CIS551: Computer and Network Security

Jonathan M. Smith
jms@cis.upenn.edu

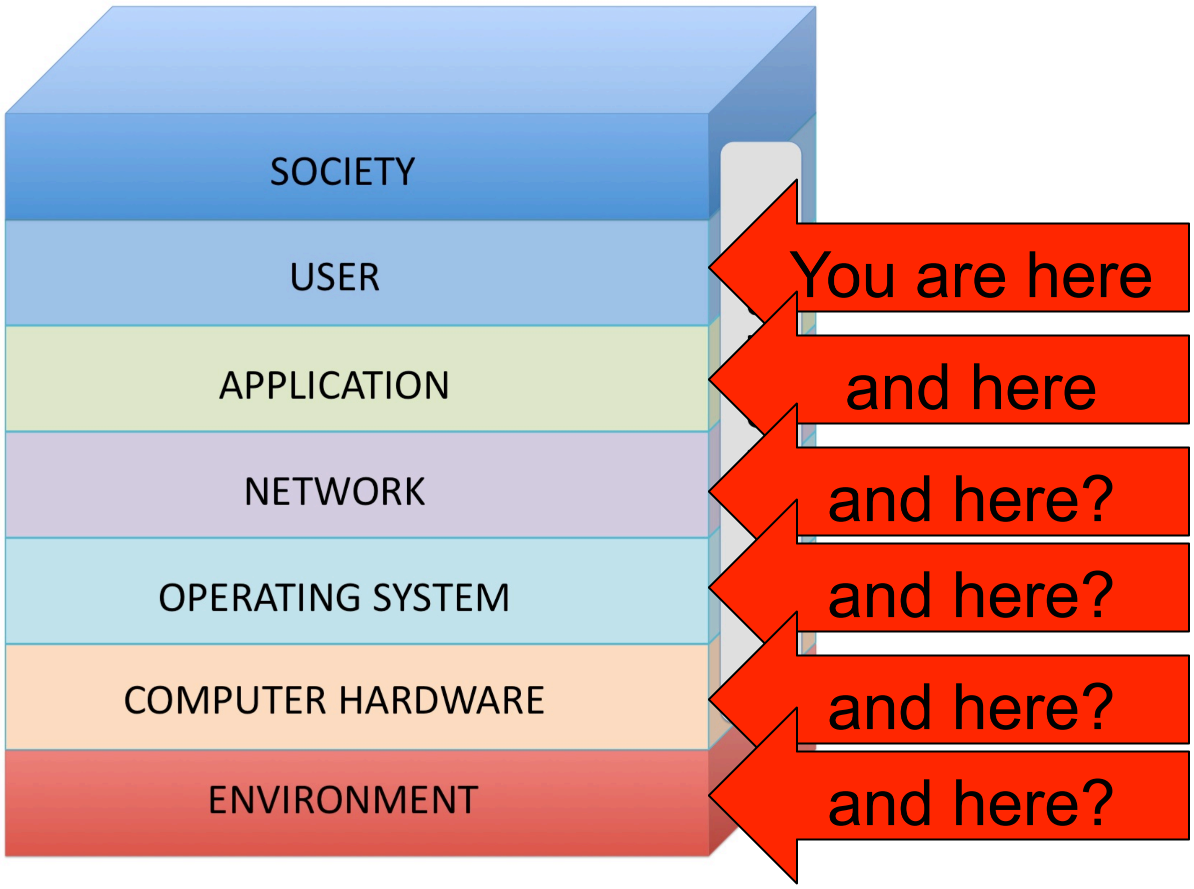
03/17/2014

Uses material from S. Zdancewic/C. Gunter

CIS551 Topics

- Computer Security
  - Software/Languages, Computer Arch.
  - Access Control, Operating Systems
  - Threats: Vulnerabilities, Viruses
- Computer Networks
  - Physical layers, Internet, WWW, Applications
  - Cryptography in several forms
  - Threats: Confidentiality, Integrity, Availability
- Systems Viewpoint
  - Users, social engineering, insider threats

Sincoskie NIS model



W.D. Sincoskie, *et al.* “Layer Dissonance and Closure in Networked Information Security” (white paper)

