ECE 459/559
Secure & Trustworthy
Computer Hardware Design

Introduction to Cryptography
Part I

Garrett S. Rose

Spring 2017

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Summary

- Substitution Ciphers
- Permutations
- Making good ciphers

Terminology & Background
Threats to Messages

- Interception
- Interruption
  - Blocking messages
- Modification
- Fabrication

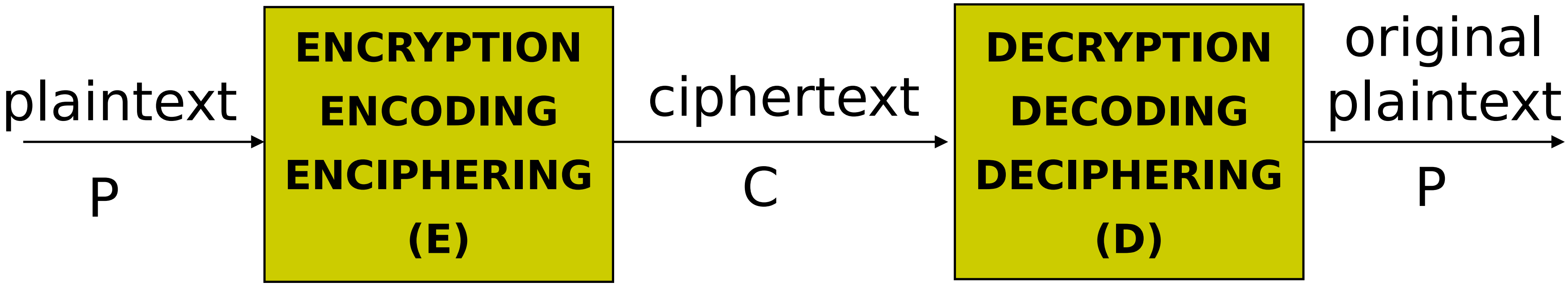
"A threat is blocked by control of a vulnerability"

[Pfleeger & Pfleeger]

Basic Terminology & Notation

- Cryptology
  - Cryptography + Cryptanalysis
- Cryptography
  - Art/science of keeping messages secure
- Cryptanalysis
  - Art/science of breaking ciphertext
  - Ex.: Enigma in WWII

Basic Cryptographic Scheme

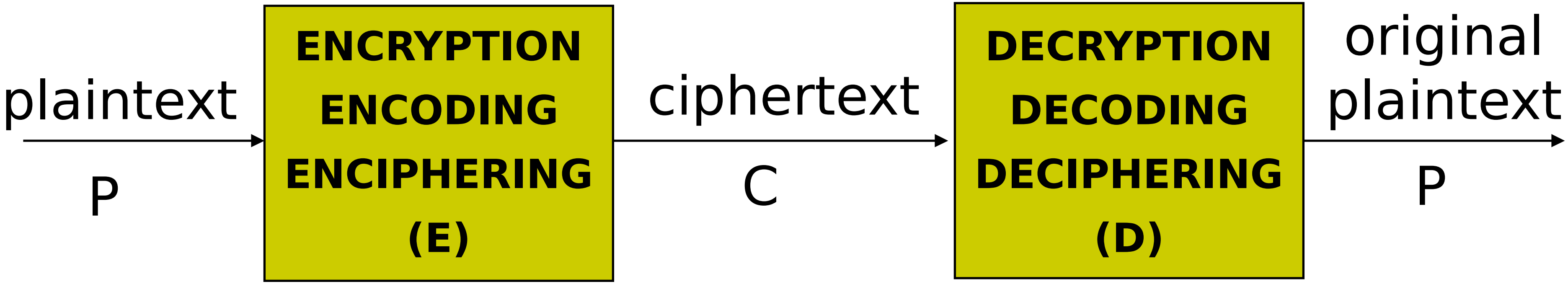


- $P = \langle p_1, p_2, \ldots, p_n \rangle$ $\quad$ p_i = i-th character of P
  - P = “DO NOT TELL ANYBODY” $\quad$ p_1 = “D”, p_2 = “O”, etc.
  - By convention, plaintext (a.k.a. “cleartext”) in uppercase
- $C = \langle c_1, c_2, \ldots, c_n \rangle$ $\quad$ c_i = i-th character of C
  - C = “ep opu ufmm bozcpez” $\quad$ c_1 = “e”, c_2 = “p”, etc.
  - By convention, ciphertext in lowercase

Benefits of Cryptography

- An improvement, not a solution!
- Minimizes problems
- Doesn't solve them all
  - Remember: There is *no* solution!
- Adds an envelope (encoding) to an open postcard (plaintext)

