SJSU SAN JOSÉ STATE UNIVERSITY

Lesson 3 – Block Ciphers 1

Yan Chen

CS166 Fall 2024

Stream Cipher Overview

- Stream ciphers uses the idea of “one-time pad”
  - “Stretch” a small key to a long keystream (any size)
  - The keystream is used to encrypt/decrypt like a one-time pad
- Algorithm to generate keystream from the key is the “heart” of each stream cipher
  - Keystream is pseudo-random (not truly random) and may repeat (important to know the upper bound)
- Stream ciphers are efficient in hardware
  - So, it was the king of crypto...
  - But not anymore since more things can be done in software

… Previously
Overview
Feistel Cipher
DES
AES
TEA
Next Lesson ...
Appendix

A5/1 vs. RC4

	A5/1	RC4
Type	Symmetric key crypto – stream cipher	
Unit Of Operations	Bit	Byte (8 bits)
Implementation Type	Hardware	Software
Input Key Length	64 bits	Vary from 1 to 255 bytes
Main Data Structure	3 LFSRs	Self-modifying lookup table
Main Operation to Generate Keystream	Step each LFSR	Swap elements in lookup table
Satisfied Kerckhoffs' Principle	Originally not	Yes?
Status Of Lifecycle	Almost dead	Almost dead
Applications	GSM	SSL, WEP, WPA

- Recall: Block ciphers uses the idea of “codebook”
  - “Electronic” version of codebook
  - Each block of plaintext has a corresponding block of ciphertext
- Solve the cons: use a “changeable” codebook instead of a “fixed” one –key is used to generate codebooks!
  - vs. stream cipher uses key to generate one-time pads
- Change key = switch the codebook
  - If the key is k bits, then the block cipher can be viewed as 2^k different codebooks
  - So, can avoid classical codebook attack by changing the key

- Taxonomy: Block ciphers are symmetric-key ciphers
  - ➢ Usually implemented in software
- Block ciphers use a round function F to encrypt the plaintext to ciphertext
  - ➢ F is iterated many "rounds"
  - ➢ For each round, input is key and output of previous round
- Most of the block ciphers follows Feistel cipher principle
- Algorithms to be covered
  - ➢ (This lesson – each block) DES, AES, TEA
  - ➢ (Next lesson – multiple blocks) Block cipher modes

… Previously
Overview
Feistel Cipher
DES
AES
TEA
Next Lesson ...
Appendix

- Plaintext and ciphertext consist of fixed-sized blocks
- P = plaintext block, C = ciphertext block
- **E**(P, K): **E**ncrypt P with key K to get C
  - ➢ That is, E(P, K) = C
- **D**(C, K): **D**ecrypt C with key K to get P
  - ➢ That is, D(C, K) = P
- Note that P = D(E(P, K), K) and C = E(D(C, K), K)
  - ➢ Since block ciphers are symmetric key ciphers!
  - What if we used different keys?

    That is, $P \stackrel{?}{=} D(E(P, K1), K2)$ or $C \stackrel{?}{=} E(D(C, K1), K2)$?

- Feistel cipher: general block cipher design principle
  - ➢ A type of block cipher, not a specific block cipher
  - ➢ Named after German-American cryptanalyst, Horst Feistel, who works in IBM led to designing DES cipher
- In Feistel's terminology, in each round, he suggested to use two operations ...
  - ➢ Substitution (mainly for confusion): each plaintext block is uniquely replaced by a ciphertext block
  - ➢ Permutation (mainly for diffusion): a sequence of block elements are replaced by a permutation of that sequence.

- To encrypt each plaintext block...
  - Split the block into left and right halves: $P = (L_0, R_0)$
  - For each round $i = 1, 2, ..., n$, compute

    $L_i = R_{i-1}$ (new left = old right)

    $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$ (new right = old left $\oplus$ round F result)

    where K_i is called "subkey"
  - After n rounds, ciphertext block $C = (L_n, R_n)$
- Permutation is the operation that produces L_i
  - Since a "swap" is involved
- Substitution is the operation that produces R_i

… Previously
Overview
Feistel Cipher
DES
AES
TEA
Next Lesson ...
Appendix

- To decrypt, just "inverse" the encryption
  - ➢ Start with ciphertext block $C = (L_n, R_n)$
  - ➢ For each round $i = n, n-1, \ldots, 1$, compute

    $R_{i-1} = L_i$

    $L_{i-1} = R_i \oplus F(R_{i-1}, K_i)$, where F is round function, K_i is subkey

    After n rounds, plaintext block $P = (L_0, R_0)$
- For Feistel ciphers, encrypt/decrypt is always invertible
  - ➢ Even if the round function F is NOT invertible
  - ➢ Since the swap is invertible...
  - ➢ And XOR $\oplus$ is also invertible!

