A Field is:

1. A collection F of objects
2. Two binary operations x and + closed on F
3. F contains multiplicative identity 1 where (1 x y) = y for all y in F
4. F contains additive identity 0 where (0 + y) = y for all y in F.
5. For each y in F, there exists a z in F such that (y + z) = 0. (Additive inverse)
6. For each y in F, except 0, there exists a z in F such that (y x z) = 0. (Multiplicative inverse)
7. Associative, commutative, distributive laws work as expected

Shorthands:

- a^{-1} is a's multiplicative inverse
- -a is a's additive inverse
- a-b is short for a + -b
- a/b is short for a x b^{-1}

Examples:

- R with standard addition and multiplication form a field.
- Q with standard addition and multiplication form a field.
- Z with standard addition and multiplication DOESN'T form a field. (a^{-1} doesn't exists for most a.)
- Z_p forms a filed with p prime and addition and multiplication mod p. (p must be prime to make sure every element has a multiplicative inverse.)
- THEOREM: If a is prime, then there is a field of size a^n for each n > 0.
- Z_p is not convenient for high-speed processing: mod p is expensive and standard data type don't hold a prime number of values
- Since 2 is prime there is a field of size 2^n for all n > 0. This is promising because all data types can hold power-of-two different values.
- Galois Fields (Évariste Galois died age 20 in a duel, 1823)
- The set of all bit sequences of length n forms a field called GF(2^n). We will use GF(256) in this class.
- GF(256) = {00000000, 00000001, 00000010,…,11111111}

Addition:

- Interpret the bits as coefficients of a degree 7 polynomial with variable x.
- Add the two polynomials, to keep coefficients 0 or 1, mod each coefficient by 2.
- Concat the coefficients of the resulting degree 7 polynomial.
- Shortcut: Xor'ing the two bytes produces the same result.

Example:

$00001001 + 10000001$

$x^3 + x^0 \;\; + \;\; x^7 + x^0$

$x^7 + x^3 + 2x^0$

$x^7 + x^3$

10001000

Multiplication:

- Interpret the bits as coefficients of a degree 7 polynomial with variable x.
- Multiply the two polynomials, to keep coefficients 0 or 1, mod each coefficient by 2.
- Mod the result by $x^8 + x^4 + x^3 + x + 1$
- Concat the coefficients of the resulting degree 7 polynomial.
- Shortcut: No shortcut. Multiplication is expensive.

Example:

$00001001 \;\; x \;\; 10000001$

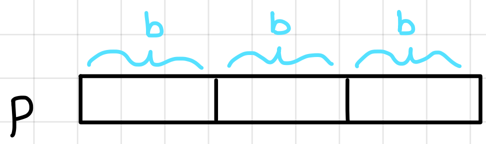
$(x^3 + x^0)(x^7 + x^0)$ mod $x^8 + x^4 + x^3 + x + 1$

$x^{10} + x^7 + x^3 + x^0$ mod $x^8 + x^4 + x^3 + x + 1$

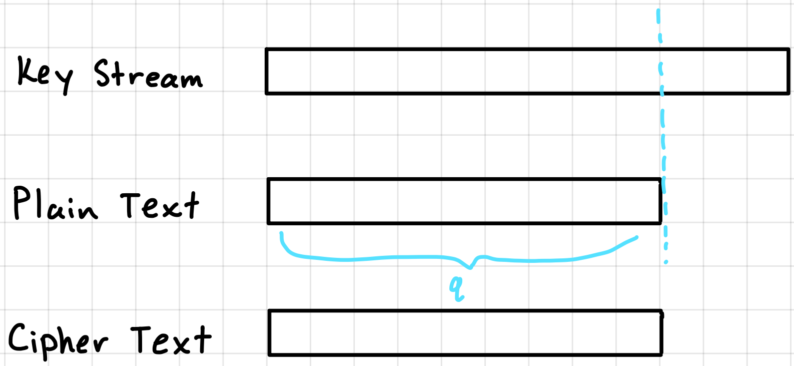
$x^7 + x^6 + x^5 + x^2 + x^0$

11100101

Padding



* Stream Cipher: no padding *



CBC / ECB

$P \longrightarrow \boxed{\text{Padding}} \longrightarrow p' \longrightarrow \boxed{\text{Encrypt}} \longrightarrow c$

$c \longrightarrow \boxed{\text{Decrypt}} \longrightarrow p' \longrightarrow \boxed{\text{unpad}} \longrightarrow p$

Need:

- unpad is inverse of pad
- p' is a multiple of b
- efficient

NOTE: There will always be padding for CBC and ECB in this class

10^* padding

- Append enough ∅'s to next multiple of b

  b = 16 bits

  p = 1111 0000 1111

  p' = 1111 0000 1111 1000

$$p' = \underbrace{1111\ 0000\ 1111\ 0000}\ \underbrace{1000\ 0000\ 0000\ 0000}_{\text{padding}}$$