Formal Notation

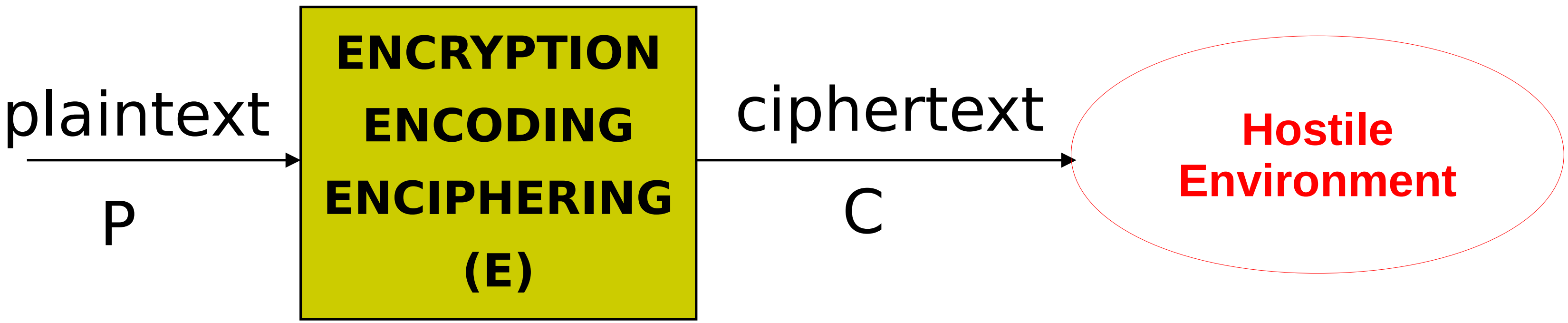


- C = E(P)   E – encryption rule/algorithm
- P = D(C   )   D – decryption rule/algorithm

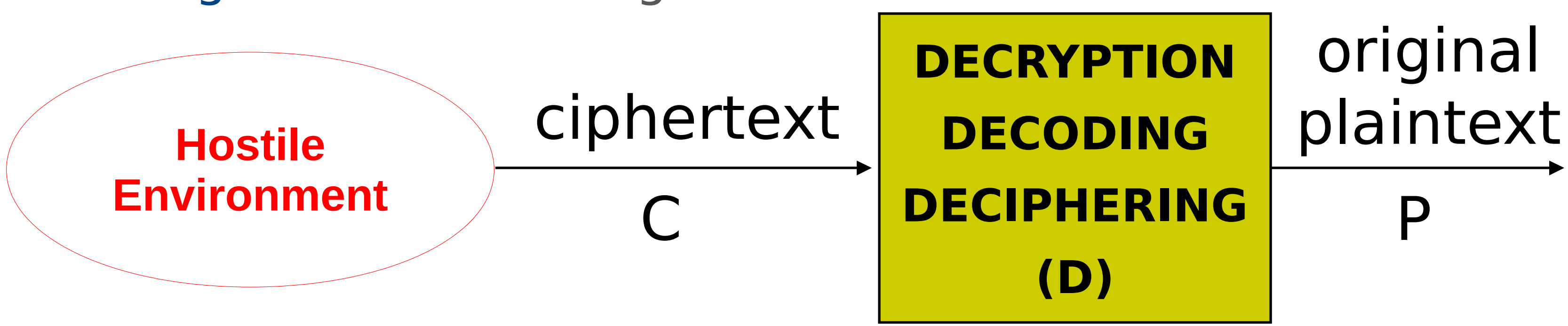
- Need cryptosystem, where:
  - P = D(C) = D(E(P))
  - i.e., able to get original message back

