

Lecture 5

Encryption Continued...

1

Why not 2-DES ?

- 2DES: $C = \text{DES}(K_1, \text{DES}(K_2, P))$
- Seems to be hard to break by "brute force", approx. 2^{111} trials
- Assume Eve is trying to break 2DES and has a single (P, C) pair

Meet-in-the-middle (or Rendesvouz) ATTACK:

- I. For each possible K'_i (where $0 < i < 2^{56}$)
 1. Compute $C'_i = \text{DES}(K'_i, P)$
 2. Store: $[K'_i, C'_i]$ in table T (sorted by C'_i)
- II. For each possible K''_i (where $0 < i < 2^{56}$)
 1. Compute $C''_i = \text{DES}^{-1}(K''_i, C)$
 2. Lookup C''_i in T ← not expensive!
 3. If lookup succeeds, output: $K_1=K'_i, K_2=K''_i$

TOTAL COST: $O(2^{56})$ operations + $O(2^{56})$ storage ²

Other Symmetric Ciphers

RC4 (Ron's Cipher #4) Stream cipher:

- ❖ Optimized for fast software implementation
- ❖ Character streaming (not bit)
- ❖ 8-bit output
- ❖ Former trade secret of RSADSI,
- ❖ Reverse-engineered and posted to the net in 1994:
- ❖ 2048-bit key
- ❖ Used in many products until about 1999-2000

3

Other Symmetric Ciphers (RC4)

```
while( length-- ){          /* state[0-255] contains key bytes */
    sx = state[ ++x & 0xFF ];
    y += sx & 0xFF 256;
    sy = state[ y ];
    state[ y ] = sx;
    state[ x ] = sy;
    *data++ ^= state[ ( sx+sy ) & 0xFF ];
}
```

Takes about a minute to implement from memory

4

Other Symmetric Ciphers

- RC5
 - Suitable for hardware and software
 - Fast, simple
 - Adaptable to processors of different word lengths
 - Variable number of rounds
 - Variable-length key (0-256 bytes)
 - Very low memory requirements
 - High security (no effective attacks, yet...)
 - Data-dependent rotations

5

Other Symmetric Ciphers

- RC5 single round pseudocode:

$$L \leftarrow L \text{ XOR } R$$
$$L \leftarrow L \lll R$$
$$L \leftarrow L + \text{subkey}[2i]$$
$$R \leftarrow R \text{ XOR } L$$
$$R \leftarrow R \lll L$$
$$R \leftarrow R + \text{subkey}[2i + 1]$$

6

AES: The Rijndael Block Cipher

7

Introduction and History

- National Institute of Science and Technology (NIST) regulates standardization in the US
- DES is an aging standard that no longer meets today's needs for strong encryption
- Triple-DES: Endorsed by NIST as a "de facto" standard
- AES: Advanced Encryption Standard
 - Finalized in 2001
 - Goal is to define the Federal Information Processing Standard (FIPS) by selecting a new encryption algorithm suitable for encrypting (non-classified non-military) government documents
 - Candidate algorithms must be:
 - Symmetric-key ciphers supporting 128, 192, and 256 bit keys
 - Royalty-Free
 - Unclassified (i.e. public domain)
 - Available for worldwide export

8

Introduction and History

- AES Round-3 Finalist Algorithms:
 - MARS
 - Candidate offering from IBM Research
 - RC6
 - By Ron Rivest of MIT & RSA Labs, creator of the widely used RC4/RC5 algorithm and "R" in RSA
 - Twofish
 - From Counterpane Internet Security, Inc. (MN)
 - Serpent
 - by Ross Anderson (UK), Eli Biham (ISR) and Lars Knudsen (NO)
 - Rijndael
 - by Joan Daemen and Vincent Rijmen (B)