Sometimes there will be no trailing ∅'s

p = 1111 1111 1111 111

p' = 1111 1111 1111 1111 ← no ∅'s

p = 1111 1111 1111 111 ← strip only the last '1', no ∅'s

Mode Examples

Given:

$E: \{0,1\}^6 \rightarrow \{0,1\}^6$

$E(x) = ROTL(x, 2)$

If needed:

nonce = 101

IV = 110111

Counter start at <1>

10* padding

Encrypt 0000 1111 0000 1111

Note:

ECB: Can Encrypt and Decrypt in parallel

CBC: Decryption can be done in parallel, but
Encryption cannot be done in parallel

CTR: Can Encrypt and Decrypt in parallel

OFB: Cannot do either Encryption
or Decryption in parallel

ECB:

p' 000011 110000 111000 ← padding Added

E

c 001100 000011 100011

E^{-1}

p' 000011 110000 111000

p 000011 110000 11 ← padding removed

CBC:

p 000011 110000 11

p' 000011 110000 111000

IV: 110111 → ⊕

110100 100011 110110

E E E

c 010011 001110 011011

p 000011 110000 11

p' 000011 110000 111000

IV: 110111 → ⊕

110100 100011 110110

E^{-1} E^{-1} E^{-1}

c 010011 001110 011011

NOTE: Decryption can be done in parallel, but not encryption

CTR:

101001   101010   101011

$\downarrow$ $\downarrow$ $\downarrow$

E   E   E

$\downarrow$ $\downarrow$ $\downarrow$

Key Stream   100110   101010   101110

$\oplus$

P   000011   110000   11

C   100101   011010   01

To decrypt, use same key stream.

XOR key stream $\oplus$ C

101001   101010   101011

$\downarrow$ $\downarrow$ $\downarrow$

E   E   E

$\downarrow$ $\downarrow$ $\downarrow$

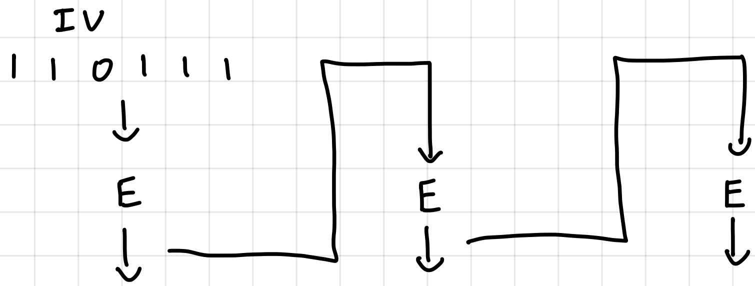
Key Stream   100110   101010   101110

$\oplus$

C   100101   011010   01

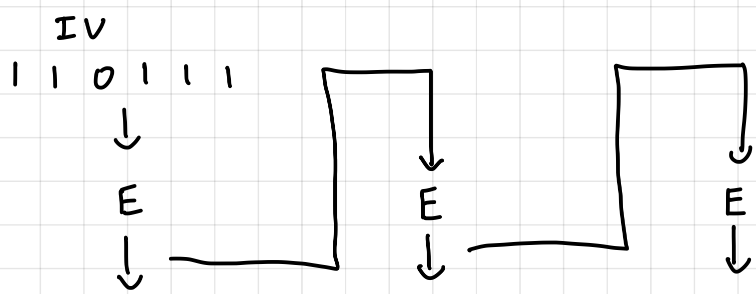
P   000011   110000   11

OFB:



Key Stream	0 1 1 1 1 1	1 1 1 1 0 1	1 1 0 1 1 1
⊕			
P	0 0 0 0 1 1	1 1 0 0 0 0	1 1 1 1
C	0 1 1 1 0 0	0 0 1 1 0 1	0 0 1 0

To Decrypt:



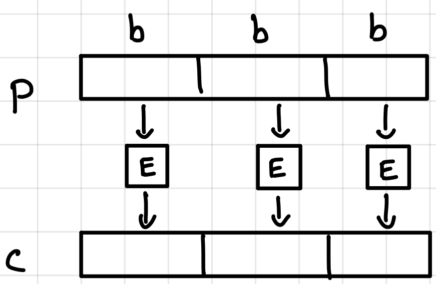
Key Stream	0 1 1 1 1 1	1 1 1 1 0 1	1 1 0 1 1 1
⊕			
C	0 1 1 1 0 0	0 0 1 1 0 1	0 0 1 0
P	0 0 0 0 1 1	1 1 0 0 0 0	1 1 1 1

Formally encryption security model: Indistinguishible from random.

