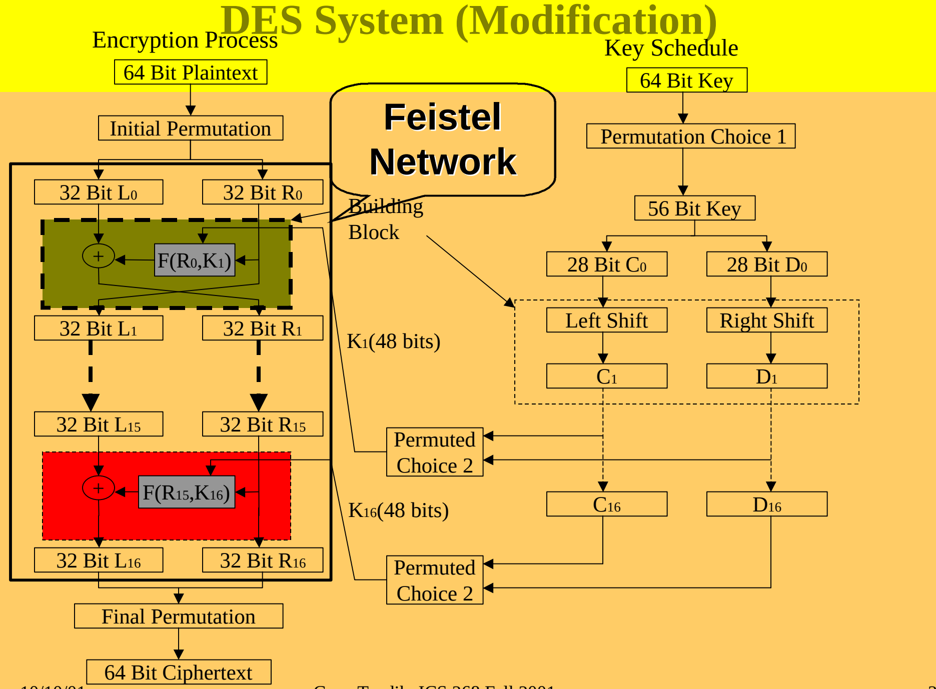
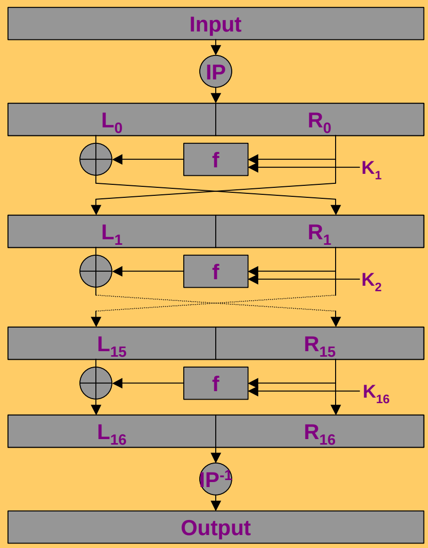
Lecture 5, October 8.

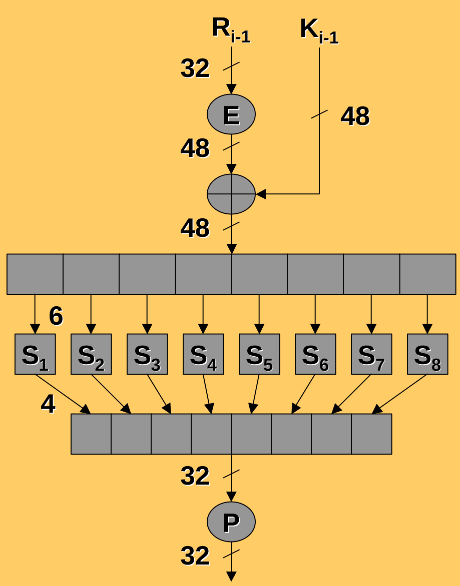
DES System (Modification)



DES System (Modified)



F Funtion



- Provide randomness by non-linear function S-box
- Every other operation of DES is linear
- Each S-box is 6 bit input, 4 bit output