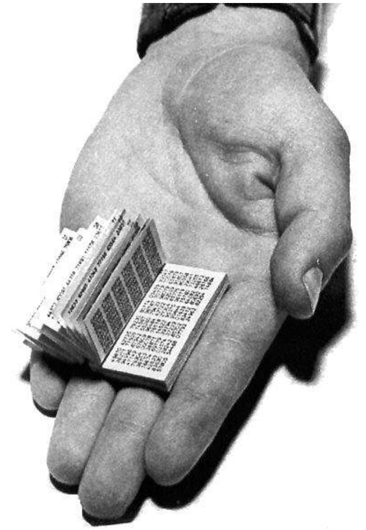
Uses material from S. Zdancewic/C. Gunter

Perfect Substitution Ciphers

$$
\begin{array}{r l}
 & p_1\ p_2\ p_3\ \ldots\ p_n \\
\oplus & b_1\ b_2\ b_3\ \ldots\ b_n \\
\hline
 & c_1\ c_2\ c_3\ \ldots\ c_n
\end{array}
$$

- Choose a string of *random* bits $b_1 \ldots b_n$ the same length as the plaintext $p_1 \ldots p_n$, XOR them to obtain the ciphertext $c_1 \ldots c_n$.
- Perfect Secrecy
  - Probability that a given message is encoded in the ciphertext is unaltered by knowledge of the ciphertext
  - Proof: Give me any plaintext message, and any ciphertext, and I can construct a key that will produce the ciphertext from the plaintext.

One-time Pads



- Another name for Perfect Substitution
- Actually used
  – Physical pad of paper
  – List of random numbers
  – Pages were torn out and destroyed after use
  – "Numbers Stations" (see Wikipedia)
- Vernam Cipher
  – Used by AT&T
  – Random sequence stored on punch tape
- Not practical for general purpose cryptography
  – But useful as component in other protocols.

Uses material from S. Zdancewic/C. Gunter

Problems with "Perfect" Substitution

- Key is the same length as the plaintext
  - Sender and receiver must agree on the same random sequence
  - Not any easier to transmit key securely than to transmit plaintext securely
- Need to be able to generate many truly random bits
  - Pseudorandom numbers generated by an algorithm aren't good enough for long messages
- Can't reuse the key
  - Not enough confusion

Diffusion and Confusion

- Diffusion
  - Ciphertext should look random
  - Protection against statistical attacks
  - Monoalphabetic -> Polyalphabetic substitution; diffusion increases
- Confusion
  - Make the relation between the key, plaintext and ciphertext complex
  - Lots of confusion -> hard to calculate key in a known plaintext attack
  - Polyalphabetic substitution: little confusion

Computational Security

- Perfect Ciphers are *unconditionally secure*
  - *No* amount of computation will help crack the cipher (i.e., the *only* strategy is brute force)
- In practice, strive for *computational security*
  - Given enough power, the attacker could crack the cipher (example: brute force attack)
  - But, an attacker with only *bounded resources* is extremely unlikely to crack it
  - Example: Assume attacker has only *polynomial time*, then encryption algorithm that can't be inverted in less than exponential time is secure.
  - Results are usually stated *probabilistically*

Uses material from S. Zdancewic/C. Gunter

Kinds of Industrial Strength Crypto

- Shared Key Cryptography
- Public Key Cryptography
- Cryptographic Hashes
- All of these aim for computational security
  - Not all methods have been *proven* to be intractable to crack.

Shared Key Cryptography

- Sender & receiver use the <u>same</u> key
- Key <u>must</u> remain <u>private</u>
- Also called *symmetric* or *secret key* cryptography
- Often are *block-ciphers*
  - Process plaintext data in blocks
- Examples: DES, Triple-DES, Blowfish, Twofish, AES, Rijndael, …

Shared Key Notation

- Encryption algorithm

  $E : key \times plain \rightarrow cipher$

  Notation: $K\{msg\} = E(K, msg)$
- Decryption algorithm

  $D : key \times cipher \rightarrow plain$
- D inverts E

  $D(K, E(K, msg)) = msg$
- Use capital "K" for shared (secret) keys
- Sometimes E is the same algorithm as D

Secure Channel: Shared Keys

Alice

Bart

K_{AB}\{Hello!\} →

← K_{AB}\{Hi!\}

K_{AB}

K_{AB}

Uses material from S. Zdancewic/C. Gunter

Data Encryption Standard (DES)

