

# More On Cryptography

## CS 239

### Computer Security

February 7, 2007

CS 236, Winter 2007

Lecture 5  
Page 1

## Outline

- Permutation ciphers
- Stream and block ciphers
- Uses of cryptography

CS 236, Winter 2007

Lecture 5  
Page 2

## Permutation Ciphers

- Instead of substituting different characters, scramble up the existing characters
- Use algorithm based on the key to control how they're scrambled
- Decryption uses key to unscramble

CS 236, Winter 2007

Lecture 5  
Page 3

## Characteristics of Permutation Ciphers

- Doesn't change the characters in the message
  - Just where they occur
- Thus, character frequency analysis doesn't help cryptanalyst

CS 236, Winter 2007

Lecture 5  
Page 4

## Columnar Transpositions

- Write the message characters in a series of columns
- Copy from top to bottom of first column, then second, etc.


CS 236, Winter 2007

Lecture 5  
Page 5

## Example of Columnar Substitution

How did this transformation happen?

T	r	a	n	s	i
e	r	\$	l	o	
o	t	o	m		
y	s	a	v	i	
n	g	s	a	c	
e	o	u	n	t	



T	e	o	y	n	e
r	r			g	o
a	t	s	s	u	
n	\$	e	a	n	
s	l		v	a	t
f	o	m	i	c	

Looks a lot more cryptic written this way:

Te0yncrr goa tssun\$oa ns1 vatf0mic

CS 236, Winter 2007

Lecture 5  
Page 6

## Attacking Columnar Transformations

- The trick is figuring out how many columns were used
- Use information about digrams, trigrams, and other patterns
- Digrams are letters that frequently occur together (“re”, “th”, “en”, e.g.)
- For each possibility, check digram frequency

CS 236, Winter 2007

Lecture 5  
Page 7

## For Example,

Te0yncrr goa tssun\$oa ns1 vatff0mic  
\$ 1 0 0

In our case, the presence of dollar signs and numerals in the text is suspicious

Maybe they belong together?

Umm, maybe there's 6 columns?

CS 236, Winter 2007

Lecture 5  
Page 8

## Double Transpositions

- Do it twice
- Using different numbers of columns each time
- Find pairs of letters that probably appeared together in the plaintext
- Figure out what transformations would put them in their positions in the ciphertext

CS 236, Winter 2007

Lecture 5  
Page 9

## Generalized Transpositions

- Any algorithm can be used to scramble the text
- Usually somehow controlled by a key
- Generality of possible transpositions makes cryptanalysis harder

CS 236, Winter 2007

Lecture 5  
Page 10

## Which Is Better, Transposition or Substitution?

- Well, neither, really
- Strong modern ciphers tend to use both
- Transposition scrambles text patterns
- Substitution hides underlying text characters/bits
- Combining them can achieve both effects
  - If you do it right . . .