S-box

Table 3.6 Primitive *S*-Box Functions

S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

Answer to Last Class Question

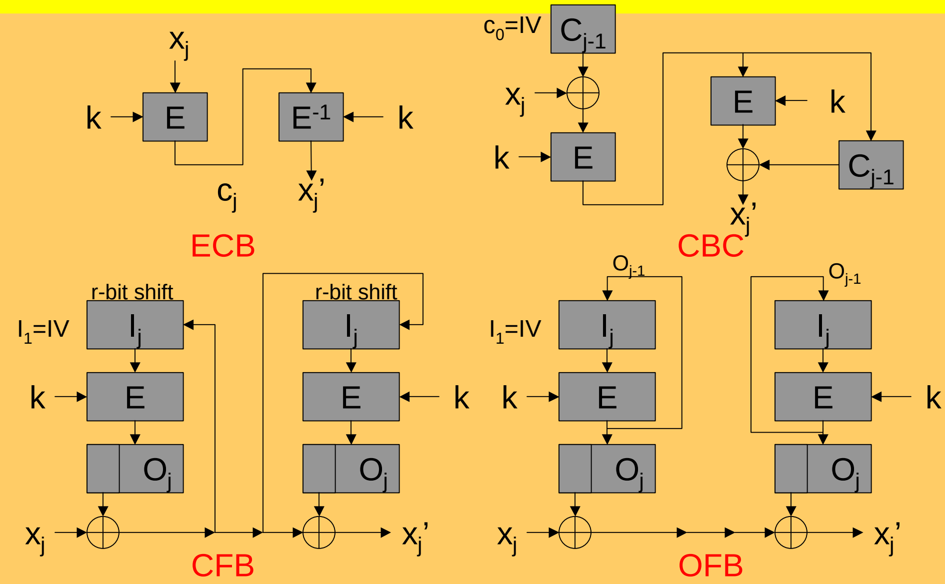
- **Question: f function has expansion and compression. How can you decrypt?**
  - **Answer: It does not matter ;-)**

- **Decryption**
  - $L_1^D = R_{16} = R_{15}$
  - $R_1^D = L_{16} \oplus f(R_{16}, K_{16})$

    $= L_{15} \oplus f(R_{15}, K_{16}) \oplus f(R_{15}, K_{16})$

    $= L_{15}$

Modes of operation



10/10/01 Gene Tsudik, ICS 268 Fall 2001 7

Modes of operation (cnt.)

- **ECB**
  1. **Encryption: $c_j \leftarrow E_K(x_j)$**
  2. **Decryption: $x_j \leftarrow E^{1}{}_{K}(c_j)$**
     - ▶ **Identical plaintext (under the same key) result in identical ciphertext**
     - ▶ **blocks are enciphered independently of other blocks**
     - ▶ **bit errors in a single ciphertext affect decipherment of that block only**
- **CBC**
  1. **Encryption: $c_0 \leftarrow$ IV, $c_j \leftarrow E_K(c_{j\,1} \oplus x_j)$**
  2. **Decryption: $c_0 \leftarrow$ IV, $x_j \leftarrow c_{j\,1} \oplus E^{1}{}_{K}(c_j)$**
     - ▶ **chaining causes ciphertext c_j to depend on all preceding plaintext**
     - ▶ **a single bit error in c_j affects decipherment of blocks c_j and c_{j+1}**
     - ▶ **self-synchronizing: error c_j (not c_{j+1}, c_{j+2}) is correctly decrypted to x_{j+2}.**
     - ▶ **Can use as a MAC: x_1, x_2, … , x_n, c_n**

10/10/01 Gene Tsudik, ICS 268 Fall 2001 8

Modes of operation (cnt.)

- CFB
  1. Encryption: $I_1 \leftarrow IV$
     1. $O_j \leftarrow E_K(I_j)$. (Compute the block cipher output)
     2. t_j : r leftmost bits of O_j (Assume the leftmost is identified as bit 1)
     3. $c_j \leftarrow x_j \oplus t_j$. (Transmit the r-bit ciphertext block c_j)
     4. Shift c_j into right end of shift register
  2. Decryption: $I_1 \leftarrow IV$, $x_j \leftarrow c_j \oplus t_j$,where t_j, O_j and I_j are as above

     - re-ordering ciphertext blocks affects decryption
     - one or more bit errors in any single r-bit ciphertext block cj affects the decipherment of next $\lceil n/r \rceil$ ciphertext blocks
     - self-synchronizing similar to CBC, but requires $\lceil n/r \rceil$ blocks to recover.
     - for r <n, throughput is decreased by a factor of n/r

Modes of operation (cnt.)

