CSC 519 Information Security

LECTURE 4: Cryptography

Dr. Esam A. Alwagait
alwagait@ksu.edu.sa

Recap form previous Lecture

- We discussed more symmetric encryption.
- Books ?
  - Security Engineering, Ross Anderson
    (available online @ http://www.cl.cam.ac.uk/~rja14/book.html)
  - Security in Computing, 4th Edition by Charles P. Pfleeger
  - Computer Security, 3rd Edition by Dieter Gollmann, Wiley, 2011
- Decryption example ! (Vernam Cipher)

One-Time Pads: Vernam Cipher

- The letters would first be converted to their numeric equivalents

V	E	R	N	A	M	C	I	P	H	E	R
21	4	17	13	0	12	2	8	15	7	4	17

- Next, we generate random numbers
  - 76 48 16 82 44 03 58 11 60 05 48 88
- The encoded message is the sum mod 26 of each coded letter with the random number

- V = 21 → + 76 = 97 → Mod 26 = 19 → T
- T= 19 → -76 = -57 → Mod 26 = -31 → Mod 26 =-5 → Mod 26 =21 =V

Plaintext	V	E	R	N	A	M	C	I	P	H	E	R
Numeric Equivalent	21	4	17	13	0	12	2	8	15	7	4	17
+ Random Number	76	48	16	82	44	3	58	11	60	5	48	88
= Sum	97	52	33	95	44	15	60	19	75	12	52	105
= mod 26	19	0	7	17	18	15	8	19	23	12	0	1
Ciphertext	t	a	h	r	s	p	i	t	x	m	a	b

The message VERNAM CIPHER is encoded as tahrsp itxmab

Cryptanalysis of transposition

- Recall that transpositions leave the plaintext letters intact, so the work for the cryptanalyst is more **exhausting**, more relies on a human's judgment of what looks right!
- Letter frequency
  - Look for a match, then break it into columns

Cryptanalysis of transposition

Five-column transposition example

Encryption tips:

- Write the message in rows with 5 letters for each row, then write the ciphertext using letters in columns
- Use an infrequent letter, such as X, to fill in any short columns

Decryption tip:

- Organize the ciphertext into groups based on dividing the total number of letters (50 here) by 5 (column type)