CS 236, Winter 2007

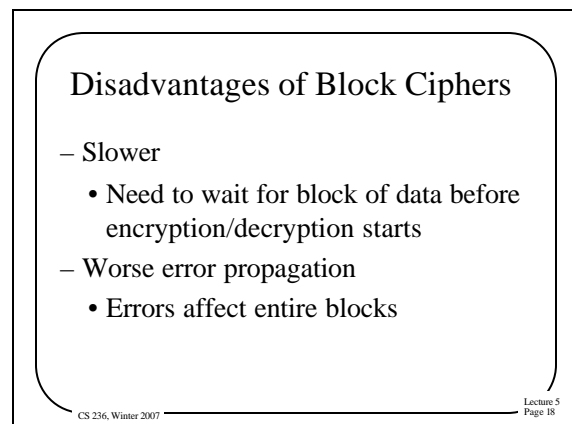
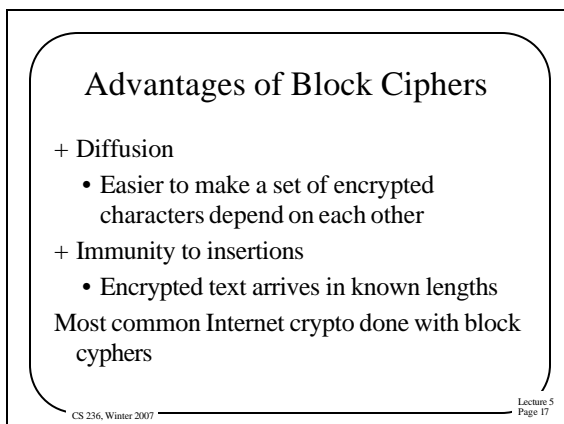
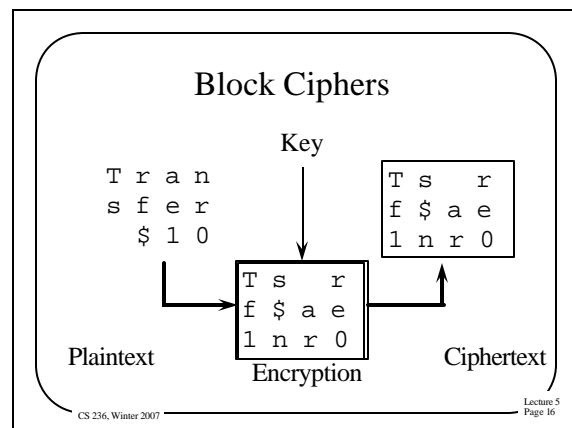
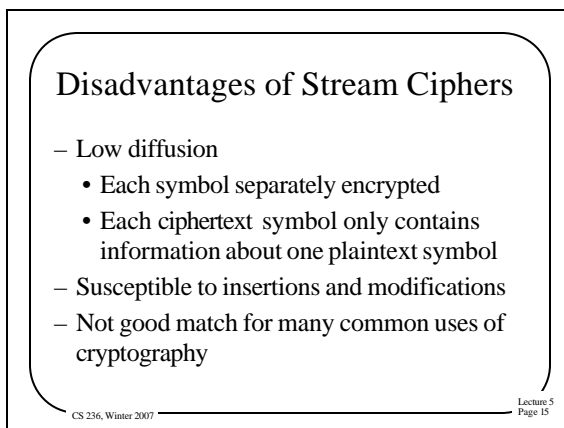
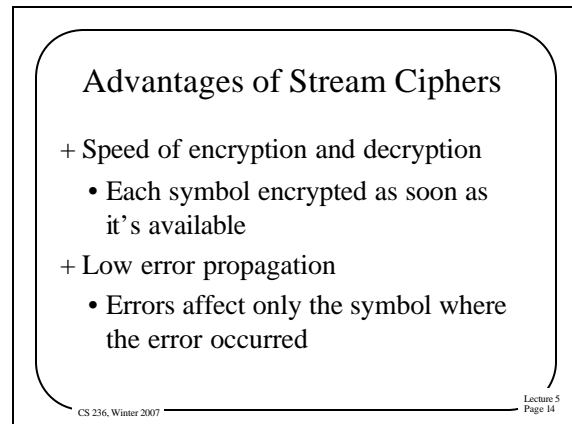
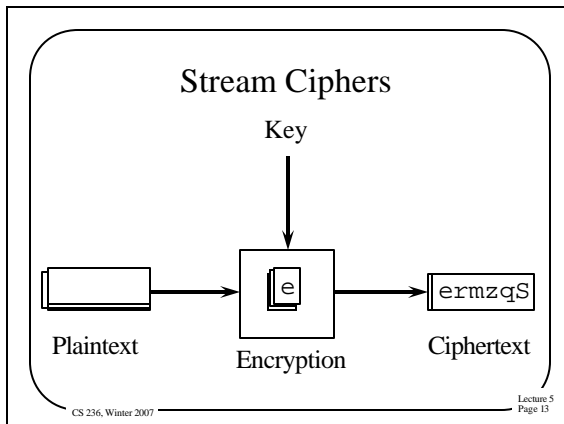
Lecture 5  
Page 11

## Stream and Block Ciphers

- Stream ciphers convert one symbol of plaintext immediately into one symbol of ciphertext
- Block ciphers work on a given sized chunk of data at a time

CS 236, Winter 2007

Lecture 5  
Page 12



### Desirable Characteristics of Ciphers

- Well matched to requirements of application
  - Amount of secrecy required should match labor to achieve it
- Freedom from complexity
  - The more complex algorithms or key choices are, the worse

CS 236, Winter 2007

Lecture 5  
Page 19

### More Characteristics

- Simplicity of implementation
  - Seemingly more important for hand ciphering
  - But relates to probability of errors in computer implementations
- Errors should not propagate

CS 236, Winter 2007

Lecture 5  
Page 20

### Yet More Characteristics

- Ciphertext size should be same as plaintext size
- Encryption should maximize *confusion*
  - Relation between plaintext and ciphertext should be complex
- Encryption should maximize *diffusion*
  - Plaintext information should be distributed throughout ciphertext