- CFB
  1. Encryption: $I_1 \leftarrow IV$
     1. $O_j \leftarrow E_K(I_j)$. (Compute the block cipher output)
     2. t_j : r leftmost bits of O_j (Assume the leftmost is identified as bit 1)
     3. $c_j \leftarrow x_j \oplus t_j$. (Transmit the r-bit ciphertext block c_j)
     4. Shift o_j into right end of shift register
  2. Decryption: $I_1 \leftarrow IV$, $x_j \leftarrow c_j \oplus t_j$,where t_j, O_j and I_j are as above

     - keystream is plaintext-independent
     - bit errors affects the decipherment of only that character
     - recovers from ciphertext bit errors, but cannot self-synchronize
     - for r <n, throughput is decreased as per the CFB mode

Breaking DES (Cryptanalysis)

- Differential Cryptanalysis
  - Differential cryptanalysis discovered in 1990; virtually all block ciphers from before that time are vulnerable...
  - ...except DES. IBM (and NSA) knew about it 15 years earlier
  - Looks for correlations in f()-function input and output
  - More precisely, the relation between input xor and output xor
    $Pr[f(X) \oplus f(X') = \Delta Y \mid X \oplus X' = \Delta X]$
- Linear cryptanalysis
  - Looks for correlations between key and cipher input and output
  - More precisely, relation between linear combination of input bits and linear combination of output bits
- Related-key cryptanalysis
  - Looks for correlations between key changes and cipher input/output

Breaking DES (Cryptanalysis)

Strength of DES Key size = 56 bits

•Brute force = 2^55 attempts

•Differential cryptanalysis = 2^47 attempts

•Linear cryptanalysis = 2^43 attempts

Longer than 56 bit keys don't make it any stronger

More than 16 rounds don't make it any stronger

DES Key Problems:

Weak keys (all 0s, all 1s, a few others)

Key size = 56 bits = 8 * 7-bit ASCII

Alphanumeric-only password converted to uppercase = 8 * ~5-bit chars = 40 bits

Breaking DES (COST)

DES was designed for efficiency in early-70's hardware

Makes it easy to build pipelined brute-force breakers in late-90's hardware

16 stages, tests 1 key per clock cycle

Can build a DES-breaker using:

* Field-programmable gate array (FPGA), software programmable hardware

* Application-specific IC (ASIC)

100 MHz ASIC = 100M keys per second per chip

Chips = $10 in 5K+ quantities $50,000 = 500 billion keys/sec

= 20 hours/key (40-bit DES takes 1 sec)

Breaking DES (COST)

•$1M = 1 hour per key (1/20 sec for 40 bit)

•$10M = 6 minutes per key (1/200 sec for 40 bits)

•US black budget is ~$25-30 billion!!!

•distributed.net = ~70 billion keys/sec with 20,000 computers (how long?)

•EFF (US non-profit) broke full DES in 2 1/2 days

•Amortised cost over 3 years = 8 cents per key

•If your secret is worth more than 8 cents, don't encrypt it with DES

•September 1998: German court rules DES "out of date and unsafe" for financial applications

DES Variants

- 2-DES (double DES) =

  E(K2,E(K1,X)) or D(K2,E(K1,X)) --- weak!
- 3-DES (triple DES) = E(K1,D(K2,E(K1,X)))or E(K3,D(K2,E(K1,X))) --- same security?
- DESx =

  K3 XOR E(K2, K1 XOR X): 2^{184} security proved by Rogaway

  K1 XOR E(K1,X)?

  E(K1, K1 XOR X)?

DES summary

- Permutation/substitution block cipher
- 64-bit data blocks
- 56-bit keys (8 parity bits)
- 16 rounds (shifts,xors)
- Key schedule
- S-box selection secret…
- DES “aging”
- 2-DES: rendezvous attack
- 3-DES: 112-bit security?
- DESX : 64-bit security?