T H I S I
S A M E S
S A G E T
O S H O W
H O W A C
O L U M N
A R T R A
N S P O S
I T I O N
W O R K S

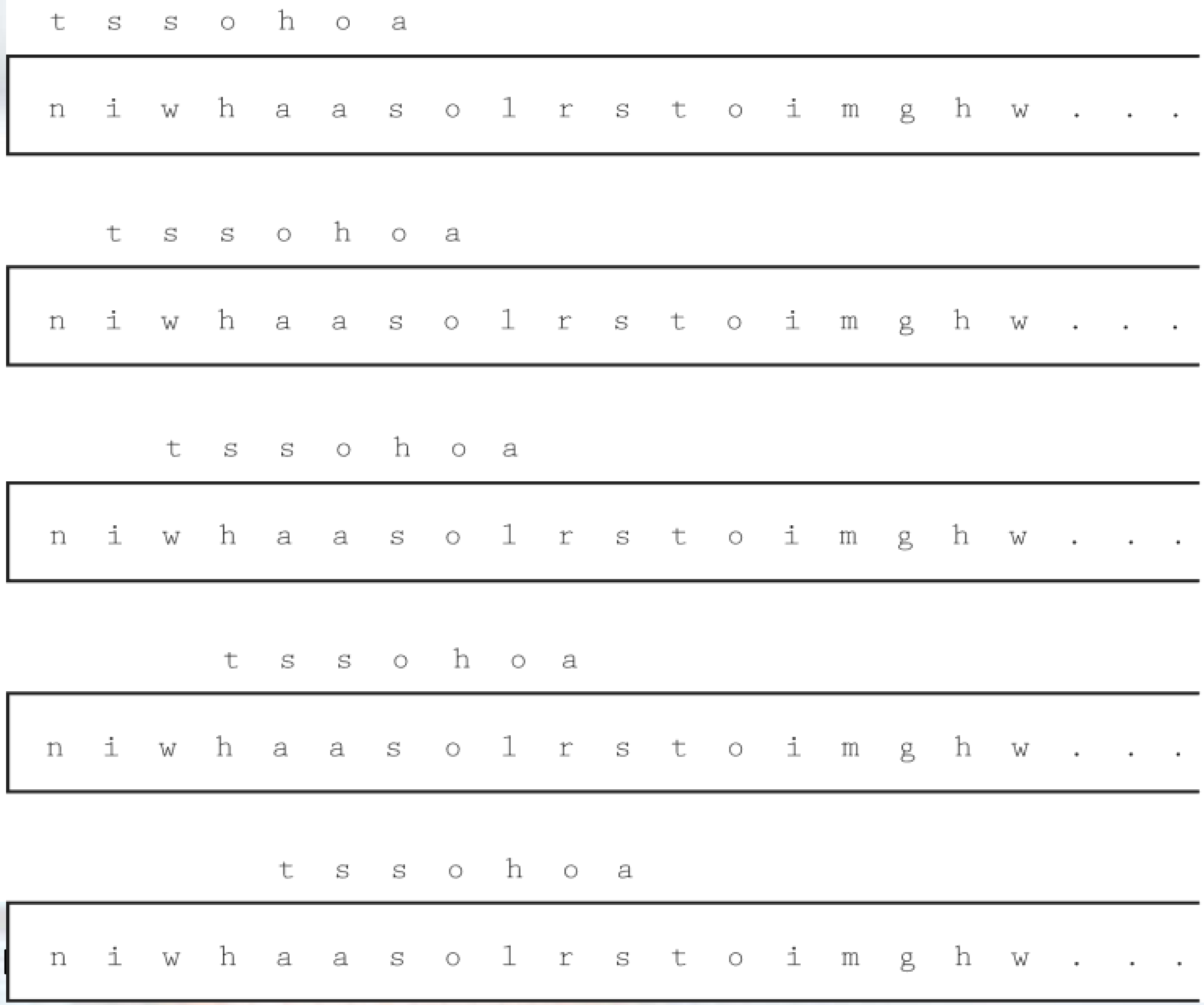
⇄

Ciphertext (50 characters):

tssoh oaniw haaso lrsto imghw
utpir seeoa mrook istwc nasns

Moving comparisons

- Start with a moving window (say 7)
- Compare c_1 to c_8, c_2 to c_9, …, c_7 to c_{14}
- Look for digrams?
- Do they look reasonable?
- Move the window, exhaustively searching all possibilities of window size 7
- Then try a distance of 8, repeat the steps above!



What does it mean for a cipher to be "good"?

Claude Shannon proposed several characteristics (in 1949):

- The amount of secrecy needed should determine the amount of labor appropriate for the encryption and decryption
- Keys and algorithm should not be complex!
  - Algorithms that work on specific text are useless
  - Restrictions complicate the use
- Simple implementation (w.r.t. time and space complexity)
  - A complicated algorithm is prone to programming errors
- Enciphering errors should not propagate
  - So it doesn’t cause corruption of remaining characters
- Ciphertext size should be no longer than the plaintext
  - More space and transmission time, prone to inference
  - without carrying more information

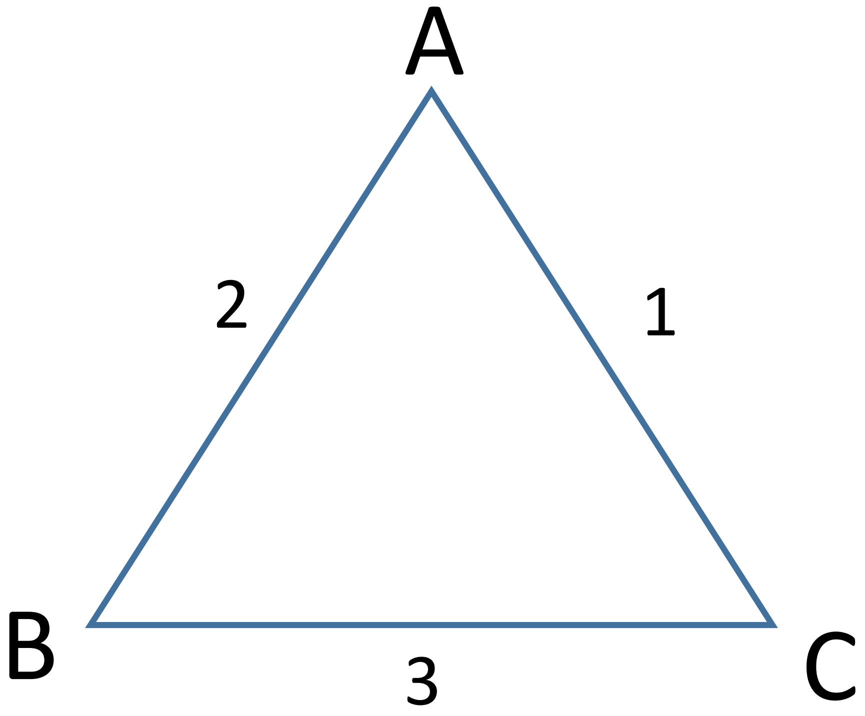
Symmetric and Asymmetric Encryption Systems