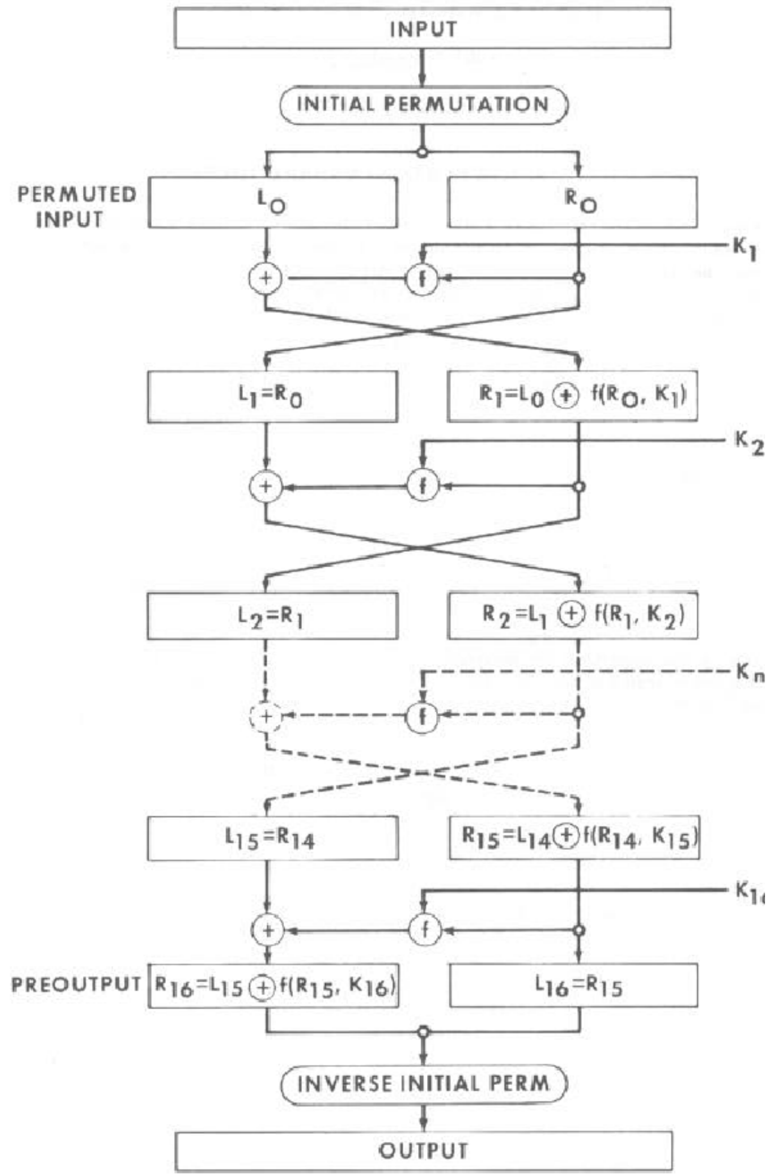
- Adopted as a standard in 1976
- Security analyzed by the National Security Agency (NSA)
  - http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
- Key length is 56 bits
  - padded to 64 bits by using 8 parity bits
- Uses simple operators on (up to) 64 bit values
  - Simple to implement in software or hardware

- Input is processed in 64 bit blocks
- Based on a series of 16 *rounds*
  - Each cycle uses permutation & substitution to combine plaintext with the key