CS 236, Winter 2007

Lecture 5  
Page 21

### Uses of Cryptography

- What can we use cryptography for?
- Lots of things
  - Secrecy
  - Authentication
  - Prevention of alteration

CS 236, Winter 2007

Lecture 5  
Page 22

### Cryptography and Secrecy

- Pretty obvious
- Only those knowing the proper keys can decrypt the message
  - Thus preserving secrecy
- Used cleverly, it can provide other forms of secrecy

CS 236, Winter 2007

Lecture 5  
Page 23

### Cryptography and Authentication

- How can I prove to you that I created a piece of data?
- What if I give you the data in encrypted form?
  - Using a key only you and I know
- Then only you or I could have created it
  - Unless one of us told someone else the key . . .

CS 236, Winter 2007

Lecture 5  
Page 24

### Some Limitations on Cryptography and Authentication

- If both parties cooperative, cryptography can authenticate
  - Problems with non-repudiation, though
- What if three parties want to share a key?
  - No longer certain who created anything
  - Public key cryptography can solve this problem
- What if I want to prove authenticity without secrecy?

CS 236, Winter 2007

Lecture 5  
Page 25

### Cryptography and Non- Alterability

- Changing one bit of an encrypted message completely garbles it
  - For many forms of cryptography
- If a checksum is part of encrypted data, that's detectable
- If you don't need secrecy, can get the same effect
  - By encrypting only the checksum

CS 236, Winter 2007

Lecture 5  
Page 26

### Cryptography and Zero- Knowledge Proofs

- With really clever use, cryptography can be used to prove I know a secret
  - Without telling you the secret
- Seems like magic, but it can work
- Basically, using multiple levels of cryptography in very clever ways

CS 236, Winter 2007

Lecture 5  
Page 27

### Symmetric and Asymmetric Cryptosystems

- Symmetric - the encrypter and decrypter share a secret key
  - Used for both encrypting and decrypting
- Asymmetric – encrypter has different key than decrypter

CS 236, Winter 2007

Lecture 5  
Page 28

### Description of Symmetric Systems

- $C = E(K, P)$
- $P = D(K, C)$
- $E()$  and  $D()$  are not necessarily the same operations

CS 236, Winter 2007

Lecture 5  
Page 29

### Advantages of Symmetric Key Systems

- + Encryption and authentication performed in a single operation
- + Well-known (and trusted) ones perform faster than asymmetric key systems
- + Doesn't require any centralized authority
  - Though key servers help a lot

CS 236, Winter 2007

Lecture 5  
Page 30

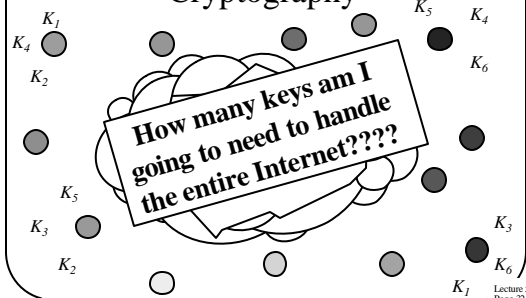
## Disadvantage of Symmetric Key Systems

- Encryption and authentication performed in a single operation
  - Makes signature more difficult
- Non-repudiation hard without servers
- Key distribution can be a problem
- Scaling

CS 236, Winter 2007

Lecture 5  
Page 31

## Scaling Problems of Symmetric Cryptography



CS 236, Winter 2007

Lecture 5  
Page 32

## Sample Symmetric Key Ciphers

- The Data Encryption Standard
- The Advanced Encryption Standard
- There are many others

CS 236, Winter 2007

Lecture 5  
Page 33

## The Data Encryption Standard

- Probably the best known symmetric key cryptosystem
- Developed in 1977
- Still much used
  - Which implies breaking it isn't trivial
- But showing its age

CS 236, Winter 2007

Lecture 5  
Page 34

## History of DES

- Developed in response to National Bureau of Standards studies
- Developed by IBM
- Analyzed, altered, and approved by the National Security Agency
- Adopted as a federal standard
- One of the most widely used encryption algorithms

CS 236, Winter 2007

Lecture 5  
Page 35

## Overview of DES Algorithm

- A block encryption algorithm
  - 64 bit blocks
- Uses substitution and permutation
  - Repeated applications
    - 16 cycles worth
- 64 bit key
  - Only 56 bits really used, though