- Feistel cipher principle involves several parameters
- Block size
  - ➢ Larger size → greater security but slower algorithm
  - ➢ Typical size is 128-bit
- Key length
  - ➢ longer length → greater security but slower algorithm
  - ➢ Typical length is at least 128-bit
- Number of rounds: typical number is 16
- Round function F
  - ➢ The greater complexity, the greater the security

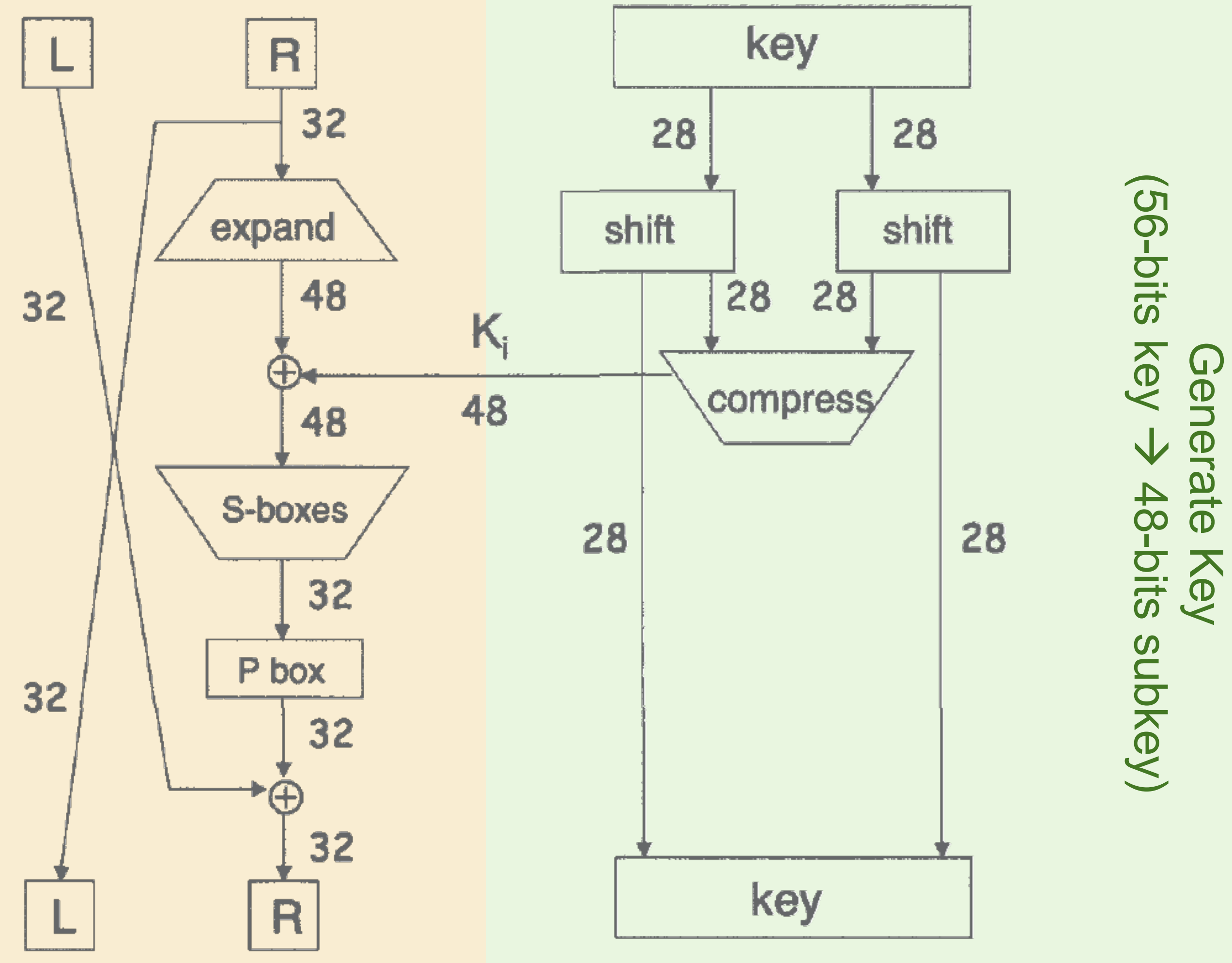
… Previously
Overview
Feistel Cipher
DES
AES
TEA
Next Lesson ...
Appendix