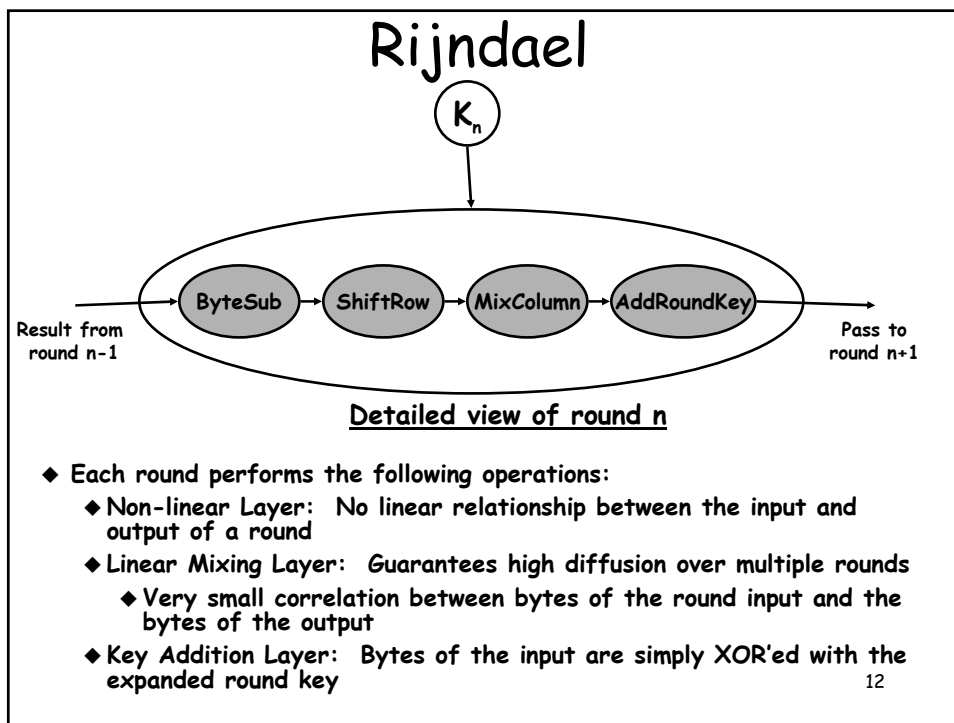
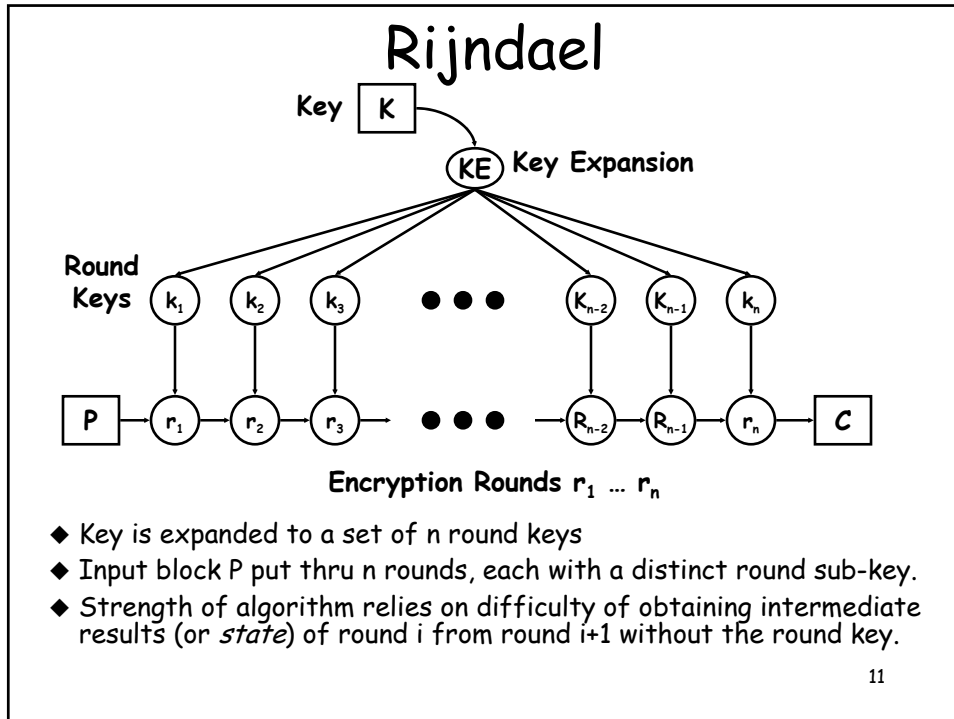
9

