Electronics and Computer Science
University of Southampton

CM214-COMP2008
Data Communications and Networks

Network Security - 1

Karl R. Wilcox
krw@ecs.soton.ac.uk

Objectives

- To look at definitions of a “secure” network
  - To consider various aspects of security
  - To look at tools and techniques

- (Peterson & Davie, Sections 8.1-3)

Background Reading

- “Security Complete”
  - Sybex, ≈£13, paperback, ≈900 pages
  - Comprehensive, practical, dry

- “Cryptonomicon”
  - Neal Stephenson, ≈ £10 paperback, ≈ 900 pages
  - Practical, interesting + good stody

What is "Security"? - 1

- That content of data in a network transfer remains private between sender & reciever(s)
  - Encryption
- That content of data in a network transfer has been altered during transmission
  - Message Integrity

What is ”Security”? – 2

- That one or all parties in a network transfer can be assured of the identities of the other parties
  - Authentication
- That it can be proven that particular data was transferred between particular parties at a particular time
  - Non-repudiation

An Alternative "Secure" Network

- All network transfers between any parties are untraceable with complete deniability(!)
  - E.g. A peer-to-peer file swapping network

Security is NOT

- Encryption
  - Although encryption plays a part
- Security is not the same as resilience
  - Although often confused / misused
- Resilience is resistance to failure
  - (including that caused by a deliberate attack)

Security Is…

- Different things to different people
- A system built of components
  - We will look at tools / technologies
  - None are the single answer to "security"
- System is as secure the weakest link
  - e-mail encrypted with 4096 bit RSA is not secure if it can be read over your shoulder