- DES (Data Encryption Standard) is a Feistel cipher
  - ➢ Developed in 1970's, based on IBM's Lucifer cipher
  - ➢ Was U.S. government standard
  - ➢ NSA (National Security Agency) was secretly involved to approve/modify it
- DES numerology
  - ➢ Block size: 64 bits
  - ➢ Key length: 56 bits (subkey length in each round = 48 bits)
  - ➢ 16 rounds using a simple round function (for a block cipher)
  - ➢ Round function uses 8 "S-boxes", each maps 6 bits to 4 bits

… Previously
Overview
Feistel Cipher
DES
AES
TEA
Next Lesson ...
Appendix

- In each round...(taken from “textbook”)

Before (i - 1)



$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

After (i)

… Previously
Overview
Feistel Cipher
DES
AES
TEA
Next Lesson ...
Appendix

- DES includes 8 substitution boxes (“S-box”)
  - Each S-box maps 6 bits to 4 bits
  - So, 8 * 6 = 48 bits (subkey) → 8 * 4 = 32 bits (half block)
  - Row address: bits 0 and 5
  - Column address: bits 1 through 4
- Example: S-box number 1

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111
01	0000	1111	0111	0100	1110	0010	1101	0001	1010	0110	1100	1011	1001	0101	0011	1000
10	0100	0001	1110	1000	1101	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101
11	1111	1100	1000	0010	0100	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101

  - Input: 101001
  - Output: 0100

- Security of DES
  - Depends heavily on S-boxes
  - Everything else in DES is linear
  - 35+ years of intense analysis has revealed no back door
  - All known attacks were essentially exhaustive key search
- But today, 56 bit DES key is too small
  - Exhaustive key search is feasible
- So, Triple DES or 3DES (112 bit key)
  - $C = E(D(E(P, K_1), K_2),K_1)$ = encrypt-decrypt-encrypt
  - $P = D(E(D(C, K_1), K_2),K_1)$ = decrypt-encrypt-decrypt

- Why not "doubled"?
  - ➢ Like C = E(E(P, K), K)
  - ➢ Still the same key = still 56 bits
- Why "tripled" with 2 keys?
  - ➢ For backward compatible: when $K_1 = K_2 = K$ same as regular DES: E(D(E(P, K), K), K) = E(P, K)
  - ➢ And 56 * 2 =112 bits key is large enough

💡Why not doubled with 2 keys like $C = E(E(P, K_1), K_2)$

… Previously
Overview
Feistel Cipher
DES
AES
TEA
Next Lesson ...
Appendix

- AES (Advanced Encryption Standard) background
  - NIST (National Institute of Standards and Technology) issued a call for proposals in 1990 to replace DES
  - NSA was openly involved
  - Based on “Rijndael” Algorithm
- AES is very important in cryptography
  - An AES instruction set is integrated into many processors
  - Some libraries for high level languages such as Java, etc.
  - Part of numerous open standards such as IPsec or TLS
  - Mandatory for US government applications

- AES has a complex (highly mathematical) structure
  - ➢ Will only cover a high-level overview of the structure
- AES numerology
  - ➢ Block size: 128 bits (or 192 or 256 bits based on the type)
  - ➢ Each block is treated as 4 x 4 bytematrix (4 * 4 * 8 = 128 bits)
  - ➢ Key length: 128, 192 or 256 bits (all for AES-128)
  - ➢ 10 to 14 rounds (depends on key length)
  - ➢ Involved 3 layers: nonlinear, linear mixing & key addition
  - ➢ 4 functions: ByteSub, ShiftRow, MixColumn, AddRoundKey

Is AES a Feistel cipher?

… Previously
Overview
Feistel Cipher
DES
AES
TEA
Next Lesson ...
Appendix

- ByteSub: substitute each byte of the 4 * 4 matrix
  - b_{ij} = ByteSub(a_{ij})

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

  - Nonlinear but invertible composition of two math operations
  - Or just view it as a lookup table
  - ByteSub is AES's "S-box" (only 1 though)
  - Primarily confusion or diffusion?

- Linear mixing layer includes 2 functions
  - MixColumn and ShiftRow
- MixColumn: multiply a constant 4 * 4 matrix
  - That is, operation on each column

$$\begin{bmatrix} a_{0i} \\ a_{1i} \\ a_{2i} \\ a_{3i} \end{bmatrix} \longrightarrow \texttt{MixColumn} \longrightarrow \begin{bmatrix} b_{0i} \\ b_{1i} \\ b_{2i} \\ b_{3i} \end{bmatrix} \quad \text{for } i = 0, 1, 2, 3.$$

  - Linear and invertible (result multiply another 4 * 4 matrix)
  - Implemented as a (big) lookup table
  - Primarily confusion or diffusion?