Equal # of random bits — Distinguisher — Real Encryption

Let $E: \{0,1\}^b \to \{0,1\}^b$ be a random permutation

ECB (electronic codebook)



World 1

on $f(x)$
return $ECB(x)$

World 2

on $f(x)$
return $|x|$ random bits

idea 1: if $f(\langle 0 \rangle_b) = f(\langle 0 \rangle_b)$
output real
else
output random

Same thing twice = ECB

$$\text{Advantage} = 1 - \frac{1}{2^b} \approx 1$$

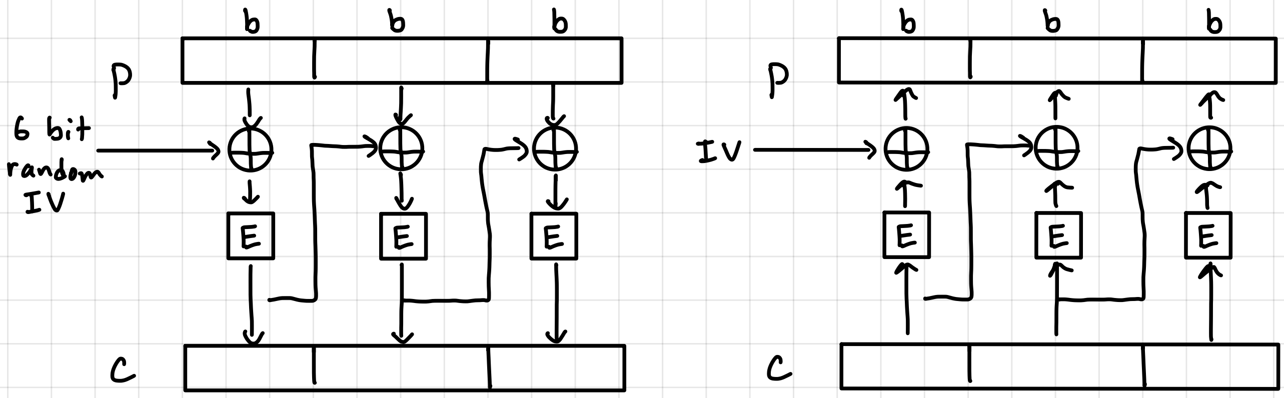
idea 2: $x = f(\langle 0 \rangle_{2b})$

$x_0 \| x_1 = x$ // split in half

if $x_0 = x_1$
output real

else
output random

CBC (cipher block chaining)



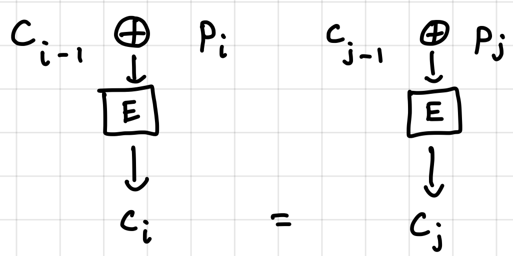
Send: $\underbrace{(IV, C)}_{\text{ciphertext expansion by 6 bits}}$

not parallelizable

parallelizable

- sends both the IV and ciphertext on the wire
  → use this to decrypt ciphertext

observe that: if $c_i = c_j$



then $c_{i-1} \oplus p_i = c_{j-1} \oplus p_j$

$$\boxed{p_i \oplus p_j = c_{i-1} \oplus c_{j-1}}$$

Distinguisher:

for $i = 1, 2, \ldots, q$

IV_i = random b bits

P_i = random b bits

$c_i = f(iv_i, P_i)$

if $(c_i = c_j)$ for any $j < i$

if $iv_i \oplus iv_j = P_i \oplus P_j$

output real CBC

else

output random

Adv ≈ 1 when a repeat occurs

Probability of repeat $\propto \dfrac{q^2}{2^b}$ (binary bound)

Thus, Adv $\approx \dfrac{q^2}{2^b}$ $\Leftarrow$ good if q is small or b is large