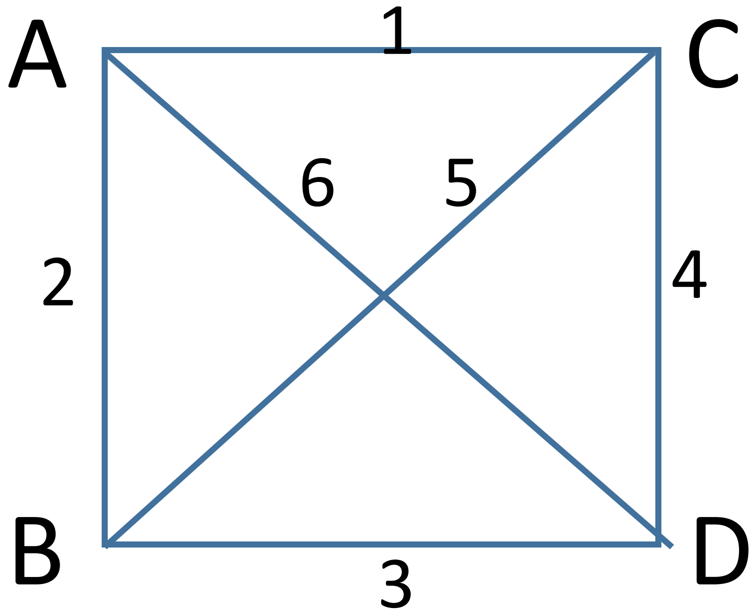
- Symmetric (also called "secret key“ or “private key”)
  - Uses one key for both encryption and decryption
- Asymmetric (also called "public key")
  - Uses two different keys, one is a private key and the other is a public key

Symmetric Encryption Systems

- Symmetric algorithms use one key for both encryption and decryption
  - The decryption algorithm is closely related to the encryption one
  - For example, the Caesar cipher with a shift of 3 uses
    - For encryption algorithm "substitute the character three letters later in the alphabet"
    - For decryption "substitute the character three letters earlier in the alphabet”
- It provides a two-way channel to users: A and B share a secret key
- How about multiple users?
- Assume A wants to share secrets with B and C as well? How many keys we need?