DES Encryption

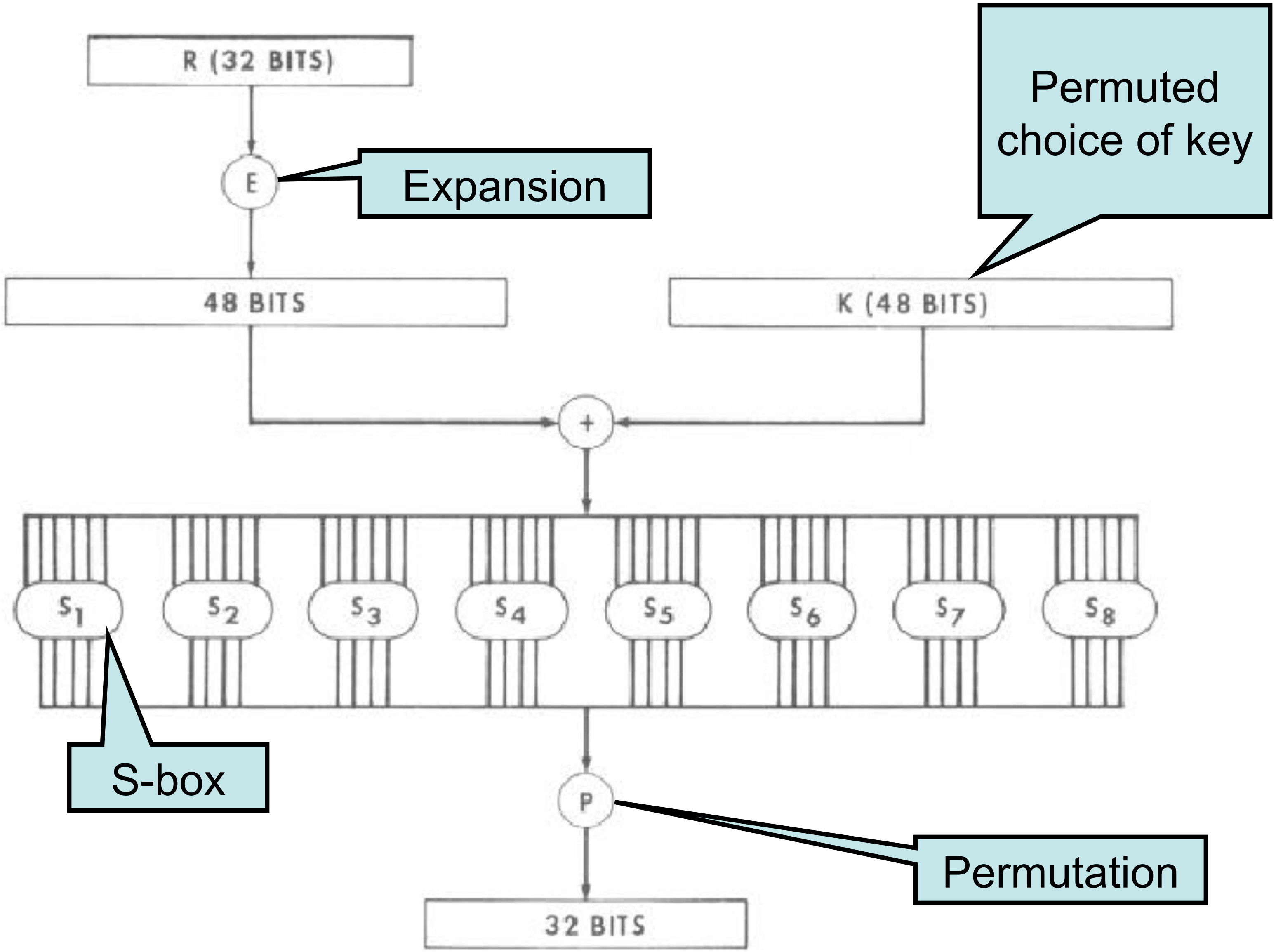
DES follows the structure of
a general class of encryption agorithms called *Feistel* ciphers:

- *Rounds* of encryption
- Each round merges one half (L) of the input with the other (R)



Uses material from S. Zdancewic/C. Gunter

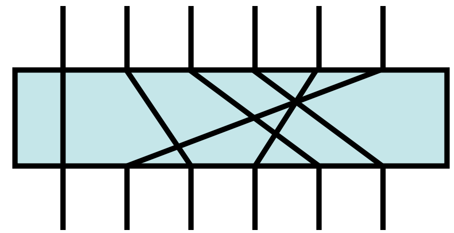
One Round of DES (f of previous slide)



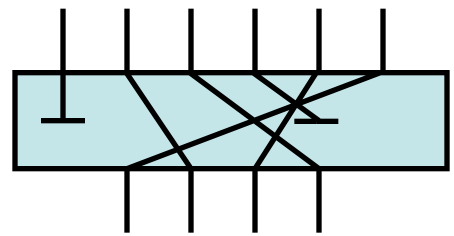
Uses material from S. Zdancewic/C. Gunter

Types of Permutations in DES

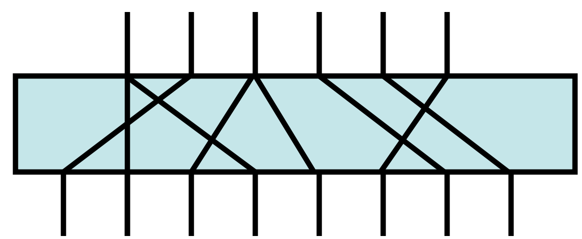
Permutation



Permuted Choice



Expansion Permutation



Uses material from S. Zdancewic/C. Gunter

DES S-Boxes (Substitution tables)