Other ciphers

Skipjack

- Classified algorithm originally designed for Clipper,
- declassified in 1998
- 32 rounds, breakable with 31 rounds
- 80 bit key, inadequate for long-term security

GOST

- GOST 28147, Russian answer to DES
- 32 rounds, 256 bit key
- Incompletely specified

Other popular ciphers

IDEA (X. Lai, J. Massey, ETH)

- Developed as PES (proposed encryption standard),
- adapted to resist differential cryptanalysis as IPES, then IDEA
- Gained popularity via PGP, 128 bit key
- Patented (Ascom CH)

Blowfish (B. Schneier, Counterpane)

- Optimised for high-speed execution on 32-bit processors
- 448 bit key, relatively slow key setup
- Fast for bulk data on most PCs/laptops

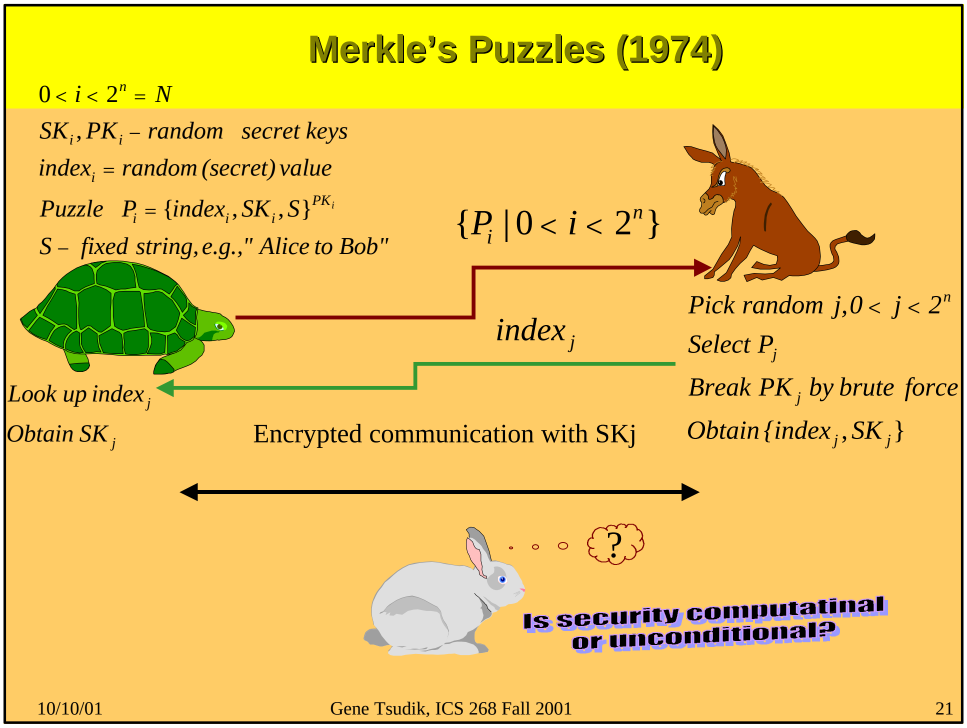
New Generation Block Cipher

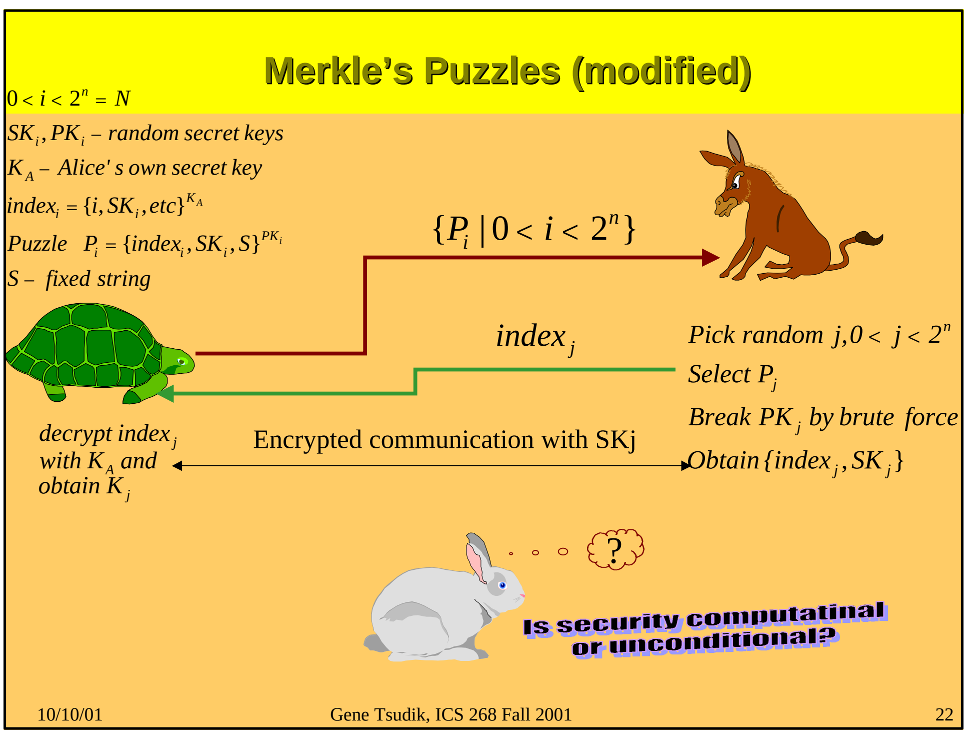
- AES (Advanced Encryption Standard)
  - Jan. 1997: initiation of the AES development
  - Sep. 1997: formal call for algorithms
    - unclassified, publicly disclosed encryption algorithm(s), available royalty-free, worldwide
    - block size: 128b, key size: 128, 192, 256b
  - Aug. 1998: a group of fifteen AES candidate
  - Mar. 1999: 2nd AES2, selected five algorithms
    - MARS, RC6, Rijndael, Serpent, and Twofish
  - Oct. 2000: Rijndael to propose for the AES
    - Fast for hardware/software implementation
    - Pretty strong

How do Alice and Bob get to share a secret key in the first place?