Symmetric Encryption Systems



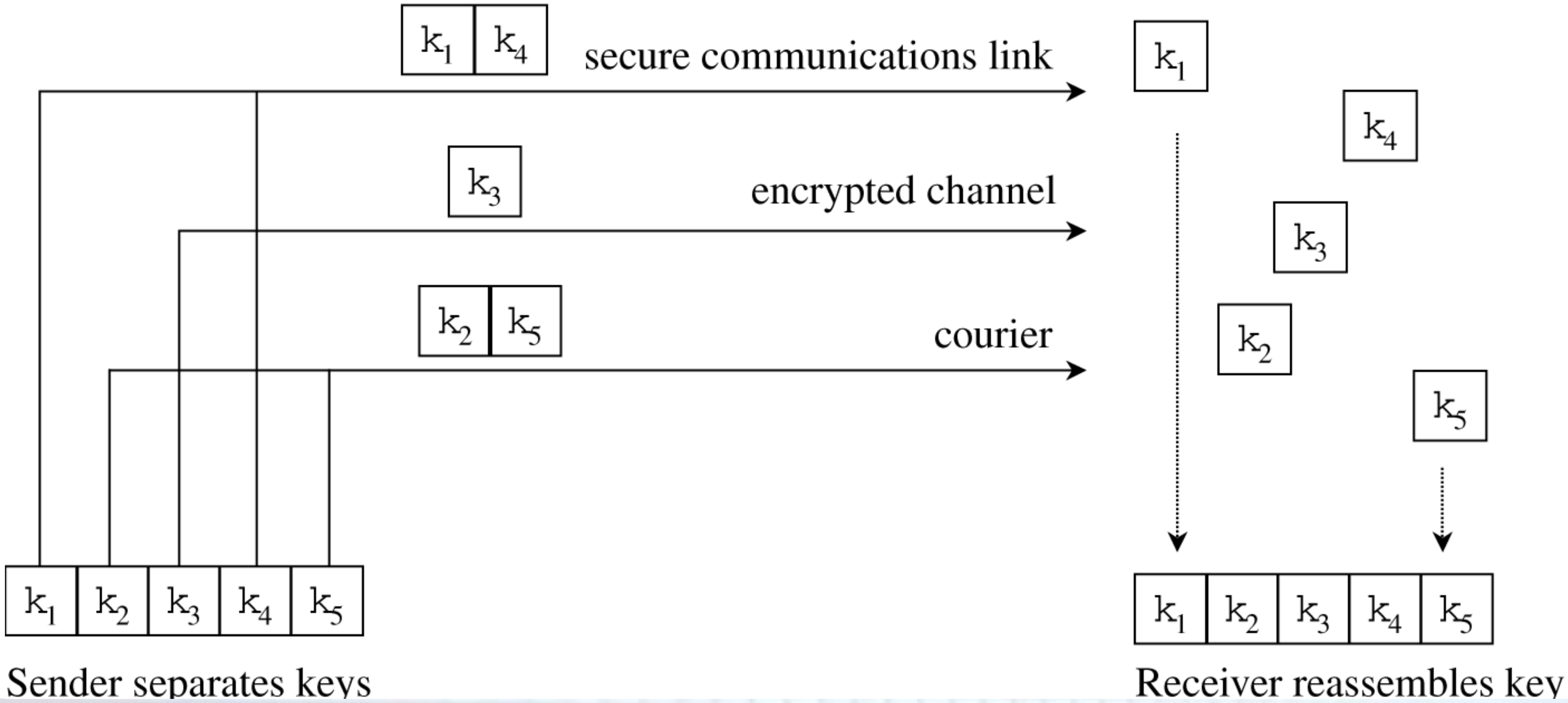


- In a situation of with 1000 users, you would need 499,500 keys!

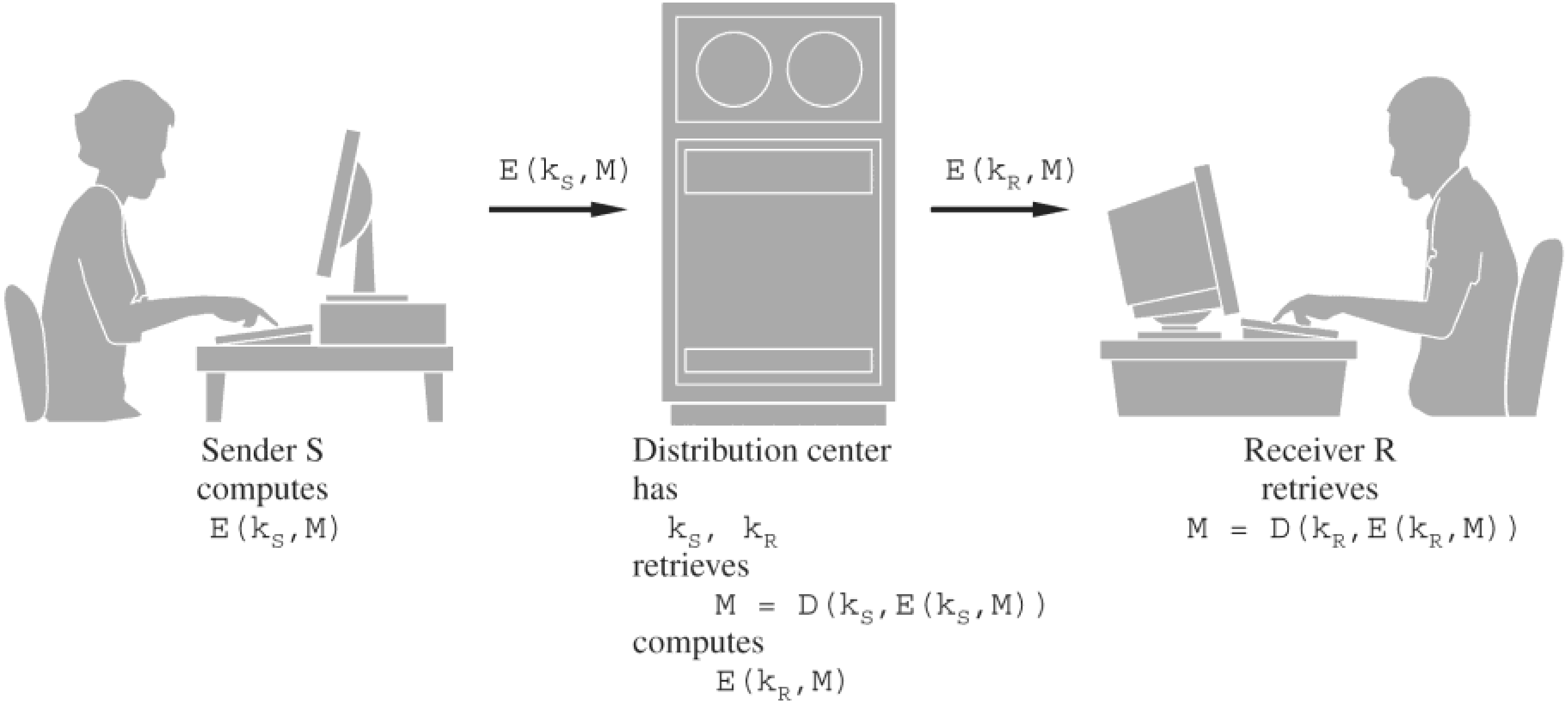
Symmetric Encryption Systems

- How about *n* users who want to communicate in pairs?
- We need $n*(n - 1)/2$ keys
- So, the number of keys needed increases at a rate proportional to the square of the number of users
- Two major issues:
  - Key distribution problem in symmetric systems
    - how about public key systems?
  - Also, the key must be kept secret, i.e., key management issue!
    - Storing, safeguarding, and activating keys