- 6 bits of input replaced by 4 bits of output
- Implemented as a lookup table
  - 8 S-Boxes
  - Each S-Box has a table of 64 entries
  - Each entry specifies a 4-bit output
- S-Box design is complex: here is the “black art” of cryptography design.
- Example desiderata:
  - No output of any S-Box should be close to a linear function of the inputs.
  - If two inputs to an S-Box differ by exactly one bit, the outputs must differ in at least two bits.
  - Each row of an S-Box should contain all 16 possible bit combinations
  - …

DES Decryption

- Use the same algorithm as encryption, but use $k_{16} \ldots k_1$ instead of $k_1 \ldots k_{16}$
- Proof that this works:
  - To obtain round j from j-1:

$$(1) \quad L_j = R_{j-1}$$
$$(2) \quad R_j = L_{j-1} \oplus f(R_{j-1}, k_j)$$

  - Rewrite in terms of round j-1:

$$(1) \quad R_{j-1} = L_j$$
$$(2) \quad L_{j-1} \oplus f(R_{j-1}, k_j) = R_j$$
$$L_{j-1} \oplus f(R_{j-1}, k_j) \oplus f(R_{j-1}, k_j) = R_j \oplus f(R_{j-1}, k_j)$$
$$L_{j-1} = R_j \oplus f(R_{j-1}, k_j)$$
$$L_{j-1} = R_j \oplus f(L_j, k_j)$$

