



EEC-484/584

Computer Networks

Lecture 23

Wenbing Zhao

wenbing@ieee.org

(Lecture notes are based on materials supplied by
Dr. Louise Moser at UCSB and Prentice-Hall)



Outline

2

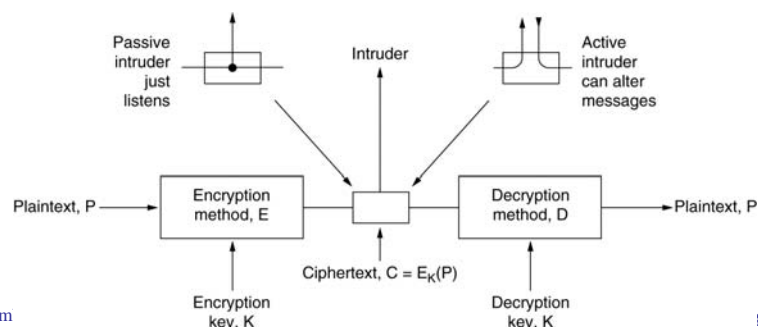
- Review of last lecture
 - Introduction to cryptography
- Today's topics
 - Symmetric-key algorithms
 - Public-key algorithms
 - Digital signatures

Cryptography

- Introduction to Cryptography
- Substitution Ciphers
- Transposition Ciphers
- One-Time Pads
- Two Fundamental Cryptographic Principles

An Introduction to Cryptography

- **Plaintext:** message to be encrypted
- **Key:** string of characters used to encrypt the message
- **Ciphertext:** encrypted message
- $D_K(E_K(P)) = P$





Introduction to Cryptography

- Cryptanalysis problems
 - **Ciphertext-only**: cryptanalyst has a quantity of ciphertext and no plaintext
 - **Known plaintext**: cryptanalyst has some matched ciphertext and plaintext
 - **Chosen plaintext**: cryptanalyst has the ability to encrypt pieces of plaintext of his own choosing
- Encryption methods
 - **Substitution** ciphers
 - **Transposition** ciphers



Substitution Ciphers

- Idea: each letter or group of letters is replaced by another letter or group of letters
- **Caesar cipher** – circularly shift by 3 letters
 - a -> D, b -> E, ... z -> C
 - More generally, shift by k letters, k is the key
- **Monoalphabetic cipher** – map each letter to some other letter
 - A b c d e f ... w x y z
 - Q W E R T Y ... V B N M <= the key

Quantum Cryptography

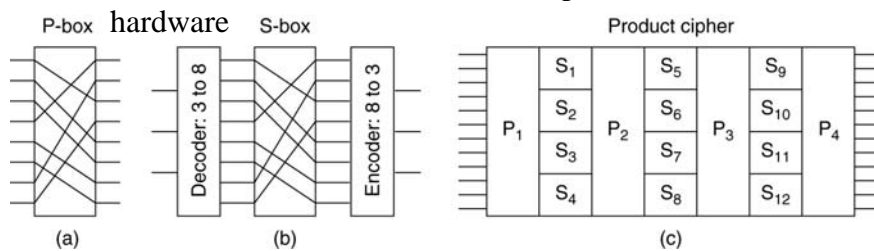
Bit number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Data	1	0	0	1	1	1	0	0	1	0	1	0	0	1	1	0
(a) What Alice sends																
(b) Bob's bases																
(c) What Bob gets																
(d) Correct basis?	No	Yes	No	Yes	No	No	No	Yes	Yes	No	Yes	Yes	Yes	No	Yes	No
(e) One-time pad		0		1					0	1		1	0	0		1
(f) Trudy's bases																
(g) Trudy's pad	x	0	x	1	x	x	x	?	1	x	?	?	0	x	?	x

Symmetric-Key Algorithms

- DES – The Data Encryption Standard
- AES – The Advanced Encryption Standard
- Cipher Modes
- Other Ciphers
- Cryptanalysis

