i-1]$
    - 매 4번째 i: $W[i] = W[i-4] \oplus SubWord\big(RotWord(W[i-1])\big) \oplus Rcon[i \backslash 4]$
  - SubWord: S-box 바이트 치환, RotWord: 1바이트 왼쪽 순환
- 라

3.3 블록 암호 – AES

■ 전체 흐름

- SubBytes: 각 바이트 S-box 치환 (예: 3C→EB, …)
- ShiftRows: 행 이동으로 위치 재배치
- MixColumns
- AddRoundKey: 위 결과에 라운드 키 열 XOR → 다음 상태

■ **복호화**

- 역순 라운드: InvShiftRows → InvSubBytes → AddRoundKey → InvMixColumns
- 마지막 라운드는 InvMixColumns 없음.
- 키는 마지막 라운드 키부터 역순으로 사용.

3.3 블록 암호 – AES

■ 보안성 개요

- 알고리즘 보안: AES-128/192/256에 대해 실용적 키 복구 공격 없음.
- 주의: 현실 공격은 대부분 사이드채널(타이밍, 캐시, 전력 분석).
  - 대응: 상수시간 구현, T-table 대신 비트슬라이싱 또는 AES-NI(하드웨어 명령) 사용.
  - 양자시대: Grover로 키 탐색이 가속 → AES-128도 여전히 실용적(유효 64비트 수준)이나 AES-256 권장이 늘어나는 추세.