Key Distribution in Pieces



Distribution Center for Encrypted Information



Pfleeger/Pfleeger Fig. 10-06

Asymmetric Encryption Systems (Public Key Encryption)

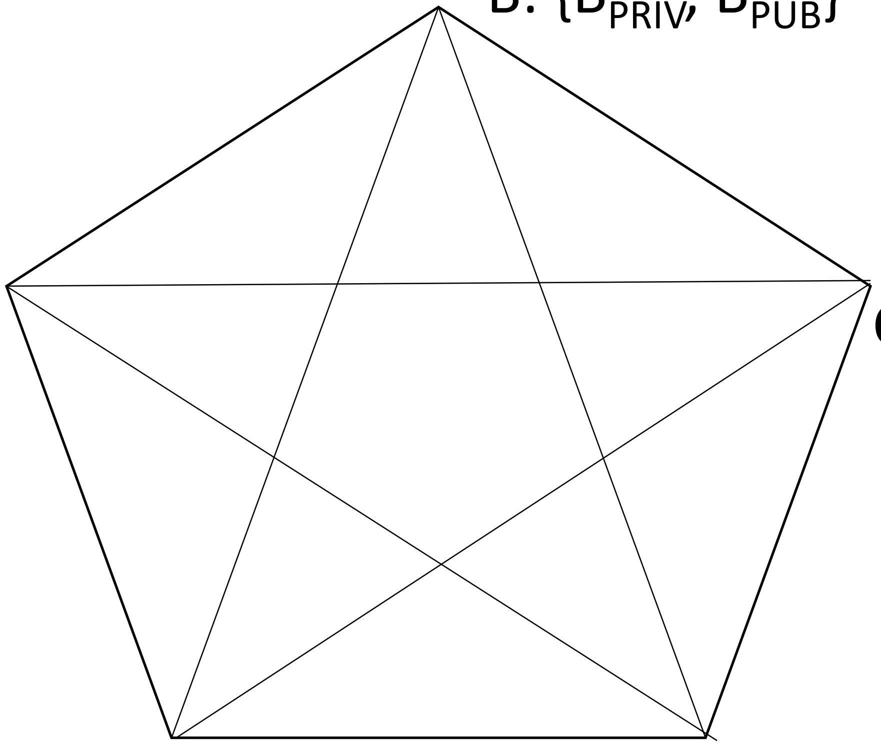
- In 1976, proposed by Diffie and Hellman proposed, a new kind of encryption system has emerged
- Allowing the key to be public, but still protect the message!
- Each user has two keys: a public key and a private key
- The user may freely publish the public key but not the private key
- The keys operate as inverses, meaning that one key “undoes” the encryption provided by the other key

Public Key Encryption

- Let k_{PRIV} be a user's private key, and let k_{PUB} be the corresponding public key
- The encrypted plaintext using the public key is decrypted by application of the private key
  - $P = D(k_{PRIV}, E(k_{PUB}, P))$
  - Here, the user decrypts with a private key what has been encrypted with the corresponding public key
- The encrypted message using the private key is decrypted by application of the public key
  - $P = D(k_{PUB}, E(k_{PRIV}, P))$

Public Key Encryption

- Only two keys are required per user!



Comparing Secret Key and Public Key Encryption

	Secret Key (Symmetric)	Public Key (Asymmetric)
Number of keys	1	2
Protection of key	Must be kept secret	One key must be kept secret, the other can be freely exposed
Best uses	Cryptographic workhorse, secrecy and integrity of data single characters to blocks of data, messages, files	Key exchange, authentication
Key distribution	Must be out-of-band	Public key can be used to distribute other keys
Speed	Fast	Slow, typically, 10,000 times slower than secret key

Rivest Shamir Adelman Encryption Algorithm

- The RSA is a public cryptosystem, introduced in 1978, that remains secure until today
- Its mathematical foundation is based on number theory
  - Specifically, determining the prime factors of a given number (large number)
    - This problem is called the factorization problem
  - Also, uses modular arithmetic, i.e., arithmetic mod n

Rivest Shamir Adelman Encryption (Key Setup)

- each user generates a public/private key pair by:
- selecting two large primes at random p, q
- computing their system modulus $n=p.q$
- Computing $\varnothing(n)=(p-1)(q-1)$
- selecting at random the encryption key e
  - where $1<e<\varnothing(n)$, $\gcd(e,\varnothing(n))=1$
- solve following equation to find decryption key d
  - $e.d=1 \bmod \varnothing(n)$ and $0 \le d \le n$
- publish their public encryption key: PU={e,n}
- keep secret private decryption key: PR={d,n}