- ShiftRow: cyclic shift (left rotation) in each row

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \longrightarrow \texttt{ShiftRow} \longrightarrow \begin{array}{c} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{11} & a_{12} & a_{13} & a_{10} \\ a_{22} & a_{23} & a_{20} & a_{21} \\ a_{33} & a_{30} & a_{31} & a_{32} \end{bmatrix} \end{array} \begin{array}{l} \text{no change} \\ \text{shift by 1} \\ \text{shift by 2} \\ \text{shift by 3} \end{array}$$

- The rotation is at byte level
- Linear and invertible
- To inverse, simply rotate same numbers to the right
- Primarily confusion or diffusion?

… Previously
Overview
Feistel Cipher
DES
AES
TEA
Next Lesson ...
Appendix

- Key addition layer includes 1 function
- AddRoundKey: XOR subkey with the block

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

- $b_{ij} = a_{ij} \oplus k_{ij}$
- Is it invertible? How?
- Primarily confusion or diffusion?

- TEA (Tiny Encryption Algorithm)
  - Invented in 1994, public domain now
  - Designed to replace DES
  - A lightweight and simple algorithm (simpler round function)
  - So, performance is impressive (fast, low memory requirement)
- TEA numerology
  - Block size: 64 bits
  - Key length: 128 bits
  - Number of rounds varies and 32 is considered secure
  - Why need more rounds than DES?

- Assuming 32 rounds:

```
(K[0], K[1], K[2], K[3]) = 128 bit key
(L, R) = plaintext (64-bit block)
delta = 0x9e3779b9
sum = 0
for i = 1 to 32
    sum += delta
    L += ((R ‹‹ 4) + K[0]) ⊕ (R + sum) ⊕ ((R ›› 5) + K[1])
    R += ((L ‹‹ 4) + K[2]) ⊕ (L + sum) ⊕ ((L ›› 5) + K[3])
next i
ciphertext = (L, R)
```

➢ ‹‹ means left (non-cyclic) shift, ›› means right (non-cyclic) shift

Based on the algorithm, is TEA a Feistel cipher?

- To decrypt, “inverse” the encryption
- Assuming 32 rounds:

```
(K[0], K[1], K[2], K[3]) = 128 bit key
(L, R) = ciphertext (64-bit block)
delta = 0x9e3779b9
sum ‹‹ 5
for i = 1 to 32
    R -= ((L ‹‹ 4) + K[2]) ⊕ (L + sum) ⊕ ((L ›› 5) + K[3])
    L -= ((R ‹‹ 4) + K[0]) ⊕ (R + sum) ⊕ ((R ›› 5) + K[1])
    sum -= delta
next i
ciphertext = (L, R)
```

- Block cipher modes
  - ECB, CBC, CTR
- Uses for symmetric crypto
  - Confidentiality
  - MAC

- Block cipher
- Feistel cipher
  - ➢ Substitution vs. permutation
  - ➢ Parameters: block size, key length, number of rounds, round function
- DES
  - ➢ S-box
  - ➢ 3DES
- AES
  - ➢ 3 layers: nonlinear, linear mixing & key addition
- 4 functions: ByteSub, ShiftRow, MixColumn, AddRoundKey
- TEA

… Previously
Overview
Feistel Cipher
DES
AES
TEA
Next Lesson ...
Appendix

- Consider a Feistel cipher with 3 rounds. For each of the following round function, represent C in terms of L_0, R_0, and subkeys K_i for i from 1 to 8
  - $F(R_{i-1}, K_i) = 0$
  - $F(R_{i-1}, K_i) = R_{i-1}$
  - $F(R_{i-1}, K_i) = K_i$
- Within a single round, DES employs both confusion and diffusion. Given one source of each.
- Which layer(s) in AES are primarily for confusion?
- Draw a diagram to illustrate the round function of TEA (you can use the one for DES on page 12 as a "template")

Lesson 3

Block Ciphers 1

… Previously

Overview

Feistel Cipher

DES

AES

TEA

Next Lesson ...

Appendix

References

- Stamp, Mark and Low, Richard M., “Applied Cryptanalysis: breaking ciphers in the real world,” John Wiley & Sons, Inc., New Jersey, USA, 2007
- Stallings, William, “Cryptography and Network Security, Principles and Practice, 6th ed.,” Pearson, USA, 2014
- Paar, Christof, “Understanding Cryptography,” Faller, Berlin, Germany, 2010
- Stamp, Mark, “Information Security, Principles and Practice, 2nd ed.,” Wiley, New Jersey, USA, 2011