For example: AES $b = 128$

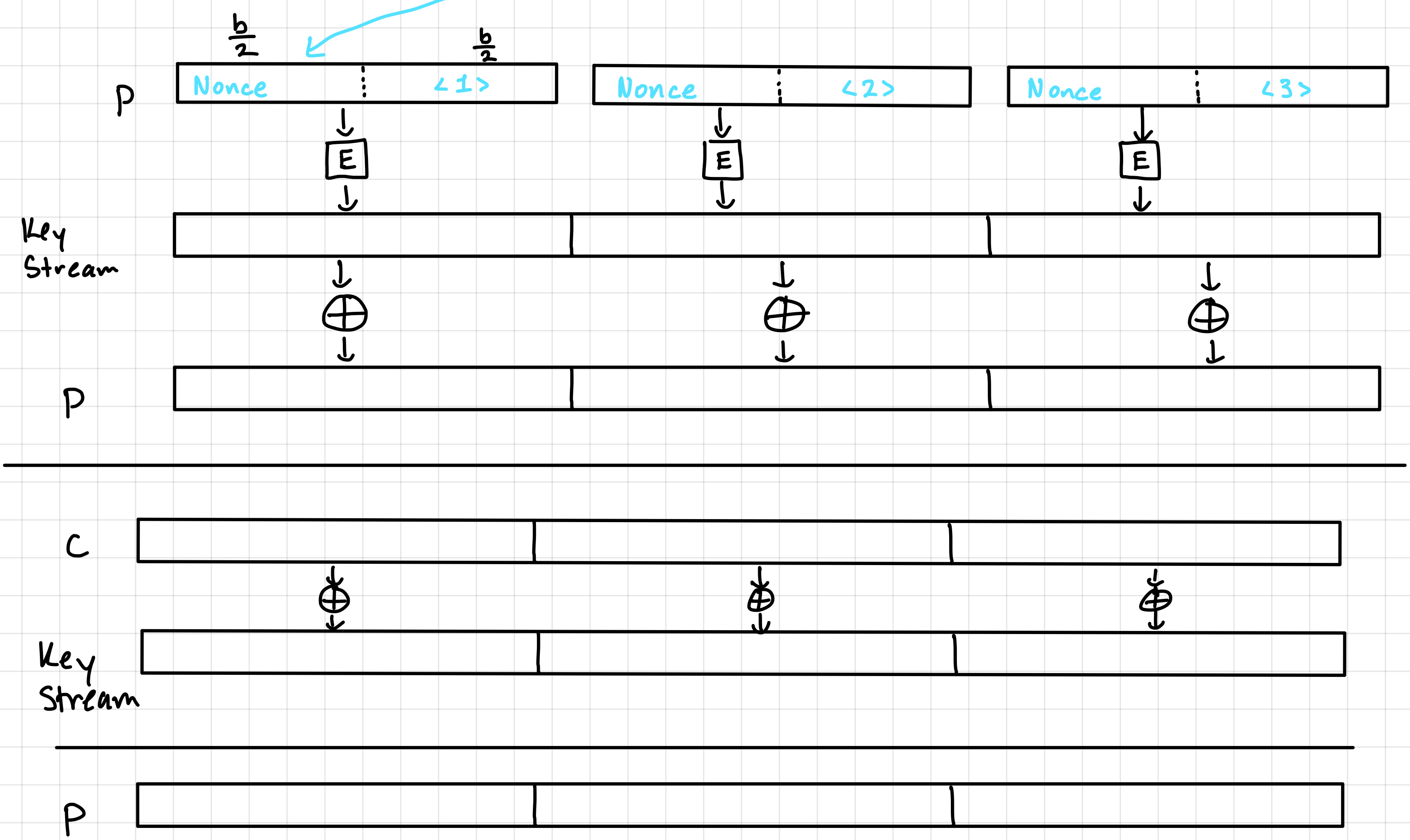
$$\frac{q^2}{2^{128}} < 2^{-32}$$

$$q^2 < 2^{96}$$

$$q < 2^{48}$$

CTR (Counter)

Nonce: number used once (doesn't have to be random)

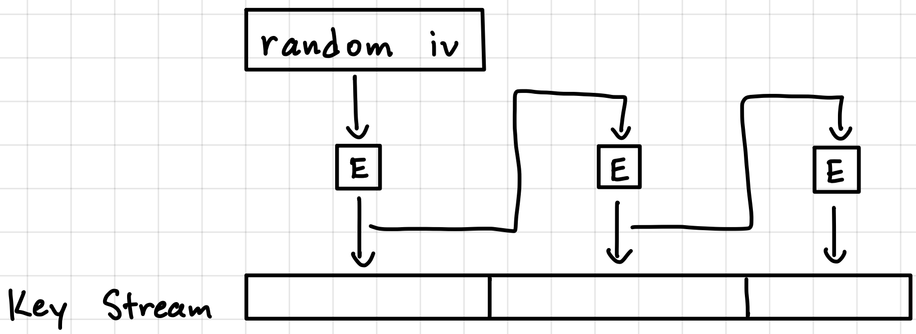


- No inverse E^{-1} needed

send(nonce, c)

- No separate decryption

OFB (Output Feedback)



Birthday bound $\approx \frac{q^2}{N}$ The probability that q random values from a domain size N has at least one respected pair.

Pr[	choice	1	matches	a	prior	choice]	$= 0$
"	"	2	"	"	"	"]	$= \frac{1}{N}$
"	"	3	"	"	"	"]	$\leq \frac{2}{N}$
"	"	4	"	"	"	"]	$\leq \frac{3}{N}$
"	"	q	"	"	"	"]	$\leq \frac{q-1}{N}$

Summation

Pr[any of the first q choices match] $\leq \Sigma = \frac{q(q-1)}{2N}$

Birthday Bound is an upper bound $< \frac{q^2}{N}$

Distinguishing: Block cipher vs. random permutation

Security bounds: range of possible attack advantages.

- lower bounds: an attacker can achieve at least this much. Show via an attack

- upper bounds: no attacker can do better than this.