RivestShamirAdelman Encryption

- Two keys are used, say *d* and *e* that are interchangeable for encryption/decryption processes
  - P = E(D(P)) = D(E(P))
- Any plaintext block P is encrypted as $\mathrm{C} = P^e \; mod \; n$
- The decryption is performed by $P = (P^e)^d \; mod \; n$
- Because the exponentiation is performed mod n, factoring P^e to uncover the encrypted plaintext is a difficult problem
- The decrypting key d is carefully chosen so that P can be recovered, without factoring P^e
- Note that the message M must be smaller than the modulus n

Rivest Shamir Adelman Encryption: Use

- to encrypt a message M the sender:
  - obtains **public key** of recipient $PU=\{e,n\}$
  - computes: $C = M^e \bmod n$, where $0 \le M < n$
- to decrypt the ciphertext C the owner:
  - uses their private key $PR=\{d,n\}$
  - computes: $M = C^d \bmod n$
- note that the message M must be smaller than the modulus n

RSA Example - Key Setup

1. Select primes: p=17 & q=11
2. Compute $n = pq$ =17 x 11=187
3. Compute $\varnothing(n)=(p-1)(q-1)$=16 x 10=160
4. Select e: gcd(e,160)=1; choose e=7
5. Determine d: de=1 mod 160 and d < 160 Value is d=23 since 23x7=161= 1x160+1
6. Publish public key PU={7,187}
7. Keep secret private key PR={23,187}

RSA Example - En/Decryption

- Sample RSA encryption/decryption is:
- Given message $M = 88$ (nb. $88<187$)
- encryption:

  $C = 88^7 \bmod 187 = 11$
- decryption:

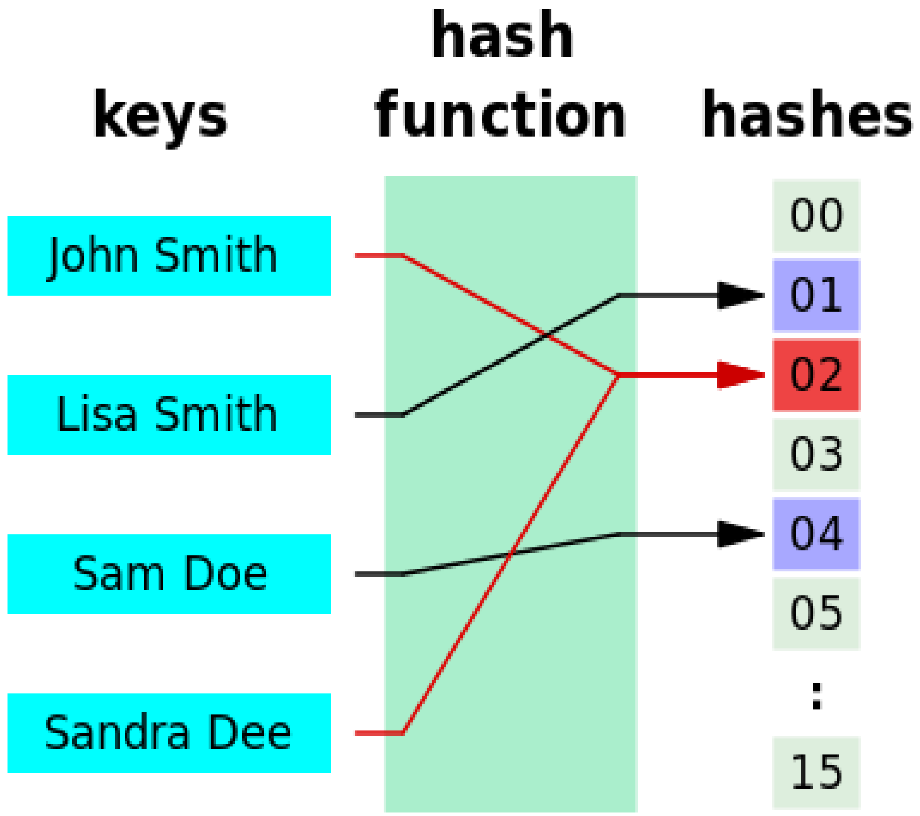
  $M = 11^{23} \bmod 187 = 88$

The Uses of Encryption

- The combined use of symmetric and asymmetric encryption leverages the best features of both
- Four different uses in cryptography:
  - Cryptographic hash functions
  - Key exchange
  - Digital signatures
  - Certificates