Data Encryption Standard

- Aim: to make encryption algorithm so complicated that not even a computer can break it in reasonably amount of time
 - P-box (permutation box) used to implement transposition in hardware
 - S-box (substitution box) used to implement substitution in hardware



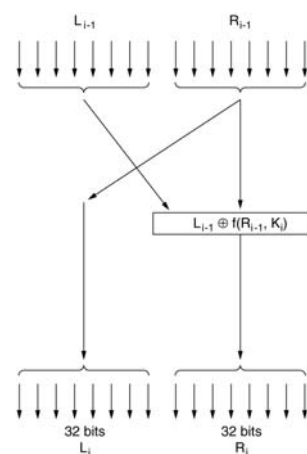
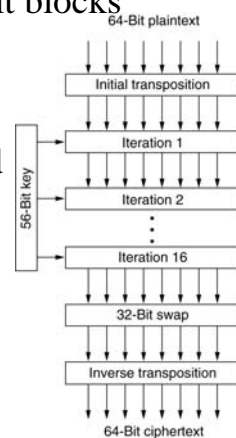
28 November 2005

EEC484/584

Wenbing Zhao

Data Encryption Standard

- Developed by IBM. US standard for unclassified info (1977)
- Same key for encryption as for decryption
- Encrypts in 64-bit blocks
- Uses 56-bit key
- Has 19 stages, 16 parameterized by different functions of the key





DES Algorithm

- Four steps of function f
 - Construct 48-bit number E by expanding 32-bit number R_{i-1} according to fixed transposition and duplication rule
 - XOR E and K_i
 - Partition output into 8 groups of 6 bits each. Input each to different S-box, S-box produces 4 output bits, result 8 4-bit numbers
 - Pass 32 bits through P-box

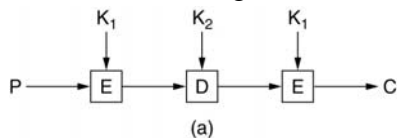


DES Algorithm

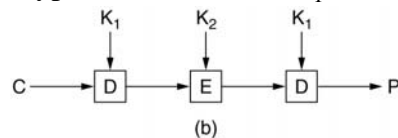
- In each of 16 iterations, different key is used
- Before algorithm starts, 56-bit transposition applied to key
- Before each iteration, key is partitioned into two 28-bit numbers, each rotated left by number of bits determined by iteration number. K_i is obtained from rotated key by applying another 56-bit transposition

Triple DES

- Triple DES – effectively increases the key length. It uses two keys and three stages
 - In first stage, the plaintext is encrypted using DES in the usual way with K_1
 - In second stage, DES is run in decryption mode, using K_2 as the key
 - In third stage, another DES encryption is done with K_1



Triple DES encryption



Triple DES decryption

28 November 2005

EEC484/584

Wenbing Zhao

AES – The Advanced Encryption Standard

- AES is a result of a cryptographic contest
 - Organized by NIST in 1997
- Rules for AES proposals
 1. The algorithm must be a symmetric block cipher
 2. The full design must be public
 3. Key lengths of 128, 192, and 256 bits supported
 4. Both software and hardware implementations required
 5. The algorithm must be public or licensed on nondiscriminatory terms
- Winner: Rijndael (from two Belgian cryptographers: Joan Daemen and Vincent Rijmen)