Rijndael

The Winner: Rijndael

- Joan Daemen (of Proton World International) and Vincent Rijmen (of Katholieke Universiteit Leuven).
- pronounced "Rhine-doll"
- Allows only 128, 192, and 256-bit key sizes (unlike other candidates)
- Variable input block length: 128, 192, or 256 bits. All nine combinations of key-block length possible.
 - A block is the smallest data size the algorithm will encrypt
- Vast speed improvement over DES in both hw and sw implementations
 - 8.416 bytes/sec on a 20MHz 8051
 - 8.8 Mbytes/sec on a 200MHz Pentium Pro

10

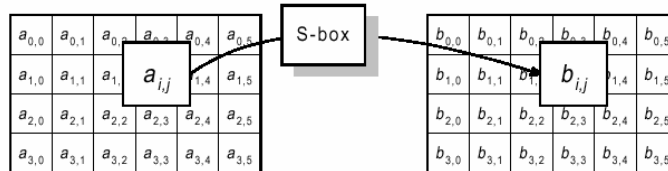


Rijndael

- Three layers provide strength against known types of cryptographic attacks: Rijndael provides "full diffusion" after only two rounds
- Immune to:
 - Linear and differential cryptanalysis
 - Related-key attacks
 - Square attack
 - Interpolation attacks
 - Weak keys
- Rijndael has been "shown" *secure*:
 - No key recovery attacks faster than exhaustive search exist
 - No known symmetry properties in the round mapping
 - No weak keys identified
 - No related-key attacks: No two keys have a high number of expanded round keys in common

13

Rijndael: ByteSub (192)



Each byte at the input of a round undergoes a non-linear byte substitution according to the following transform:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

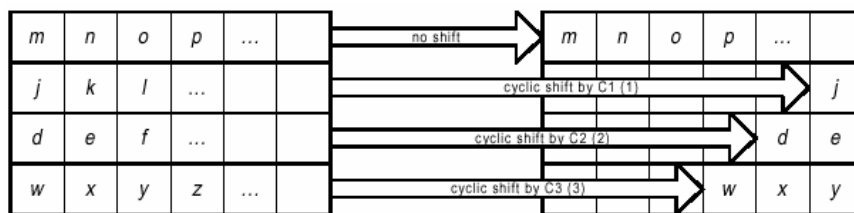
Substitution ("S")-box

14

Rijndael: ShiftRow

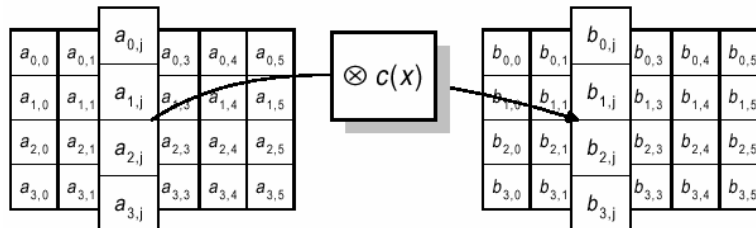
Nb	C1	C2	C3
4	1	2	3
6	1	2	3
8	1	3	4

Depending on the block length, each "row" of the block is cyclically shifted according to the above table



15

Rijndael: MixColumn



Each column is multiplied by a fixed polynomial
 $C(x) = '03'*X^3 + '01'*X^2 + '01'*X + '02'$

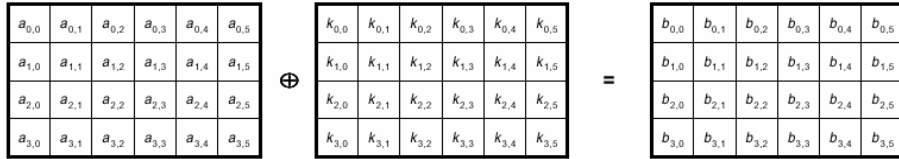
This corresponds to matrix multiplication $b(x) = c(x) \otimes a(x)$:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Not xor