CS 236, Winter 2007

Lecture 5  
Page 36

## More On DES Algorithm

- Uses substitutions to provide confusion
  - To hide the set of characters sent
- Uses transpositions to provide diffusion
  - To spread the effects of one plaintext bit into other bits
- Uses only standard arithmetic and logic functions and table lookup
- Performs 16 rounds of substitutions and permutations
  - Involving the key in each round

CS 236, Winter 2007

Lecture 5  
Page 37

## Decrypting DES

- For DES,  $D()$  is the same as  $E()$
- You decrypt with exactly the same algorithm
- If you feed ciphertext and the same key into DES, the original plaintext pops out

CS 236, Winter 2007

Lecture 5  
Page 38

## Is DES Secure?

- Apparently, reasonably
- NSA alterations believed to have increased security against differential cryptanalysis
- Some keys are known to be weak with DES
  - So good implementations reject them
- To date, only brute force attacks have publicly cracked DES

CS 236, Winter 2007

Lecture 5  
Page 39

## Key Length and DES

- Easiest brute force attack is to try all keys
  - Looking for a meaningful output
- Cost of attack proportional to number of possible keys
- Is  $2^{56}$  enough keys?
- Not if you seriously care
  - Cracked via brute force in 1998
  - Took lots of computers and time
  - But computers keep getting faster . . .

CS 236, Winter 2007

Lecture 5  
Page 40

## Does This Mean DES is Unsafe?

- Depends on what you use it for
- Takes lots of compute power to crack
- On the other hand, computers will continue to get faster
- And motivated opponents can harness vast resources
- Increasingly being replaced by AES

CS 236, Winter 2007

Lecture 5  
Page 41

## The Advanced Encryption Standard

- A relatively new cryptographic algorithm
- Intended to be the replacement for DES
- Chosen by NIST
  - Through an open competition
- Chosen cipher was originally called Rijndael
  - Developed by Dutch researchers
  - Uses combination of permutation and substitution

CS 236, Winter 2007

Lecture 5  
Page 42

## Increased Popularity of AES

- Gradually replacing DES
  - As was intended
- Various RFCs describe using AES in IPSEC
- FreeS/WAN IPSEC (for Linux) includes AES
- Some commercial VPNs use AES
- Various Windows AES products available
  - Used for at least some purposes in Vista

CS 236, Winter 2007

Lecture 5  
Page 43

## Public Key Encryption Systems

- The encrypter and decrypter have different keys

$$C = E(K_E, P)$$

$$P = D(K_D, C)$$

- Often, works the other way, too

$$C \stackrel{?}{=} E(K_D, P)$$

$$P \stackrel{?}{=} D(K_E, C)$$

CS 236, Winter 2007

Lecture 5  
Page 44

## History of Public Key Cryptography

- Invented by Diffie and Hellman in 1976
- Merkle and Hellman developed Knapsack algorithm in 1978
- Rivest-Shamir-Adelman developed RSA in 1978
  - Most popular public key algorithm
- Many public key cryptography advances secretly developed by British and US government cryptographers earlier

CS 236, Winter 2007

Lecture 5  
Page 45

## Practical Use of Public Key Cryptography

- Keys are created in pairs
- One key is kept secret by the owner
- The other is made public to the world
- If you want to send an encrypted message to someone, encrypt with his public key
  - Only he has private key to decrypt

CS 236, Winter 2007

Lecture 5  
Page 46

## Authentication With Shared Keys

- If only two people know the key, and I didn't create a properly encrypted message -
  - The other guy must have
- But what if he claims he didn't?
- Or what if there are more than two?
- Requires authentication servers

CS 236, Winter 2007

Lecture 5  
Page 47

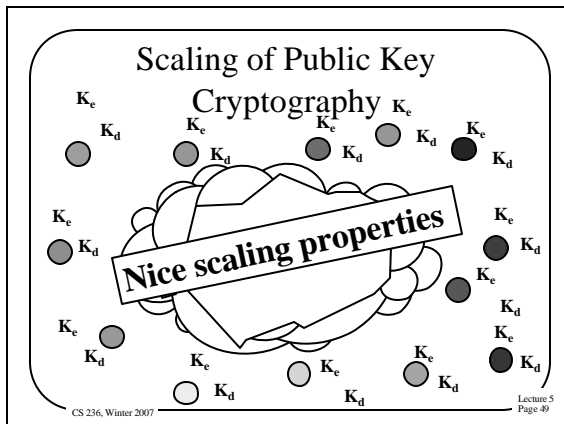
## Authentication With Public Keys

- If I want to "sign" a message, encrypt it with my private key
- Only I know private key, so no one else could create that message
- Everyone knows my public key, so everyone can check my claim directly