28 November 2005

EEC484/584

Wenbing Zhao

An Outline of Rijndael

```

#define LENGTH 16                /* # bytes in data block or key */
#define NROWS 4                 /* number of rows in state */
#define NCOLS 4                 /* number of columns in state */
#define ROUNDS 10              /* number of iterations */
typedef unsigned char byte;     /* unsigned 8-bit integer */

rijndael(byte plaintext[LENGTH], byte ciphertext[LENGTH], byte key[LENGTH])
{
    int r;                       /* loop index */
    byte state[NROWS][NCOLS];    /* current state */
    struct {byte k[NROWS][NCOLS];} rk[ROUNDS + 1]; /* round keys */

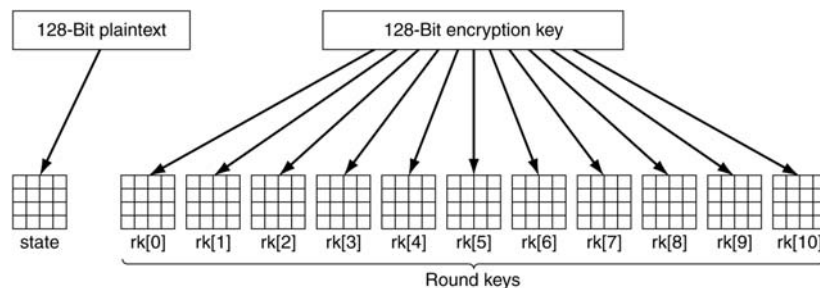
    expand_key(key, rk);          /* construct the round keys */
    copy_plaintext_to_state(state, plaintext); /* init current state */
    xor_roundkey_into_state(state, rk[0]); /* XOR key into state */

    for (r = 1; r <= ROUNDS; r++) {
        substitute(state);        /* apply S-box to each byte */
        rotate_rows(state);       /* rotate row i by i bytes */
        if (r < ROUNDS) mix_columns(state); /* mix function */
        xor_roundkey_into_state(state, rk[r]); /* XOR key into state */
    }
    copy_state_to_ciphertext(ciphertext, state); /* return result */
}

```

AES

■ Creating of the *state* and *rk* arrays



Cipher Modes

- Despite all the complexity, AES and DES (or any block cipher) is basically a monoalphabetic substitution cipher using big characters
 - Whenever the same plaintext block goes in the front end, the same ciphertext block comes out the back end
 - If you encrypt the plaintext abcdefgh 100 times with same DES key, you get the same ciphertext 100 times
 - An intruder can exploit this property to help subvert the cipher

Electronic Code Book Mode

- The plaintext of a file encrypted as 16 DES blocks
 - One can make a copy of a block that contains a bigger bonus and replace the block that contains a smaller bonus

Name	Position	Bonus
A d a m s , L e s l i e	C l e r k	\$ _ _ _ _ 1 0
B l a c k , R o b i n	B o s s	\$ 5 0 0 , 0 0 0
C o l l i n s , K i m	M a n a g e r	\$ 1 0 0 , 0 0 0
D a v i s , B o b b i e	J a n i t o r	\$ _ _ _ _ 5

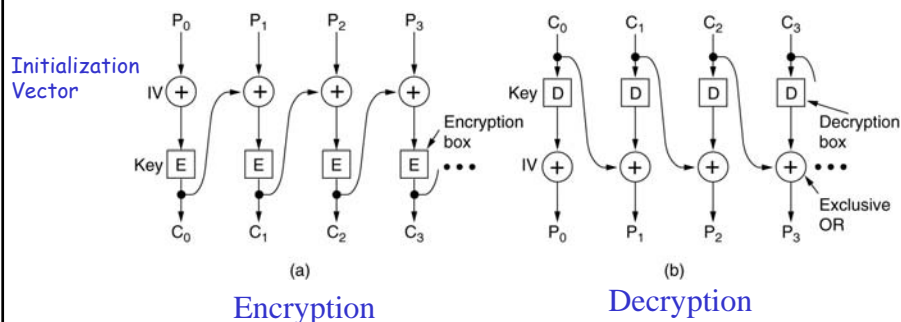
Bytes ← 16 8 8

Cipher Block Chaining Mode

- To avoid the ECB mode problem: replacing a block will cause the plaintext decrypted starting at the replaced to be garbage
- Exclusive OR the encrypted text with the next block of plaintext before encryption: $C_0 = E(P_0 \text{ XOR } IV)$, $C_1 = E(P_1 \text{ XOR } C_0)$, etc.
- **Drawback:** must wait until full 64-bit (128-bit) block to arrive to decrypt