Uses material from S. Zdancewic/C. Gunter

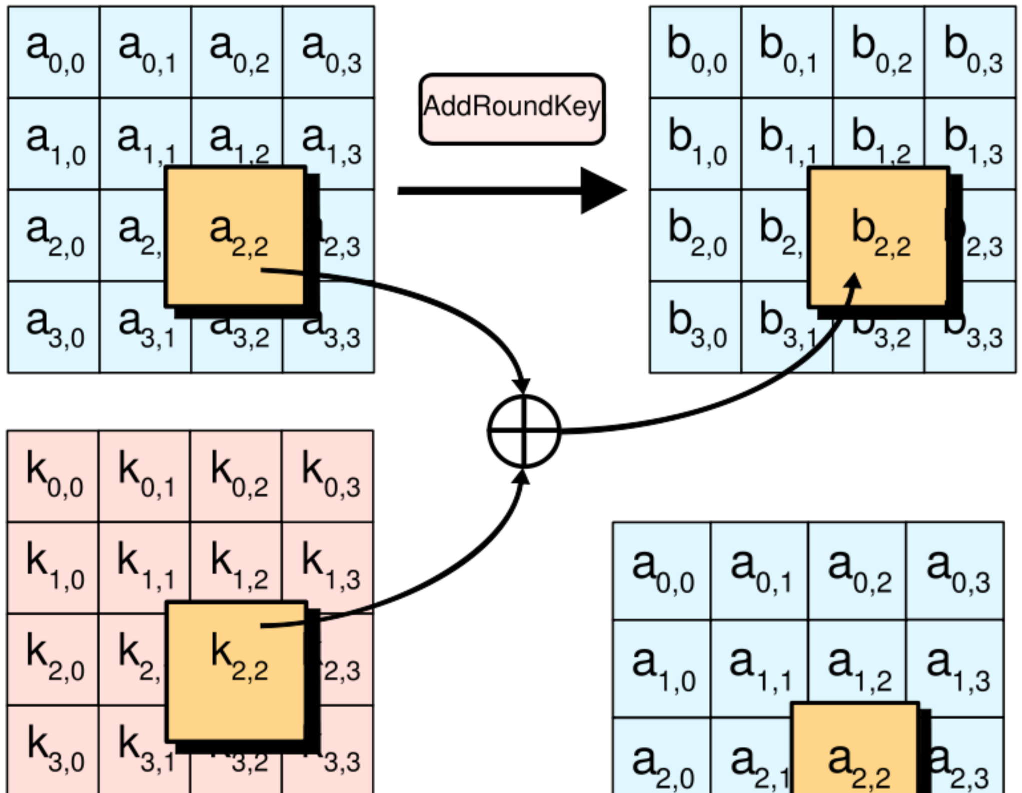
Problems with DES

- Key length too short: 56 bits
  - www.distributed.net broke a DES challenge in 1999 in under 24 hours (parallel attack)
- Other problems
  - Bit-wise complementation of key produces bit-wise complemented ciphertext
  - Not all keys are good (half 0's half 1's)
  - Differential cryptanalysis (1990): Carefully choose pairs of plaintext that differ in particular known ways (e.g. they are complements)
    - But particular choice of S boxes is secure against this (!) (developers of DES knew about differential cryptanalysis before it was "publically" known in the research community)

Advanced Encryption Standard (AES)

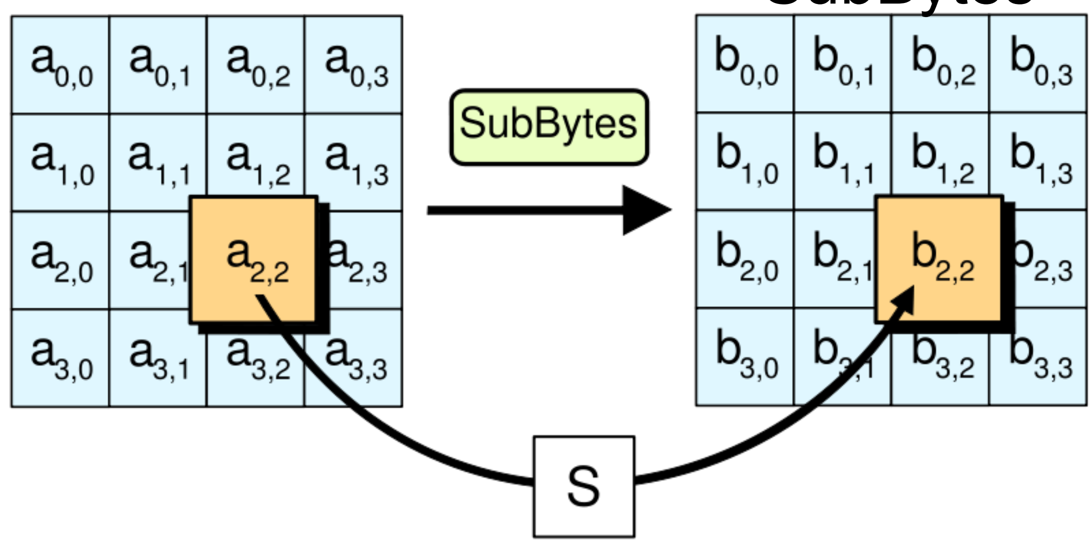
- National Institute of Standards & Technology NIST
  - Computer Security Research Center (CSRC)
  - http://csrc.nist.gov/
- Uses the Rijndael algorithm
  - Invented by Belgian researchers
    Dr. Joan Daemen & Dr. Vincent Rijmen
  - Adopted May 26, 2002
  - Key length: 128, 192, or 256 bits
  - Block size: 128, 192, or 256 bits
- Not a Feistel cipher
  - 10 rounds, each consisting of: SubBytes / Shift Rows / Mix Columns / Add Round Key