Lower bound on Block cipher vs. random permutation

Let $E: \{0,1\}^k \to (\{0,1\}^b \to \{0,1\}^b)$ be a block cipher

World 1

k = random k bits

$f = E_k$

World 2

$f = \{0,1\}^b \to \{0,1\}^b$ random perm

Brute Force: Try all possible

Distinguish(f):

$x_0 = f(\langle 0 \rangle)$

$x_1 = f(\langle 1 \rangle)$

for $i = 1$ to t

  if $(x_0 = E_{\langle i \rangle}(\langle 0 \rangle)$ and $x_1 = E_{\langle i \rangle}(\langle 1 \rangle))$

    output "block cipher"

output "random perm"

Advantage = Pr[output block cipher | f is block cipher]

  − Pr[output block cipher | f is random perm]

percentage of keys tried over t time

$$= \frac{t}{2^k} - \left(\frac{1}{2^b} \times \frac{1}{2^b}\right)t$$

$$= \frac{t}{2^k} - \frac{t}{2^{2b}}$$

$$= t\left(\frac{1}{2^k} - \frac{1}{2^{2b}}\right)$$

Much smaller than $\frac{1}{2^k}$

$$\boxed{\text{Advantage} \approx \frac{t}{2^k}}$$

lower bound on Block Cipher Security : $\frac{t}{2^k}$

upper bound on Block Cipher Security : ??? ε

// Unknown placeholder for unknown upper bound.

Assume to be $\approx \frac{t}{2^k}$ for a good block cipher

Distinguishing: Block cipher vs. random function

Block cipher   random perm   random func

$\frac{t}{2^k}$ $\frac{q}{2^b}$

$$\boxed{\frac{t}{2^k} + \frac{q}{2^b}}$$

t Time spent offline in key search

q # of black box invocations

Upper bound on CTR encryption: no adversary can do better

* Proven via "reduction"

Let f be either a block cipher with random key or a random function

BC Distinguisher (f)

    let g = CTR encryption using f

    if Real or Random Distinguisher (g) = "real"

        output "block cipher"

    else

        output "random function"

A Reduction

If RRDist exist, then BCDist exists

Note: if f is a block cipher, then g is exactly CTR mode

if f is a random function, then g output uniform random bits

These are the two worlds a real or random distinguisher looks as.

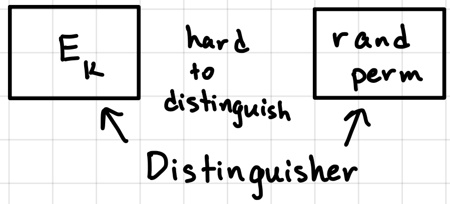
if BCDistinguisher adv $< x$, then RRDistinguisher adv $< x$

BCDistinguish advantage $< \frac{t}{2^k} + \frac{q}{2^b}$

so RRDistinguisher adv $< \frac{t}{2^k} + \frac{q}{2^b}$

* Reductions will not be studied in this class

Block cipher is intended to resemble a random permutation



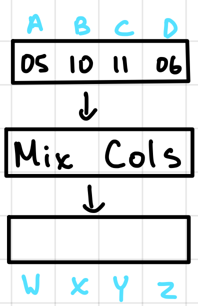
Byte Substitution:

$$S(x) = x^{-1} \cdot \underbrace{C_1 + C_2}_{\text{affine cipher}}$$

over $GF(2^8)$

$$\underset{00000010}{02} \cdot \underset{00000011}{B} \leftarrow 0\text{x}03$$

$$x \cdot x+1 = x^2 + x = 00000110$$



10 = 10000   11 = 10001

05 = 101   06 = 0110

$$w = 2 \cdot A + 3 \cdot B + 1 \cdot C + 1 \cdot D$$

$$= (x)(x^2+1) + (x+1)(x^4) + (x^4+1) + (x^2+x)$$

$$= (x^3+x) + (x^5+x^4) + (x^4+1) + (x^2+x)$$

$$= x^5 + x^3 + x^2 + 1$$

$$= \underbrace{0010}\underbrace{1101}$$

$$= 2D$$

AES Example - Input (128 bit key and message)

Key in English: Thats my Kung Fu (16 ASCII characters, 1 byte each)

Translation into Hex:

T	h	a	t	s		m	y		K	u	n	g		F	u
54	68	61	74	73	20	6D	79	20	4B	75	6E	67	20	46	75

Key in Hex (128 bits): 54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75

Plaintext in English: Two One Nine Two (16 ASCII characters, 1 byte each)

Translation into Hex:

T	w	o		O	n	e		N	i	n	e		T	w	o
54	77	6F	20	4F	6E	65	20	4E	69	6E	65	20	54	77	6F

Plaintext in Hex (128 bits): 54 77 6F 20 4F 6E 65 20 4E 69 6E 65 20 54 77 6F

AES Example - The first Roundkey

- Key in Hex (128 bits): 54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75
- $w[0] = (54, 68, 61, 74), w[1] = (73, 20, 6D, 79), w[2] = (20, 4B, 75, 6E), w[3] = (67, 20, 46, 75)$
- $g(w[3])$:
  - circular byte left shift of $w[3]$: $(20, 46, 75, 67)$
  - Byte Substitution (S-Box): $(B7, 5A, 9D, 85)$
  - Adding round constant $(01, 00, 00, 00)$ gives: $g(w[3]) = (B6, 5A, 9D, 85)$
- $w[4] = w[0] \oplus g(w[3]) = (E2, 32, FC, F1)$:

0101 0100	0110 1000	0110 0001	0111 0100
1011 0110	0101 1010	1001 1101	1000 0101
1110 0010	0011 0010	1111 1100	1111 0001
E2	32	FC	F1

- $w[5] = w[4] \oplus w[1] = (91, 12, 91, 88)$, $w[6] = w[5] \oplus w[2] = (B1, 59, E4, E6)$, $w[7] = w[6] \oplus w[3] = (D6, 79, A2, 93)$
- first roundkey: E2 32 FC F1 91 12 91 88 B1 59 E4 E6 D6 79 A2 93

AES Example - All RoundKeys

- Round 0: 54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75
- Round 1: E2 32 FC F1 91 12 91 88 B1 59 E4 E6 D6 79 A2 93
- Round 2: 56 08 20 07 C7 1A B1 8F 76 43 55 69 A0 3A F7 FA
- Round 3: D2 60 0D E7 15 7A BC 68 63 39 E9 01 C3 03 1E FB
- Round 4: A1 12 02 C9 B4 68 BE A1 D7 51 57 A0 14 52 49 5B
- Round 5: B1 29 3B 33 05 41 85 92 D2 10 D2 32 C6 42 9B 69
- Round 6: BD 3D C2 B7 B8 7C 47 15 6A 6C 95 27 AC 2E 0E 4E
- Round 7: CC 96 ED 16 74 EA AA 03 1E 86 3F 24 B2 A8 31 6A
- Round 8: 8E 51 EF 21 FA BB 45 22 E4 3D 7A 06 56 95 4B 6C
- Round 9: BF E2 BF 90 45 59 FA B2 A1 64 80 B4 F7 F1 CB D8
- Round 10: 28 FD DE F8 6D A4 24 4A CC C0 A4 FE 3B 31 6F 26

AES Example - Add Roundkey, Round 0

- State Matrix and Roundkey No.0 Matrix:

$$\begin{pmatrix} 54 & 4F & 4E & 20 \\ 77 & 6E & 69 & 54 \\ 6F & 65 & 6E & 77 \\ 20 & 20 & 65 & 6F \end{pmatrix} \qquad \begin{pmatrix} 54 & 73 & 20 & 67 \\ 68 & 20 & 4B & 20 \\ 61 & 6D & 75 & 46 \\ 74 & 79 & 6E & 75 \end{pmatrix}$$

- XOR the corresponding entries, e.g., $69 \oplus 4B = 22$

$$\begin{array}{c} 0110\ 1001 \\ 0100\ 1011 \\ \hline 0010\ 0010 \end{array}$$

- the new State Matrix is

$$\begin{pmatrix} 00 & 3C & 6E & 47 \\ 1F & 4E & 22 & 74 \\ 0E & 08 & 1B & 31 \\ 54 & 59 & 0B & 1A \end{pmatrix}$$

AES Example - Round 1, Substitution Bytes

- current State Matrix is

$$\begin{pmatrix} 00 & 3C & 6E & 47 \\ 1F & 4E & 22 & 74 \\ 0E & 08 & 1B & 31 \\ 54 & 59 & 0B & 1A \end{pmatrix}$$

- substitute each entry (byte) of current state matrix by corresponding entry in AES S-Box
- for instance: byte 6E is substituted by entry of S-Box in row 6 and column E, i.e., by 9F
- this leads to new State Matrix

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ C0 & 2F & 93 & 92 \\ AB & 30 & AF & C7 \\ 20 & CB & 2B & A2 \end{pmatrix}$$

- this non-linear layer is for resistance to differential and linear cryptanalysis attacks

AES Example - Round 1, Shift Row

- the current State Matrix is

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ C0 & 2F & 93 & 92 \\ AB & 30 & AF & C7 \\ 20 & CB & 2B & A2 \end{pmatrix}$$

- four rows are shifted cyclically to the left by offsets of 0,1,2, and 3
- the new State Matrix is

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix}$$

- this linear mixing step causes diffusion of the bits over multiple rounds

AES Example - Round 1, Mix Column

- Mix Column multiplies fixed matrix against current State Matrix:

$$\begin{pmatrix} 02\,03\,01\,01 \\ 01\,02\,03\,01 \\ 01\,01\,02\,03 \\ 03\,01\,01\,02 \end{pmatrix} \begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix} = \begin{pmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{pmatrix}$$

- entry BA is result of $(02 \bullet 63) \oplus (03 \bullet 2F) \oplus (01 \bullet AF) \oplus (01 \bullet A2)$:
  - $02 \bullet 63 = 00000010 \bullet 01100011 = 11000110$
  - $03 \bullet 2F = (02 \bullet 2F) \oplus 2F = (00000010 \bullet 00101111) \oplus 00101111 = 01110001$
  - $01 \bullet AF = AF = 10101111$ and $01 \bullet A2 = A2 = 10100010$
  - hence

$$\begin{array}{c} 11000110 \\ 01110001 \\ 10101111 \\ 10100010 \\ \hline 10111010 \end{array}$$