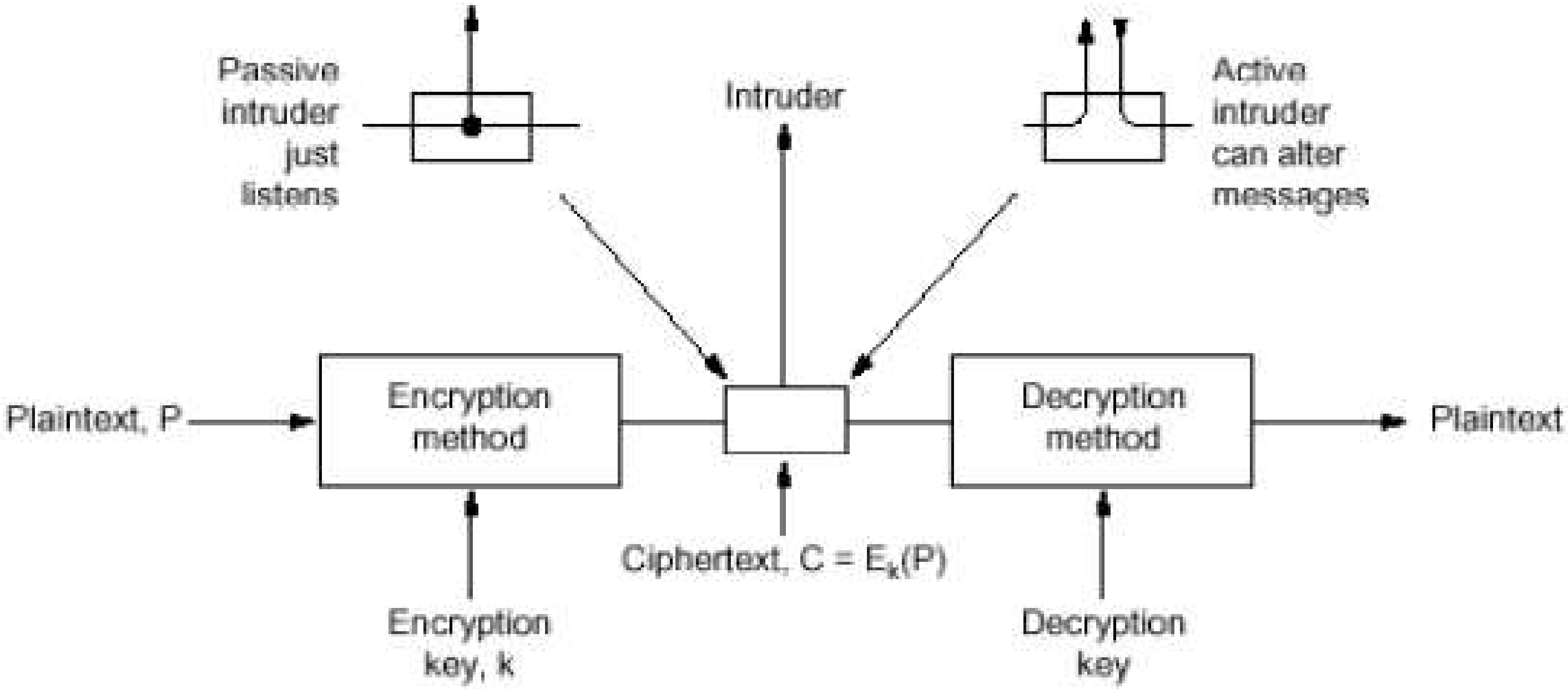
Building a Secure System

- Determine the required features
- Choose the tools
  - E.g. DES, IDEA, RSA, MD5
- Choose the techniques
  - E.g. Authentication, digital signatures, key distribution
- Or the applications
  - SSL, SSH, PGP

Security Threats

- Eavesdropping
- Impersonation
  - Person
  - Address
  - Computer
- Message duplication
- Key interception
- Cracking
  - Known plaintext
  - Brute force
- Line security
- Social engineering

Security Model



By convention, the parties in a transfer are known as “Alice” and “Bob”

Traditional Cryptography

- Substitution / transposition cyphers
- Little use for network security
- Too open to attack
  - Known plaintext
  - Letter frequency
  - Brute force

One Time Pads

- Traditional, totally secure method
- Plain text of P bits
- Key K_B of at least B bits
- Coded message is $C = K_B \; xor \; P$
- Decode by applying same key

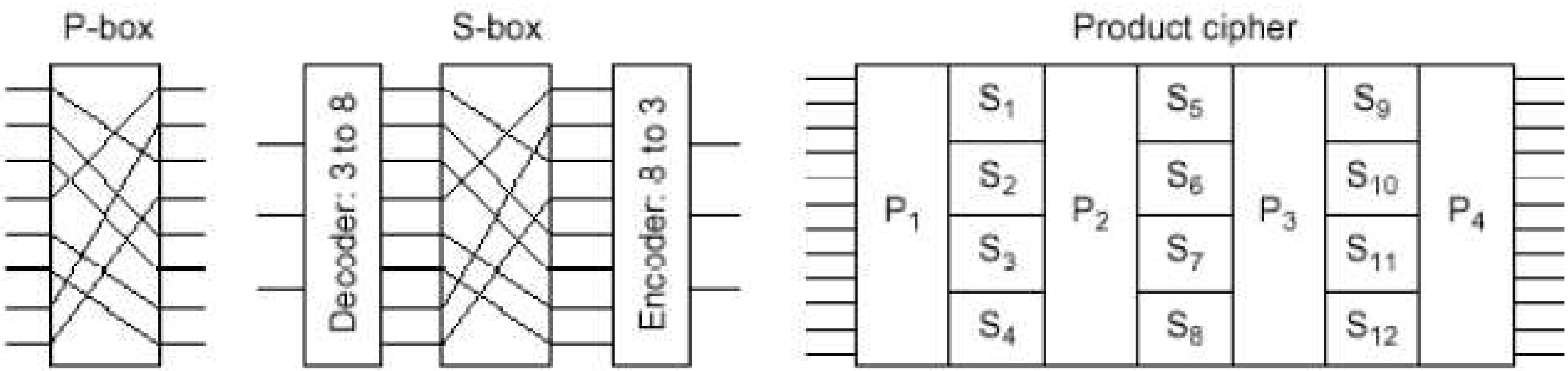
One Time Pad Requirements

- Totally secure & unbreakable if:
  - Keys are genuinely random
  - The pad remains secret
  - It is never re-used
- Sounds ideal (simple, unbreakable)
  - Key distribution is the problem
  - Need to securely share as much key information as message information