16

Rijndael: Key Expansion and Addition



Each word is simply XOR'ed with the expanded round key

Key Expansion algorithm:

```

KeyExpansion(int* Key[4*Nk], int* EKey[Nb*(Nr+1)])
{
    for(i = 0; i < Nk; i++)
        EKey[i] = (Key[4*i], Key[4*i+1], Key[4*i+2], Key[4*i+3]);
    for(i = Nk; i < Nb * (Nr + 1); i++)
    {
        temp = EKey[i - 1];
        if (i % Nk == 0)
            temp = SubByte(RotByte(temp)) ^ Rcon[i / Nk];
        EKey[i] = EKey[i - Nk] ^ temp;
    }
}
    
```

17

Rijndael: Implementations

- Well-suited for software implementations on 8-bit processors (important for "Smart Cards")
 - Atomic operations focus on bytes and nibbles, not 32- or 64-bit integers
 - Layers such as ByteSub can be efficiently implemented using small tables in ROM (e.g. < 256 bytes).
 - No special instructions are required to speed up operation, e.g. barrel rotates
- For 32-bit implementations:
 - An entire round can be implemented via a fast table lookup routine on machines with 32-bit or higher word lengths
 - Considerable parallelism exists in the algorithm
 - Each layer of Rijndael operates in a parallel manner on the bytes of the round state, all four component transforms act on individual parts of the block
 - Although the Key expansion is complicated and cannot benefit much from parallelism, it only needs to be performed *once* until the two parties switch keys.

18

Rijndael: Implementations

- Hardware Implementations
 - Rijndael performs very well in software, but there are cases when better performance is required (e.g. server and VPN applications).
 - Multiple S-Box engines, round-key XORs, and byte shifts can all be implemented efficiently in hardware when absolute speed is required
 - Small amount of hardware can vastly speed up 8-bit implementations
- Inverse Cipher
 - Except for the non-linear ByteSub step, each part of Rijndael has a straightforward inverse and the operations simply need to be undone in the reverse order.
 - However, Rijndael was specially written so that the same code that encrypts a block can also decrypt the same block simply by changing certain tables and polynomials for each layer. The rest of the operation remains identical.

19

Conclusions and The Future

- Rijndael is an extremely fast, state-of-the-art, highly secure algorithm
- Amenable to efficient implementation in both hw and sw; requires no special instructions to obtain good performance on any computing platform
- Triple-DES, still highly secure and supported by NIST, is expected to be common for the foreseeable future.

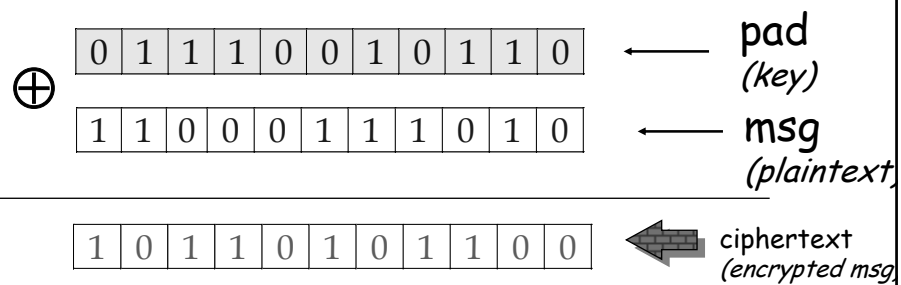
20

Reminder:
World's best cipher!

21

One-time pad

For each character:



22

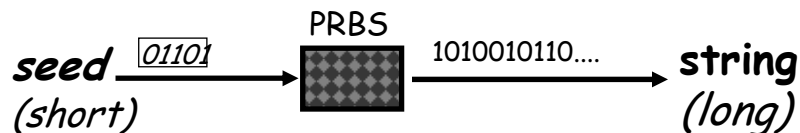
One-time pad (cont.)

- Symmetric
- Pad is selected at random
- Pad is as long as plaintext
- Perfectly secure, but...
- One time only:
so sending the pad is just as hard as
sending the msg

23

A more realistic version: Pseudo-random OTP

Pseudo-random bit string (PRBS) generator:



Computationally Hard to guess a bit (after seeing
many others)

24