Announcements

The room for TA/reader office hour is changed to **ICS2 214, 215, 216, 217**.

- Time: still *Thu 5-6 PM*

About course prerequisite

- CS161 + one or two of (CS143A, CS131, CS132) is the ideal background
- If not having the above, judge based on previous years' lecture slides and homeworks

http://sconce.ics.uci.edu/134-S19/

Cryptography: History, Simple Encryption Methods and Preliminaries

[lecture slides are adapted from previous slides by Prof. Gene Tsudik]

Cryptography

The word cryptography comes from the Greek words κρυπτός (hidden or secret) and γράφειν (writing).

Thus, historically cryptography has been:

The **art** of secret writing

Most of today's cryptography is well-grounded in mathematics and it's unclear whether there's still an "art" aspect to it.

Cryptography can be used at several different levels

- Algorithms: encryption, digital signatures, hashing, Random Number Generators (RNGs), secure erasure

- Protocols (2 or more parties): key distribution, authentication, identification, log-in, e-payment, etc.

- Systems: electronic cash, secure file-systems, smartcards, VPNs, e-voting, crypto-currencies, etc.

- Attacks: on all the above

Some Applications of Cryptography

- Network, operating system security
- Protect Internet, phone, space communication
- Electronic payments (e-commerce)
- Database security
- Software/content piracy protection
- Pay TV (e.g., satellite)
- Military communications
- Voting

Open vs. Closed Design Model

- **Open design**: algorithm, protocol, system design (and even possible plaintext) are public information. Only key(s) are kept secret.

- **Closed design**: as much information as possible is kept secret.

Core Issue in Network security : How to Communicate Securely?

Alice

Bob

Eve(sdropper)

Looks deceptively simple …
But, the devil is in the details

Note: even storage is a form of communication

The Biggest “Headache” is that…

Good security must be

Effective

Yet

Unobtrusive

Because security is not a service in and of itself, but a burden!

Cryptography is Old …

- Most sub-fields in CS are fairly new (20-30 years):
  - Graphics, compilers, software, OS, architecture

- And, some are quite old:
  - Predate computing and electronic comm.
  - Cryptography, database, networking

Some History: Caesar's Cipher

Homo

Hominem

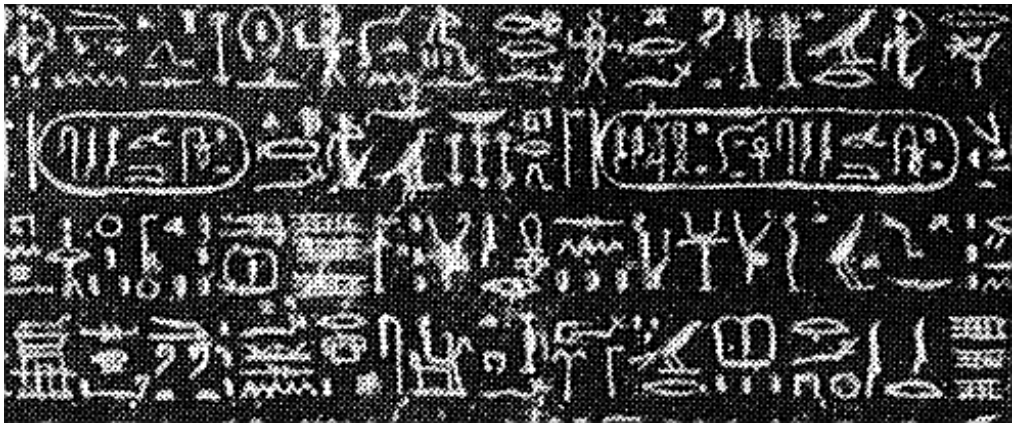
Lupus!

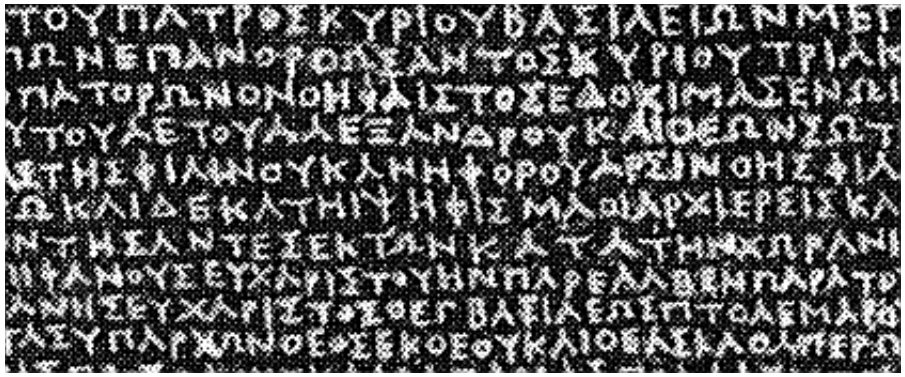
Krpr

Krplqhp

Oxsxv!