AES Example - Add Roundkey, Round 1

- State Matrix and Roundkey No.1 Matrix:

$$\begin{pmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{pmatrix} \qquad \begin{pmatrix} E2 & 91 & B1 & D6 \\ 32 & 12 & 59 & 79 \\ FC & 91 & E4 & A2 \\ F1 & 88 & E6 & 93 \end{pmatrix}$$

- XOR yields new State Matrix

$$\begin{pmatrix} 58 & 15 & 59 & CD \\ 47 & B6 & D4 & 39 \\ 08 & 1C & E2 & DF \\ 8B & BA & E8 & CE \end{pmatrix}$$

- AES output after Round 1: 58 47 08 8B 15 B6 1C BA 59 D4 E2 E8 CD 39 DF CE

AES Example - Round 2

- after Substitute Byte and after Shift Rows:

$$\begin{pmatrix} 6A & 59 & CB & BD \\ A0 & 4E & 48 & 12 \\ 30 & 9C & 98 & 9E \\ 3D & F4 & 9B & 8B \end{pmatrix} \qquad \begin{pmatrix} 6A & 59 & CB & BD \\ 4E & 48 & 12 & A0 \\ 98 & 9E & 30 & 9B \\ 8B & 3D & F4 & 9B \end{pmatrix}$$

- after Mixcolumns and after Roundkey:

$$\begin{pmatrix} 15 & C9 & 7F & 9D \\ CE & 4D & 4B & C2 \\ 89 & 71 & BE & 88 \\ 65 & 47 & 97 & CD \end{pmatrix} \qquad \begin{pmatrix} 43 & 0E & 09 & 3D \\ C6 & 57 & 08 & F8 \\ A9 & C0 & EB & 7F \\ 62 & C8 & FE & 37 \end{pmatrix}$$

AES Example - Round 3

- after Substitute Byte and after Shift Rows:

$$\begin{pmatrix} 1A & AB & 01 & 27 \\ B4 & 5B & 30 & 41 \\ D3 & BA & E9 & D2 \\ AA & E8 & BB & 9A \end{pmatrix} \qquad \begin{pmatrix} 1A & AB & 01 & 27 \\ 5B & 30 & 41 & B4 \\ E9 & D2 & D3 & BA \\ A9 & AA & E8 & BB \end{pmatrix}$$

- after Mixcolumns and after Roundkey:

$$\begin{pmatrix} AA & 65 & FA & 88 \\ 16 & 0C & 05 & 3A \\ 3D & C1 & DE & 2A \\ B3 & 4B & 5A & 0A \end{pmatrix} \qquad \begin{pmatrix} 78 & 70 & 99 & 4B \\ 76 & 76 & 3C & 39 \\ 30 & 7D & 37 & 34 \\ 54 & 23 & 5B & F1 \end{pmatrix}$$

AES Example - Round 4

- after Substitute Byte and after Shift Rows:

$$\begin{pmatrix} BC & 51 & EE & B3 \\ 38 & 38 & EB & 12 \\ 04 & FF & 9A & 18 \\ 20 & 26 & 39 & A1 \end{pmatrix} \qquad \begin{pmatrix} BC & 51 & EE & B3 \\ 38 & EB & 12 & 38 \\ 9A & 18 & 04 & FF \\ A1 & 20 & 26 & 39 \end{pmatrix}$$

- after Mixcolumns and after Roundkey:

$$\begin{pmatrix} 10 & BC & D3 & F3 \\ D8 & 94 & E0 & E0 \\ 53 & EA & 9E & 25 \\ 24 & 40 & 73 & 7B \end{pmatrix} \qquad \begin{pmatrix} B1 & 08 & 04 & E7 \\ CA & FC & B1 & B2 \\ 51 & 54 & C9 & 6C \\ ED & E1 & D3 & 20 \end{pmatrix}$$

AES Example - Round 5

- after Substitute Byte and after Shift Rows:

$$\begin{pmatrix} C8 & 30 & F2 & 94 \\ 74 & B0 & C8 & 37 \\ D1 & 20 & DD & 50 \\ 55 & F8 & 66 & B7 \end{pmatrix} \qquad \begin{pmatrix} C8 & 30 & F2 & 94 \\ B0 & C8 & 37 & 74 \\ DD & 50 & D1 & 20 \\ B7 & 55 & F8 & 66 \end{pmatrix}$$

- after Mixcolumns and after Roundkey:

$$\begin{pmatrix} 2A & 26 & 8F & E9 \\ 78 & 1E & 0C & 7A \\ 1B & A7 & 6F & 0A \\ 5B & 62 & 00 & 3F \end{pmatrix} \qquad \begin{pmatrix} 9B & 23 & 5D & 2F \\ 51 & 5F & 1C & 38 \\ 20 & 22 & BD & 91 \\ 68 & F0 & 32 & 56 \end{pmatrix}$$