CS 236, Winter 2007

Lecture 5  
Page 48





### Key Management Issues

- To communicate via shared key cryptography, key must be distributed
  - In trusted fashion
- To communicate via public key cryptography, need to find out each other's public key
  - “Simply publish public keys”

CS 236, Winter 2007 Lecture 5  
Page 50

### Issues of Key Publication

- Security of public key cryptography depends on using the right public key
- If I am fooled into using the wrong one, that key's owner reads my message
- Need high assurance that a given key belongs to a particular person
- Which requires a *key distribution infrastructure*

CS 236, Winter 2007 Lecture 5  
Page 51

### RSA Algorithm

- Most popular public key cryptographic algorithm
- In wide use
- Has withstood much cryptanalysis
- Based on hard problem of factoring large numbers

CS 236, Winter 2007 Lecture 5  
Page 52

### RSA Keys

- Keys are functions of a pair of 100-200 digit prime numbers
- Relationship between public and private key is complex
- Recovering plaintext without private key (even knowing public key) is supposedly equivalent to factoring product of the prime numbers

CS 236, Winter 2007 Lecture 5  
Page 53

### Comparison of DES and RSA

- DES is much more complex
- However, DES uses only simple arithmetic, logic, and table lookup
- RSA uses exponentiation to large powers
  - Computationally 1000 times more expensive in hardware, 100 times in software
- Key selection also more expensive

CS 236, Winter 2007 Lecture 5  
Page 54

## Security of RSA

- Conjectured that security depends on factoring large numbers
  - But never proven
  - Some variants proven equivalent to factoring problem
- Probably the conjecture is correct

CS 236, Winter 2007

Lecture 5  
Page 55

## Attacks on Factoring RSA Keys

- In 2005, a 640 bit RSA key was successfully factored
  - Took 30 CPU years of 2.2 GHz machines
  - 5 months calendar time
- Research on integer factorization suggests keys up to 2048 bits may be insecure
- Size will keep increasing
- The longer the key, the more expensive the encryption and decryption

CS 236, Winter 2007

Lecture 5  
Page 56

## Combined Use of Symmetric and Asymmetric Cryptography

- Very common to use both in a single session
- Asymmetric cryptography essentially used to “bootstrap” symmetric crypto
- Use RSA (or another PK algorithm) to authenticate and establish a *session key*
- Use DES/Triple DES/AES using session key for the rest of the transmission

CS 236, Winter 2007

Lecture 5  
Page 57

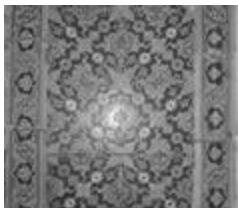
## Steganography

- Another means of hiding data in plain sight
- In general terms, refers to embedding data into some other data
- In modern use, usually hiding data in an image
  - People have talked about using sound and other kinds of data

CS 236, Winter 2007

Lecture 5  
Page 58

## An Example



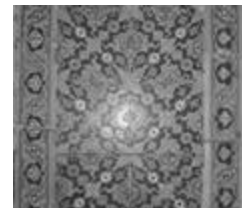
Transfer  
\$100 to my  
savings  
account

Run these through  
outguess

CS 236, Winter 2007

Lecture 5  
Page 59

## Voila!



The one on the right has the message hidden in it

CS 236, Winter 2007

Lecture 5  
Page 60

### How It Works

- Encode the message in the low order bits of the image
- Differences in these bits aren't human-visible
- Other more sophisticated methods also work

CS 236, Winter 2007

Lecture 5  
Page 61

### Detecting Steganography

- Analyze image data to look for unlikely patterns
  - Typically looking in “the likely places”
- Steganographers obviously try to hide these patterns
  - Making bits look like plausible visual data

CS 236, Winter 2007

Lecture 5  
Page 62

### Foiling Steganography

- What if you don't care about the message?
- You just don't want the message to be passed?
- Can try to “wash out” the steganography
- Do your own alteration of the image
  - Again, trying to preserve visual properties
- Again, steganographers try to make their techniques robust against this

CS 236, Winter 2007

Lecture 5  
Page 63

### What's Steganography Good For?

- Not much, as far as we can tell
- Used by some printer manufacturers to prove stuff came from them
- Stories of use by Al-Qaeda
  - No evidence of truth of stories
- Not a lot of evidence it's used for anything serious
- What kinds of things do you think it would be useful for?

CS 236, Winter 2007

Lecture 5  
Page 64