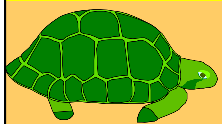
Or

Why do we need public key cryptography


Merkle's Puzzles (1974)
$0 < i < 2^n = N$
SK_i, PK_i – random secret keys
$index_i$ = random (secret) value
Puzzle $P_i = \{index_i, SK_i, S\}^{PK_i}$
S – fixed string, e.g., "Alice to Bob"
$\{P_i \mid 0 < i < 2^n\}$
$index_j$
Pick random $j, 0 < j < 2^n$
Select P_j
Break PK_j by brute force
Obtain $\{index_j, SK_j\}$
Look up $index_j$
Obtain SK_j
Encrypted communication with SKj
?
Is security computatinal or unconditional?
10/10/01
Gene Tsudik, ICS 268 Fall 2001
21

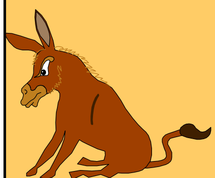

Merkle's Puzzles (modified)
$0 < i < 2^n = N$
SK_i, PK_i – random secret keys
K_A – Alice's own secret key
$index_i = \{i, SK_i, etc\}^{K_A}$
Puzzle $P_i = \{index_i, SK_i, S\}^{PK_i}$
S – fixed string
$\{P_i \mid 0 < i < 2^n\}$
$index_j$
Pick random $j, 0 < j < 2^n$
Select P_j
Break PK_j by brute force
Obtain $\{index_j, SK_j\}$
decrypt $index_j$ with K_A and obtain K_j
Encrypted communication with SKj
?
Is security computatinal or unconditional?
10/10/01
Gene Tsudik, ICS 268 Fall 2001
22

Merkle's Puzzles (contd.)



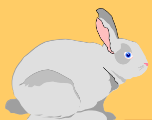
Alice:

- O(N) work to generate, compose, send puzzles
- Memory: Ka



Bob

- O(N) work to break a single puzzle



Eve

- O(N^2) work to break, on the average N/2 puzzles

Can we get better than N^2 to N advantage?