Cipher Block Chaining Mode

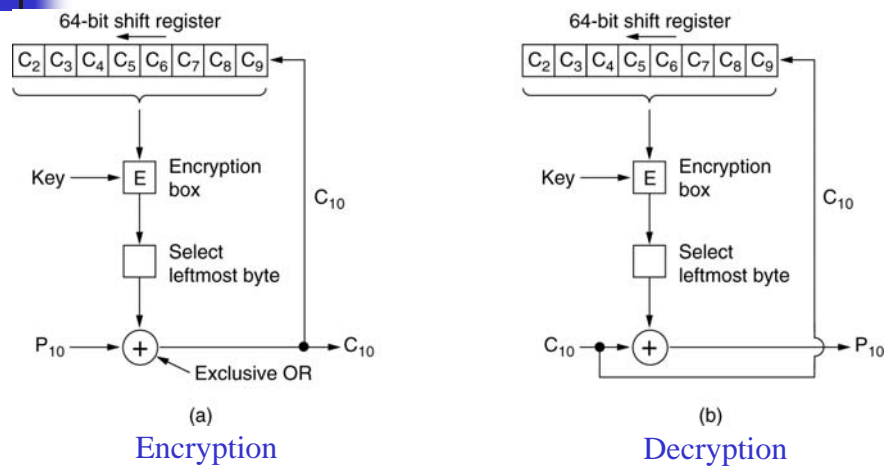
- Exclusive OR the encrypted text with the next block of plaintext before encryption: $C_0 = E(P_0 \text{ XOR } IV)$, $C_1 = E(P_1 \text{ XOR } C_0)$, etc.



Cipher Feedback Mode

- To enable byte-by-byte encryption
 - When plaintext byte n (P_n) arrives, DES algorithm operates a 64-bit register to generate a 64-bit ciphertext (128-bit register needed for AES)
 - Leftmost byte of that ciphertext is extracted and XORed with P_n
 - That byte is transmitted on the transmission line
 - The shift register is shifted left 8 bits, causing C_{n-8} to fall off the left end, and C_n is inserted in the position just vacated at the right end by C_9
- **Drawback:** One byte of transmission error will ruin 8 bytes of data

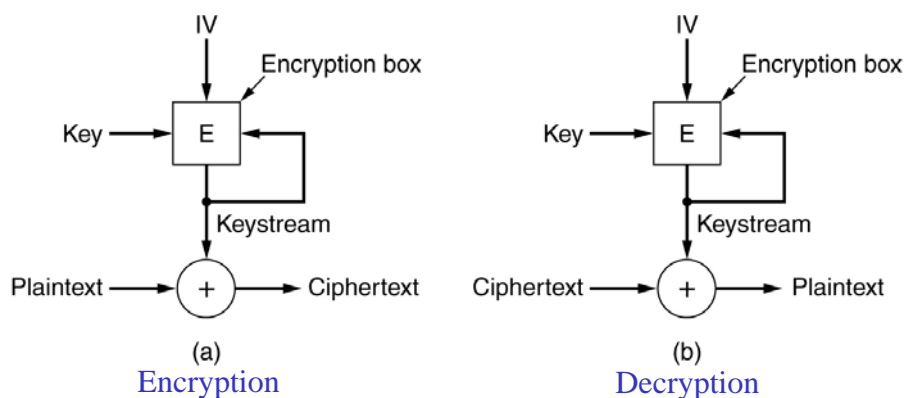
Cipher Feedback Mode



Stream Cipher Mode

- To be insensitive to transmission error, an arbitrarily large sequence of output blocks, called the **keystream**, is treated like a one-time pad and XORed with the plaintext to get the ciphertext
 - It works by encrypting an IV, using a key to get an output block
 - The output block is then encrypted, using the key to get a second output block
 - This block is then encrypted to get a third block, and so on
- **The keystream is independent of the data**, so (1) It can be computed in advance (2) It is completely insensitive to transmission errors