AES Example - Round 6

- after Substitute Byte and after Shift Rows:

$$\begin{pmatrix} 14 & 26 & 4C & 15 \\ D1 & CF & 9C & 07 \\ B7 & 93 & 7A & 81 \\ 45 & 8C & 23 & B1 \end{pmatrix} \qquad \begin{pmatrix} 14 & 26 & 4C & 15 \\ CF & 9C & 07 & D1 \\ 7A & 81 & B7 & 93 \\ B1 & 45 & 8C & 23 \end{pmatrix}$$

- after Mixcolumns and after Roundkey:

$$\begin{pmatrix} A9 & 37 & AA & F2 \\ AE & D8 & 0C & 21 \\ E7 & 6C & B1 & 9C \\ F0 & FD & 67 & 3B \end{pmatrix} \qquad \begin{pmatrix} 14 & 8F & C0 & 5E \\ 93 & A4 & 60 & 0F \\ 25 & 2B & 24 & 92 \\ 77 & E8 & 40 & 75 \end{pmatrix}$$

AES Example - Round 7

- after Substitute Byte and after Shift Rows:

$$\begin{pmatrix} FA & 73 & BA & 58 \\ DC & 49 & D0 & 76 \\ 3F & F1 & 36 & 4F \\ F5 & 9B & 09 & 9D \end{pmatrix} \qquad \begin{pmatrix} FA & 73 & BA & 58 \\ 49 & D0 & 76 & DC \\ 36 & 4F & 3F & F1 \\ 9D & F5 & 9B & 09 \end{pmatrix}$$

- after Mixcolumns and after Roundkey:

$$\begin{pmatrix} 9F & 37 & 51 & 37 \\ AF & EC & 8C & FA \\ 63 & 39 & 04 & 66 \\ 4B & FB & B1 & D7 \end{pmatrix} \qquad \begin{pmatrix} 53 & 43 & 4F & 85 \\ 39 & 06 & 0A & 52 \\ 8E & 93 & 3B & 57 \\ 5D & F8 & 95 & BD \end{pmatrix}$$

AES Example - Round 8

- after Substitute Byte and after Shift Rows:

$$\begin{pmatrix} ED & 1A & 84 & 97 \\ 12 & 6F & 67 & 00 \\ 19 & DC & E2 & 5B \\ 4C & 41 & 2A & 7A \end{pmatrix} \qquad \begin{pmatrix} ED & 1A & 84 & 97 \\ 6F & 67 & 00 & 12 \\ E2 & 5B & 19 & DC \\ 7A & 4C & 41 & 2A \end{pmatrix}$$

- after Mixcolumns and after Roundkey:

$$\begin{pmatrix} E8 & 8A & 4B & F5 \\ 74 & 75 & EE & E6 \\ D3 & 1F & 75 & 58 \\ 55 & 8A & 0C & 38 \end{pmatrix} \qquad \begin{pmatrix} 66 & 70 & AF & A3 \\ 25 & CE & D3 & 73 \\ 3C & 5A & 0F & 13 \\ 74 & A8 & 0A & 54 \end{pmatrix}$$

AES Example - Round 9

- after Substitute Byte and after Shift Rows:

$$\begin{pmatrix} 33 & 51 & 79 & 0A \\ 3F & 8B & 66 & 8F \\ EB & BE & 76 & 7D \\ 92 & C2 & 67 & 20 \end{pmatrix} \qquad \begin{pmatrix} 33 & 51 & 79 & 0A \\ 8B & 66 & 8F & 3F \\ 76 & 7D & EB & BE \\ 20 & 92 & C2 & 67 \end{pmatrix}$$

- after Mixcolumns and after Roundkey:

$$\begin{pmatrix} B6 & E7 & 51 & 8C \\ 84 & 88 & 98 & CA \\ 34 & 60 & 66 & FB \\ E8 & D7 & 70 & 51 \end{pmatrix} \qquad \begin{pmatrix} 09 & A2 & F0 & 7B \\ 66 & D1 & FC & 3B \\ 8B & 9A & E6 & 30 \\ 78 & 65 & C4 & 89 \end{pmatrix}$$

AES Example - Round 10

- after Substitute Byte and after Shift Rows:

$$\begin{pmatrix} 01 & 3A & 8C & 21 \\ 33 & 3E & B0 & E2 \\ 3D & B8 & 8E & 04 \\ BC & 4D & 1C & A7 \end{pmatrix} \qquad \begin{pmatrix} 01 & 3A & 8C & 21 \\ 3E & B0 & E2 & 33 \\ 8E & 04 & 3D & B8 \\ A7 & BC & 4D & 1C \end{pmatrix}$$

- after Roundkey (Attention: no Mix columns in last round):

$$\begin{pmatrix} 29 & 57 & 40 & 1A \\ C3 & 14 & 22 & 02 \\ 50 & 20 & 99 & D7 \\ 5F & F6 & B3 & 3A \end{pmatrix}$$

- ciphertext: 29 C3 50 5F 57 14 20 F6 40 22 99 B3 1A 02 D7 3A