Cryptography in Practice

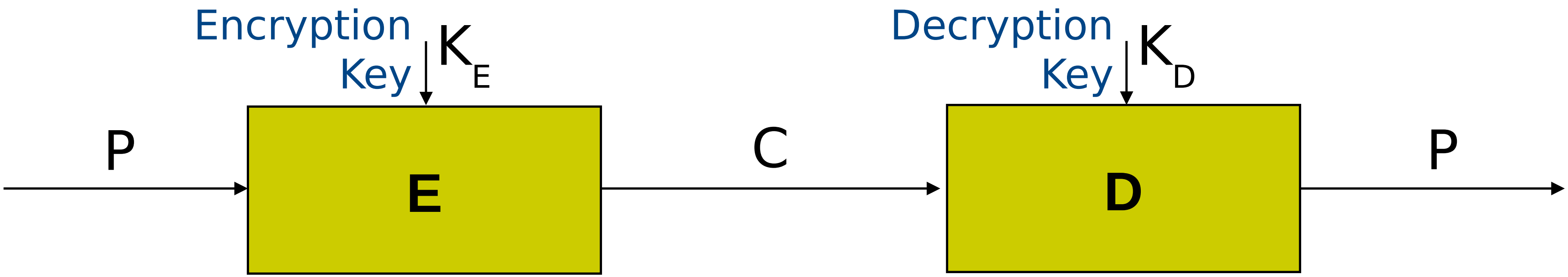
- Sending a secure message



- Receiving a secure message



Cryptosystem with Keys



- $C = E(K_E, P)$
  - E = set of encryption algorithms / K_E selects $E_i \in E$
- $P = D(K_D, C)$
  - D = set of decryption algorithms / K_D selects $D_i \in D$
- Crypto algorithms and keys are like door locks and keys
- We need: $P = D(K_D, E(K_E, P))$

Classification of Crytosystems (w.r.t. Keys)

- **Keyless cryptosystems** exist (e.g. Caesar's Cipher)
  - Less secure
- **Symmetric cryptosystems**: $K_E = K_D$
  - Classic
  - Encipher and decipher using same key
    (or one key is easily derived from other)
- **Asymmetric cryptosystems**: $K_E \neq K_D$
  - Public key system
  - Encrypt and decrypt using different keys
    (computationally infeasible to derive one from other)

Cryptanalysis Goals

- Break a single message
- Recognize patterns in encrypted messages
  – gain ability to break subsequent messages
- Infer meaning without breaking encryption
  - Unusual volume of messages between troops may indicate attack
  - Busiest node may be enemy headquarters
- Deduce the key, facilitate breaking subsequent messages
- Find vulnerabilities in implementation or environment of an encryption algorithm
- Find general weakness in encryption algorithm

Information for Cryptanalysts

- Intercepted encrypted messages
- Known encryption algorithms
- Intercepted plaintext
- Data known or suspected to be ciphertext
- Math or statistical tools and techniques
- Properties of natural languages
  - Especially adversary's natural language
  - To confuse enemy, U.S. used Navajo language in WWII
- Properties of computer systems
- Ingenuity and ol' fashioned luck
- There are *no* rules!

Breakable Encryption

- Theoretically, it is possible to device unbreakable cryptosystems
- Practical cryptosystems almost always breakable,
  given adequate time and computing resources
- Trick is to make breaking cryptosystem <u>hard enough</u> for intruder

Example of Breakable Encryption

- Breakability of encryption algorithm – message uses 25 characters
- 26^{25} possible decryptions ~ 10^{35} decryptions
- Only one is correct
- Brute force approach:
  - 10^{10} (10 bln) decryption/sec => $10^{35}/10^{10} = 10^{15}$ sec = 32 mln yrs!
  - Infeasible with current technology
- Be smarter – use ingenuity
  - Could reduce 26^{25} to, say, 10^{15} decryptions to check
  - 10^{10} decryption/sec => $10^{15}/10^{10} = 10^{5}$ sec ≈ 1 day

Requirements of Crypto Protocols

- Messages should get to destination
- Only recipient should get it
- Only recipient should see it
- Need proof of sender's identity
- Message shouldn't be corrupted in transit
- Message should be sent/received once

- Proofs that message was sent/received (non-repudiation)

Representing Characters

- Letters (uppercase only) represented by numbers 0-25 (modulo 26)

```
A   B   C   D   …   X   Y   Z

0   1   2   3       23  24  25
```

- Operations on letters:

```
A + 2 = C

X + 4 = B (circular)

...
```

Basic Types of Ciphers

- Substitution ciphers
  - Letters of P replaced with other letters by E
- Transportation (permutation) ciphers
  - Order of letters of P rearranged by E
- Product ciphers
  - E "=" E_1 "+" E_2 "+" … "+" E_n
  - Combine two or more ciphers to enhance security

The Caesar Cipher (A Substitution Cipher)

- $c_i = E(p_i) = (p_i + 3) \bmod 26$ (26 letters in English alphabet)
  - Change each letter to third letter following it (circularly)

    A → D, B → E, ..., X → A, Y → B, Z → C
- Can represent as permutation of π: π(i) = (i + 3) mod 26

  π(0) = 3, π(1) = 4, ..., π(23) = 26 mod 26 = 0, π(24) = 1, π(25) = 2
- Key = 3 OR key = 'D' (since D represents 3)