AES Operations



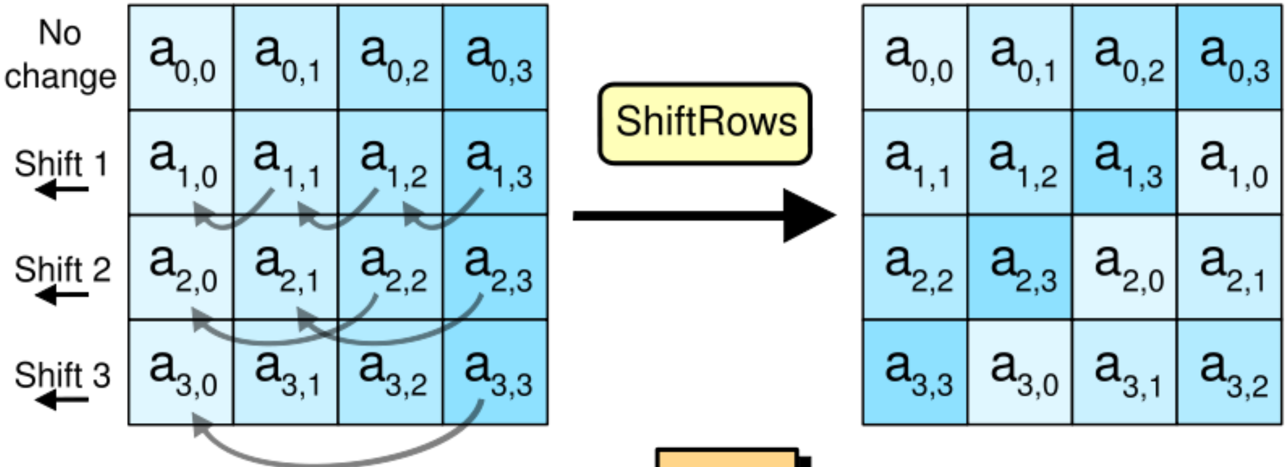
Add Round Key

SubBytes



Uses material from S. Zdancewic/C. Gunter

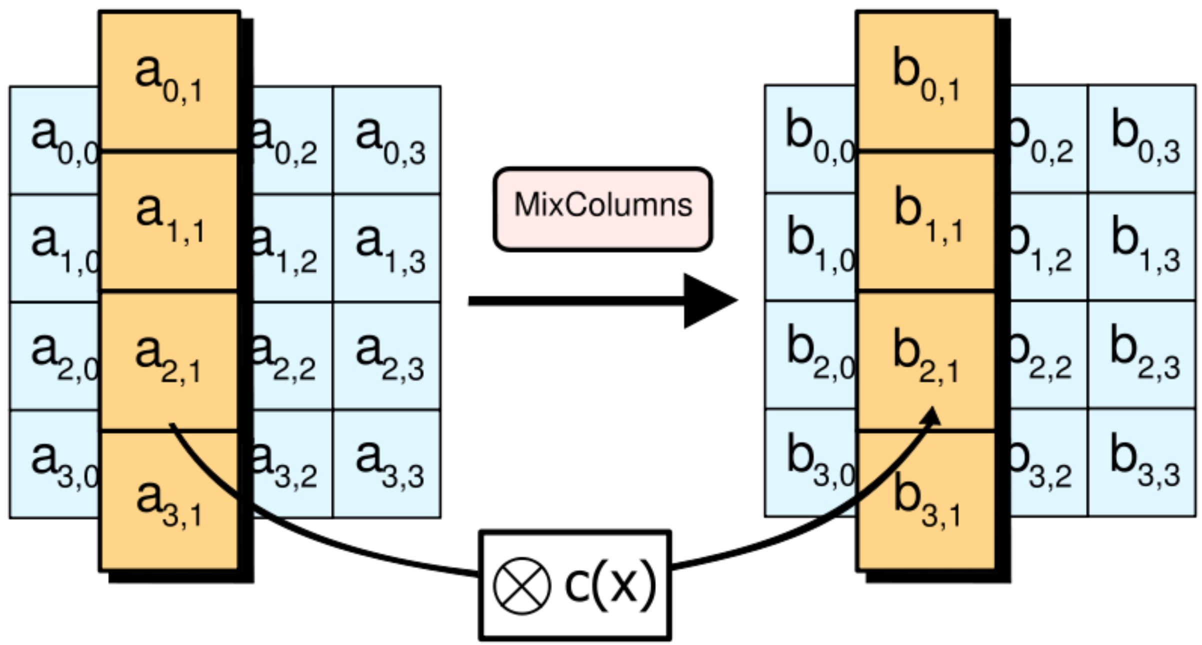
AES Operations



ShiftRows

MixColumns

$$c(x) = 3x^3 + x^2 + x + 2$$



Uses material from S. Zdancewic/C. Gunter

Block Cipher Modes of Operation

- Often want to encrypt large pieces of data, but block ciphers only work on fixed, small chunks.
- Various Options:
  - *Electronic Code Book* – each block of plaintext bits is encoded independently using the same key:

    $C_j = K\{P_j\}$

  - *Cipher Block Chaining* – each block of plaintext is XORed with the preceding block of ciphertext, starting with initialization vector C_0:

    $C_j = K\{P_j \oplus C_{j-1}\}$ and C_0 is an initialization vector

  - Other options: Cipher Feedback (to convert a block cipher to a streaming cipher), Counter mode (XOR an encryption counter with each block)