Network Cryptography

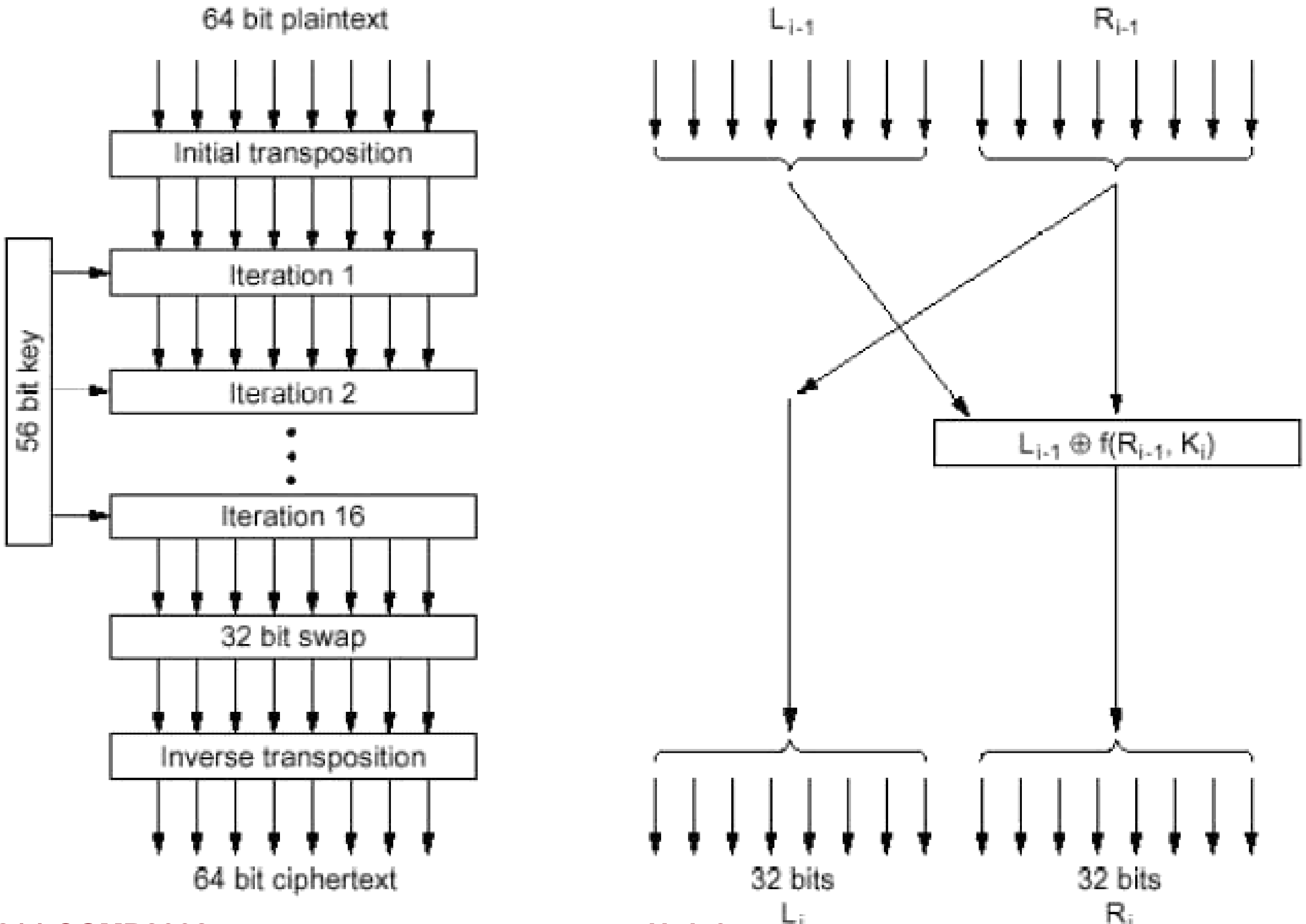
- Uses simpler keys (shorter than plaintext)
- Complex algorithms
  - Lots of iterations
  - Permuting bits
  - Substituting bit sequences

Cryptography Elements



- Easy to implement in hardware
- Reasonably easy to implement in software
  - But computationally intensive