The Caesar Cipher

- Example:
  - P (plaintext): `HELLO WORLD`
  - C (ciphertext): `khoor zruog`
- Caesar Cipher is a monoalphabetic substitution cipher (= simple substitution cipher)
  - One key is used
  - One letter substitutes the letter in P

Attacking a Substitution Cipher

- Exhaustive search
  - If key space small, try all possible keys until you find right one
  - Caesar cipher has only 26 possible keys

    from A to Z OR from 0 to 25
- Statistical analysis (attack)
  - Compare to so called 1-gram (unigram) model of English
  - Shows frequency of each character used in English language
  - The longer the C, the more effective statistical analysis is

1-grams (Unigrams) for English

a	0.080	h	0.060	n	0.070	t	0.090
b	0.015	i	0.065	o	0.080	u	0.030
c	0.030	j	0.005	p	0.020	v	0.010
d	0.040	k	0.005	q	0.002	w	0.015
e	0.130	l	0.035	r	0.065	x	0.005
f	0.020	m	0.030	s	0.060	y	0.020
g	0.015					z	0.002

Statistical Attack

- Compute frequency $f(c_i)$ for each letter c_i used in ciphertext
- Example: C = “khoor zruog”
  - 10 characters: 3x 'o', 2x 'r', 1x {'k', 'h', 'z', 'u', 'g'}
  - $f(c_i)$:

$f('g') = 0.1$ $f('h') = 0.1$ $f('k') = 0.1$ $f('o') = 0.3$ $f('r') = 0.2$

$f('u') = 0.1$ $f('z') = 0.1$ $f(c_i) = 0$ for any other c_i

- Apply 1-gram model of English
  - Frequency of usage for characters in language
  - 1-grams provided on previous slide

Statistical Attack – Step 2

- Phi φ(i) – correlation of frequency of letters in ciphertext with frequency of corresponding letters in alphabet – given key I
- For key i: $\varphi(i) = \Sigma_{0 \leq c \leq 25}\ f(c) \bullet p(c - I)$
  - c representation of character (a=0, …, z=25)
  - f(c) is frequency of character x in English
  - Intuition: sum of probabilities for words in P, if i were the key

Statistical Attack – Step 2

- Example: C = “khoor zruog” (P = “HELLO WORLD”)

  f(c):

  f('g') = 0.1 f('h') = 0.1 f('k') = 0.1 f('o') = 0.3 f('r') = 0.2

  f('u') = 0.1 f('z') = 0.1 $f(c_i) = 0$ for any other c_i

  c:

  g – 6, h – 7, k – 10, o – 14, r – 17, u – 20, z – 25

- $\varphi(i) = 0.1 \bullet p(6 - i) + 0.1 \bullet p(7 - i) + 0.1 \bullet p(10 - i) +$

  $0.3 \bullet p(14 - i) + 0.2 \bullet p(17 - i) + 0.1 \bullet p(20 - i) +$

  $0.1 \bullet p(25 - i)$

Statistical Attack – Step 2 (Calculations)

i	φ(*i*)	*i*	φ(*i*)	*i*	φ(*i*)	*i*	φ(*i*)
0	0.0482	7	0.0442	13	0.0520	19	0.0315
1	0.0364	8	0.0202	14	0.0535	20	0.0302
2	0.0410	9	0.0267	15	0.0226	21	0.0517
3	0.0575	10	0.0635	16	0.0322	22	0.0380
4	0.0252	11	0.0262	17	0.0392	23	0.0370
5	0.0190	12	0.0325	18	0.0299	24	0.0316
6	0.0660					25	0.0430

Statistical Attack – Step 3 (The Result)

- Most probably keys (largest φ(i) values):
  - I = 6, φ(i) = 0.0660
    plaintext: EBIIL TLOLA
  - I = 10, φ(i) = 0.0635
    plaintext: AXEEH PHKEW
  - I = 3, φ(i) = 0.0575
    plaintext: HELLO WORLD
  - I = 14, φ(i) = 0.0535
    plaintext: WTAAD LDGAS
- Only English phrase is for i = 3
  - That's the key (3 or 'D') – code broken

Caesar's Problem

- Conclusion: Key is too short
  - 1-character key – monoalphabetic substitution
  - Can be found by exhaustive search
  - Statistical frequencies not concealed well by short key – Looks too much like “regular” English letters
- Solution: May the key longer
  - N-character ($n \geq 2$) – polyalphabetic substitution
  - Makes exhaustive search much more difficult
  - Statistical frequencies concealed much better