Cryptographic Hash Functions

- A hash function is any algorithm that maps data of arbitrary length to data of a fixed length
- Typically not invertible, meaning that it is not possible to reconstruct the input datum *x* from its hash value *h(x)* alone
- Used to ensure integrity of the message
- Sometimes it is more important than secrecy of the message?
  - Example: retrieval of legal documents, other examples?
- Digitally sealing a file so that any alteration can be detected
- Widely-used algorithms:
  - MD4/MD5 (by Ron Rivest and RSA Laboratories)
    - Produces 128-bit digest
  - SHA/SHS
    - Produces 160-bit digest



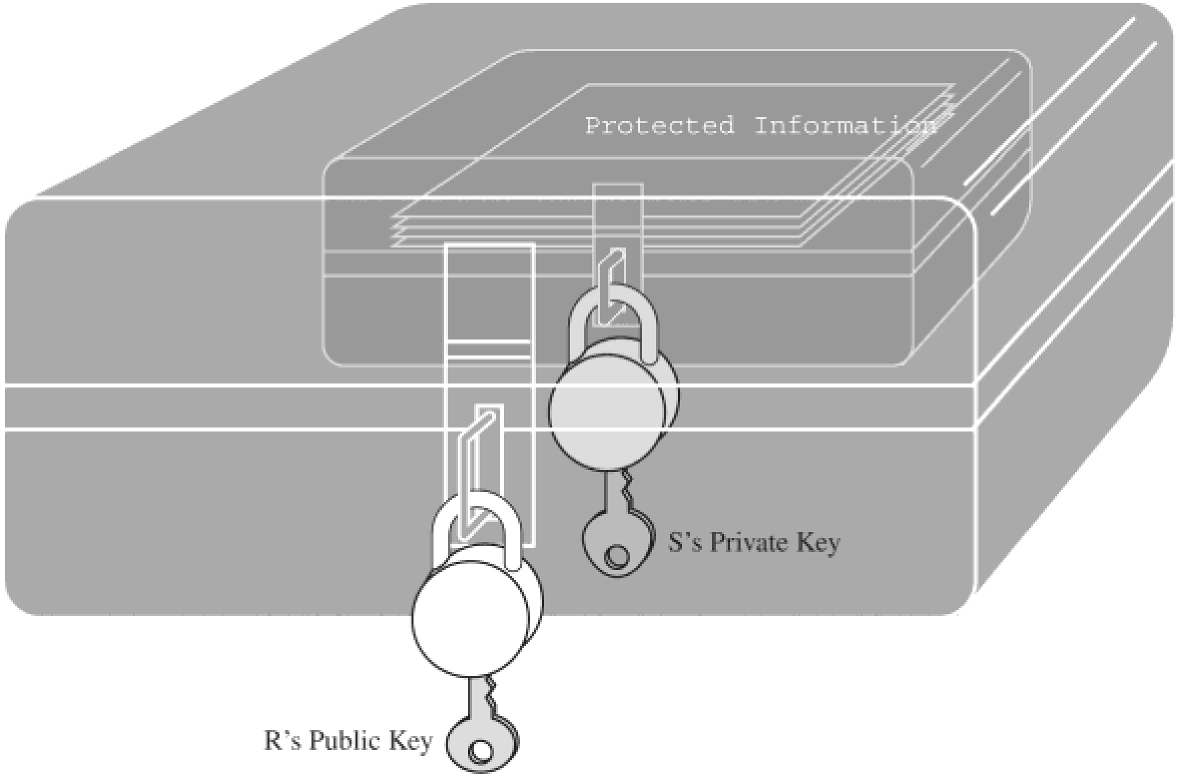
Source: Wikipedia.org

Key Exchange

- Typical scenario: before establishing an encrypted session, you need a secure means to exchange keys
- It addresses the problem of two previously unknown parties exchanging cryptographic keys
- Examples: online payment, shopping, secure email, etc.