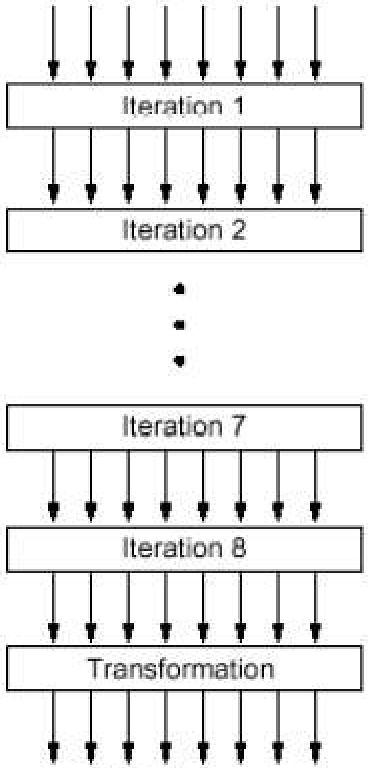
Data Encryption Standard

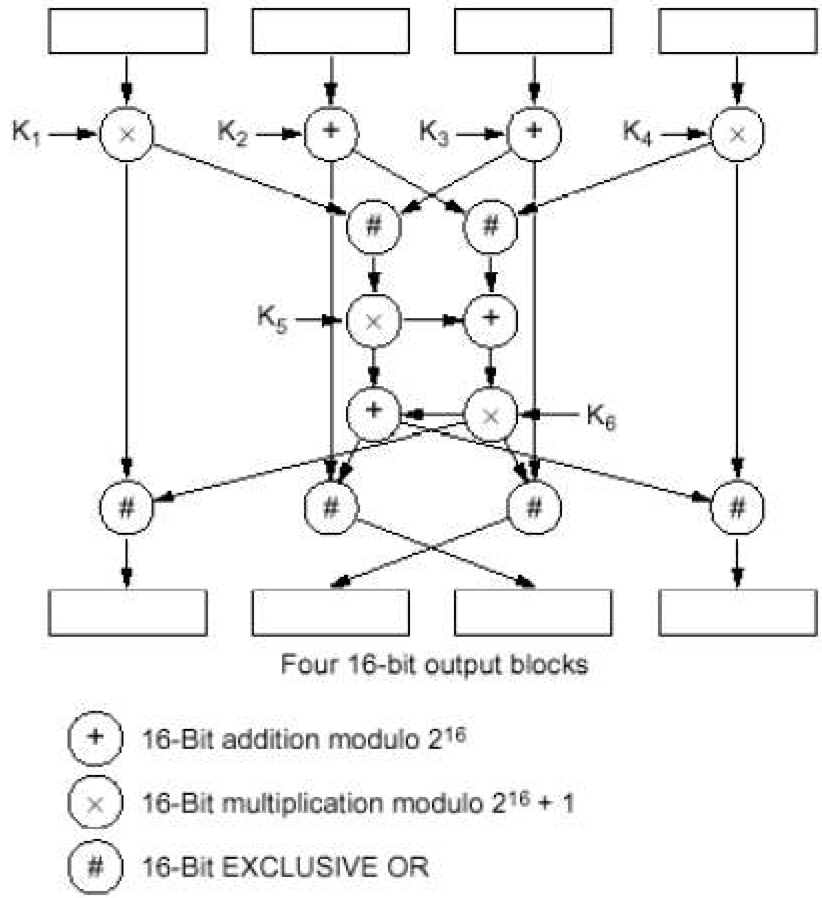


How Secure is DES?

- Can be broken
  - By brute force in a “reasonable” time
    - 1999 $250K computer 22 hours
  - Even faster with some known plain text
- Triple DES
  - Apply 3 times, 2 keys, 112 encryption
    - Cannot be broken in “reasonable” time

International Data Encryption Standard





Encryption Keys

- Private keys (symmetric encryption)
  - Known only to involved parties
  - How to distribute?
- Public keys (asymmetric encryption)
  - One key is public (published)
  - One key remains private
  - Needs 2 pairs of keys for two-way communication

Public Key Algorithms

- Use two complementary keys K_E & K_D
- $D(E(P)) = P$
- Need to ensure:
  - D not easily generated from E
  - E cannot be broken by plain text attack
- E is public, D **must** remain private
- Any algorithm meeting these criteria will work

Rivest, Shamir, Alderman (RSA) Algorithm

1. Take two (100+ digit) prime numbers p and q
2. Calculate $n = pq$, $z = (p - 1)(q - 1)$
3. Choose d relatively prime to z
4. Find e such that $ed = 1 \bmod z$

To encode:

$C = P ** e \bmod n$

To decode:

$P = C ** d \bmod n$

RSA Features

- Publish ***e*** and ***n***
- Keep ***d*** private

- Note, if we can factorise ***n*** we can break encryption
  - Fortunately ***n*** is very, very large

RSA Example

- Pick two prime numbers, $p = 3$, $q = 11$
- This gives $n = 33$, $z = 20$
- Choose a number d with no common divisors with 20, say 7
- We want to find another number e that when multiplied by d divided modulo 20 leaves remainder 1
  - $e = 3$

RSA Calculations

Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	P^3 (mod 33)	C^7	C^7 (mod 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	1	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	5	E

Sender's computation (Symbolic through P^3 (mod 33)); Receiver's computation (P^3 (mod 33) through Symbolic)

- Even with “toy” example consider large numbers involved
  - Infinite precision integer arithmetic

Message Digests

- Encryption is computationally intensive
- May not be primary aim
  - May wish to ensure message not tampered with in transit
- Need a “one-way” function between plain text and a (shorter) bit string
  - i.e. can generate bit string from plain text but not vice versa
  - Still need to encrypt digest

Signatures

- An encrypted message digest also acts as a digital signature
  - The message is sent in plain text
  - Only the digest is encrypted
  - Only this message could generate the digest
  - Only the sender could have encrypted the digest

Summary

- Encryption has more than one purpose
- And more than one way of implementation

- Self test – You wish to securely transmit a very long message
  – Should you compress or encrypt first?