Stream Cipher Mode





Stream Cipher Mode

- It is essential never to use the same (key, IV) pair twice with a stream cipher because doing so will generate the same keystream each time
- Using the same keystream twice exposes the ciphertext to a **keystream reuse attack**

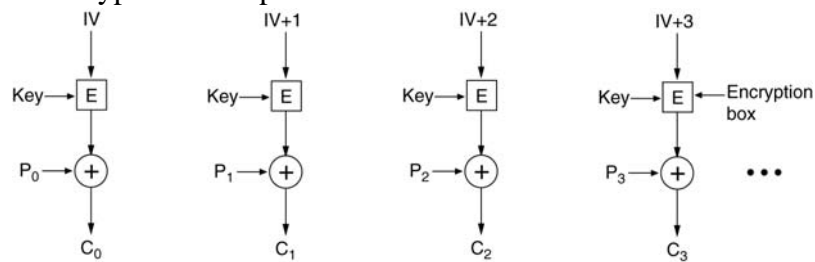


Keystream Reuse Attack

- Plaintext block, P_0 , is encrypted with the keystream to get $P_0 \text{ XOR } K_0$
- Later, a second plaintext block, Q_0 , is encrypted with the same keystream to get $Q_0 \text{ XOR } K_0$
- An intruder who captures both ciphertext blocks can simply XOR them together to get $P_0 \text{ XOR } Q_0$, which eliminates the key
- The intruder now has the XOR of the two plaintext blocks
- If one of them is known or can be guessed, the other can also be found
- In any event, the XOR of two plaintext streams can be attacked by using statistical properties of the message

Counter Mode

- To allow random access to encrypted data
 - The IV plus a constant is encrypted, and the resulting ciphertext XORed with the plaintext
 - By stepping the IV by 1 for each new block, it is easy to decrypt a block anywhere in the file without first having to decrypt all of its predecessors



28 November 2005

EEC484/584

Wenbing Zhao

Other Ciphers

- Some common symmetric-key cryptographic algorithms

Cipher	Author	Key length	Comments
Blowfish	Bruce Schneier	1–448 bits	Old and slow
DES	IBM	56 bits	Too weak to use now
IDEA	Massey and Xuejia	128 bits	Good, but patented
RC4	Ronald Rivest	1–2048 bits	Caution: some keys are weak
RC5	Ronald Rivest	128–256 bits	Good, but patented
Rijndael	Daemen and Rijmen	128–256 bits	Best choice
Serpent	Anderson, Biham, Knudsen	128–256 bits	Very strong
Triple DES	IBM	168 bits	Second best choice
Twofish	Bruce Schneier	128–256 bits	Very strong; widely used

28 November 2005

EEC484/584

Wenbing Zhao



Cryptanalysis

- **Differential cryptanalysis:** can be used to attack any block cipher (Biham and Shamir, 1993)
 - It works by beginning with a pair of plaintext blocks that differ in only a small number of bits and watching carefully what happens on each internal iteration as the encryption proceeds
 - In many cases, some bit patterns are much more common than other patterns, and this observation lead to a probabilistic attack



Cryptanalysis

- **Linear cryptanalysis:** it can break DES with only 243 known plaintexts (Matsui, 1994)
 - It works by XORing certain bits in the plaintext and ciphertext together and examining the result for patterns
 - When this is done repeatedly, half bits should be 0s and half should be 1s
 - Often, however, ciphers introduce a bias in one direction or the other, and this bias, however small, can be exploited to reduce the work factor



Cryptanalysis

- Using **analysis of the electrical power consumption** to find secret keys
 - Computers typically use 3 volts to represent a 1 bit, and 0 volts to represent a 0 bit. Thus processing a 1 takes more electrical energy than processing a 0
 - If a cryptographic algorithm consists of a loop in which the key bits are processed in order, an attacker who replaces the main clock with a slow clock (e.g., 100Hz) can precisely monitor the power consumed by each machine instruction
 - From this data, deducing the key is surprisingly easy