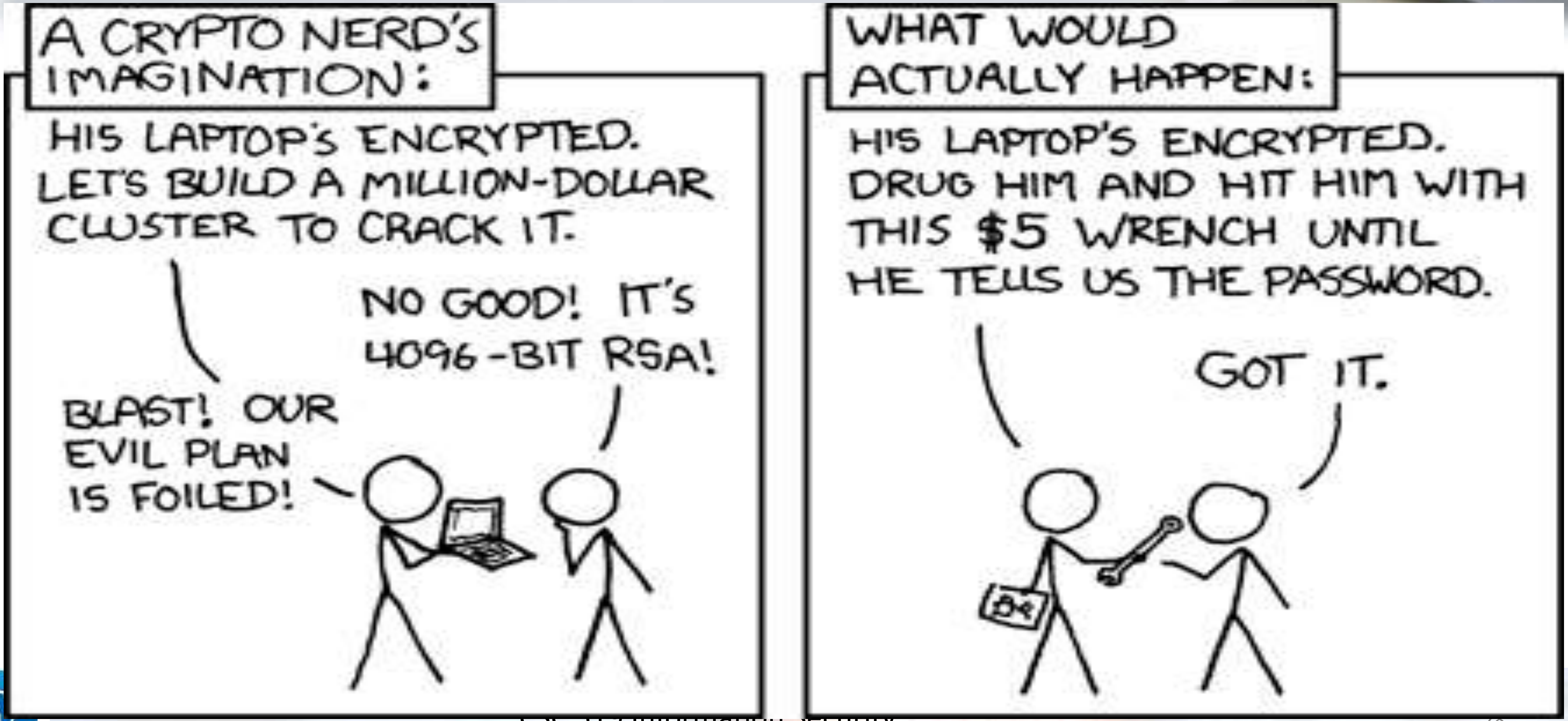
Key Exchange

- Uses both symmetric and asymmetric systems
- Suppose S and R want to derive a shared symmetric key, and have public/private keys
  - k_{PRIV-S}, k_{PUB-S}, k_{PRIV-R}, k_{PUB-R}
- S generates K and then sends to R:
  - $E(k_{PUB-R}, E(k_{PRIV-S}, K))$
  - Like two boxes, one assures the source of K came from S
  - And the other protects the whole message



Pfleeger/Pfleeger Fig. 02-11

However…


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

Cryptography and Security (Summary 1/4)

Characteristic	Description	Protection
Confidentiality	Ensures that only authorized parties can view the information	Encrypted information can only be viewed by those who have been provided the key
Integrity	Ensures that the information is correct and no unauthorized person or malicious software has altered that data	Encrypted information cannot be changed except by authorized users who have the key
Availability	Ensures that data is accessible to authorized users	Authorized users are provided the decryption key to access the information
Authenticity	Provides proof of the genuineness of the user	Cryptography can prove that the sender was legitimate and not an imposter
Non-repudiation	Proves that a user performed an action	Cryptographic non-repudiation prevents an individual from fraudulently denying they were involved in a transaction

Table 11-1 Information protections by cryptography

Cryptography and Security (Summary 2/4)

Symmetric Cryptographic Algorithms

Characteristic	Protection?
Confidentiality	Yes
Integrity	Yes
Availability	Yes
Authenticity	No
Non-repudiation	No

Table 11-3 Information protections by symmetric cryptography

Cryptography and Security (Summary 3/4)

Hashing Algorithms

Characteristic	Protection?
Confidentiality	No
Integrity	Yes
Availability	No
Authenticity	No
Non-repudiation	No

Table 11-2 Information protections by hashing cryptography

Cryptography and Security (Summary 4/4)

Asymmetric Cryptographic Algorithms

Characteristic	Protection?
Confidentiality	Yes
Integrity	Yes
Availability	Yes
Authenticity	Yes
Non-repudiation	Yes

Table 11-6 Information protections by asymmetric cryptography