Cryptanalysis

- **Timing analysis:** cryptographic algorithms are full of if statements that test bits in the round keys
 - If the then and else parts take different amounts of time, by slowing down the clock and seeing how long various steps take, it may also be possible to deduce the round keys
 - Once all the round keys are known, the original key can usually be computed

Public-Key Algorithms

- Distributing keys => the weakest link in most cryptosystems
 - No matter how strong a cryptosystem was, if an intruder could steal the key, the system was worthless
 - Cryptologists always took for granted that the encryption key and decryption key were the same
- Diffie and Hellman (1976) proposed a radically new kind of cryptosystem: encryption and decryption keys were different
 - $D(E(P)) = P$
 - It is exceedingly difficult to deduce D from E
 - E cannot be broken by a chosen plaintext attack

Public-Key Algorithms

- Public-key cryptography:
 - Encryption algorithm and the encryption key can be made public
- How to establish a secure channel
 - Alice and Bob have never had previous contact
 - Alice sends Bob $E_B(P)$ (message P encrypted using Bob's public encryption key E_B)
 - Bob receives the encrypted message and retrieves the plaintext by using his private key $P = D_B(E_B(P))$
 - Bobs then sends a reply $E_A(R)$ to Alice

RSA

- Rivest, Shamir, Adleman, 1978: a good method for public-key cryptography
- RSA method:
 - Choose two large primes, p and q (typically 1024 bits)
 - Compute $n = p \times q$ and $z = (p-1) \times (q-1)$
 - Choose a number relatively prime to z and call it d
 - Find e such that $e \times d = 1 \pmod{z}$
- To encrypt a message, P , Compute $C = P^e \pmod{n}$
- To decrypt C , compute $P = C^d \pmod{n}$
- The public key consists of the pair (e, n)
- The private key consists of the pair (d, n)

RSA

- An example of the RSA algorithm
 - $P = 3, q = 11 \Rightarrow n = 33$ and $z = 20$
 - A suitable value for $d = 7$
 - e can be found by solving the eq. $7e = 1 \pmod{20} \Rightarrow e = 3$
 - $C = P^3 \pmod{33}, P = C^7 \pmod{33}$

Plaintext (P)		Ciphertext (C)			After decryption	
Symbolic	Numeric	P^3	$P^3 \pmod{33}$	C^7	$C^7 \pmod{33}$	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E

Sender's computation
Receiver's computation



Other Public-Key Algorithms

- A method based on the difficulty of computing discrete logarithms (El Gamal, 1985 and Schnorr, 1991)
- Knapsack algorithm (Merkle and Hellman, 1978). Not considered secure and not used in practice any more
 - Someone owns a large number of objects, each with a different weight
 - The owner encodes the message by secretly selecting a subset of the objects and placing them in the knapsack
 - The total weight of the objects in the knapsack is made public, as is the list of all possible objects
 - The list of objects in the knapsack is kept secret

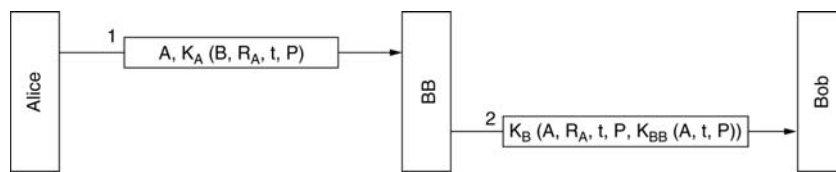


Digital Signatures

- **Requirement on digital signatures:** one party can send a signed message to another party in such a way that the following conditions hold:
 - The receiver can verify the claimed identity of the sender
 - The sender cannot later repudiate the contents of the message
 - The receiver cannot possibly have concocted the message himself

Symmetric-Key Signatures

- Big Brother (BB): a central authority that knows everything and whom everyone trusts
 - Each user chooses a secret key and shares it with BB
- Digital signatures with Big Brother



Public-Key Signatures

- Digital signatures using public-key cryptography
 - Requires $E(D(P)) = P$ (in addition to $D(E(P)) = P$)