Some History: Rosetta Stone





Some History: Enigma

Alan Turing
(1912-1954)

Historical (Primitive) Ciphers

- **Shift (e.g., Caesar):** $Enc_k(x) = x+k \bmod 26$
- **Affine:** $Enc_{k1,k2}(x) = k1 * x + k2 \bmod 26$
- **Substitution:** $Enc_{perm}(x) = perm(x)$
- **Vigenere:** $Enc_K(x) = (X[0]+K[0], X[1]+K[1], \ldots, X[n]+K[N])$
- **Vernam:** One-Time Pad (OTP)

Shift (Caesar) Cipher

Example:

$K = 11$

W	*E*	*W*	*I*	*L*	*L*	*M*	*E*	*E*	*T*	*A*	*T*	*M*	*I*	*D*	*N*	*I*	*G*	*H*	*T*
22	4	22	8	11	11	12	4	4	19	0	19	12	8	3	13	8	6	7	19
7	15	7	19	22	22	23	15	15	4	11	4	23	19	14	24	19	17	18	4
H	*P*	*H*	*T*	*W*	*W*	*X*	*P*	*P*	*E*	*L*	*E*	*X*	*T*	*O*	*Y*	*T*	*R*	*S*	*E*

- How many possible keys are there?
- How many trials are needed to find the key?

Substitution Cipher

Example:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

X N Y A H P O G Z Q W B T S F L R C V M U E K J D I

KEY

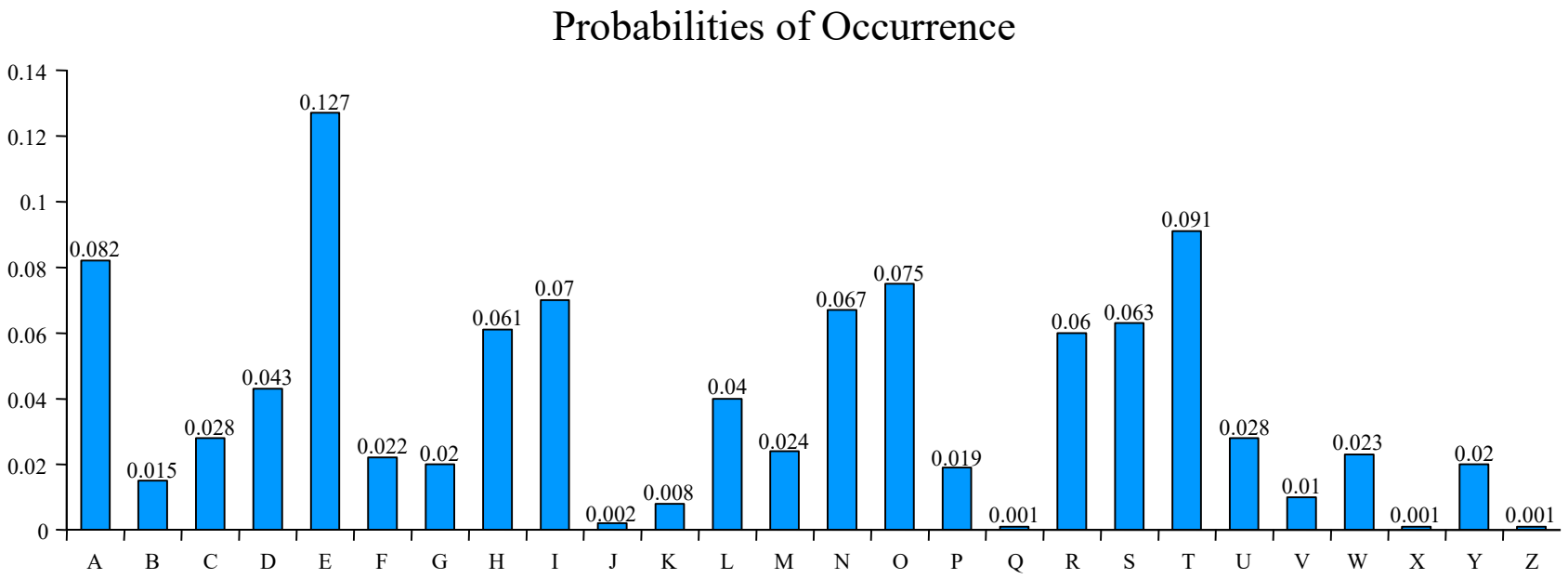
W E W I L L M E E T A T M I D N I G H T

K H K Z B B T H H M X M T Z A S Z O G M

- How many possible keys are there?
- How many trials are needed to find the key?

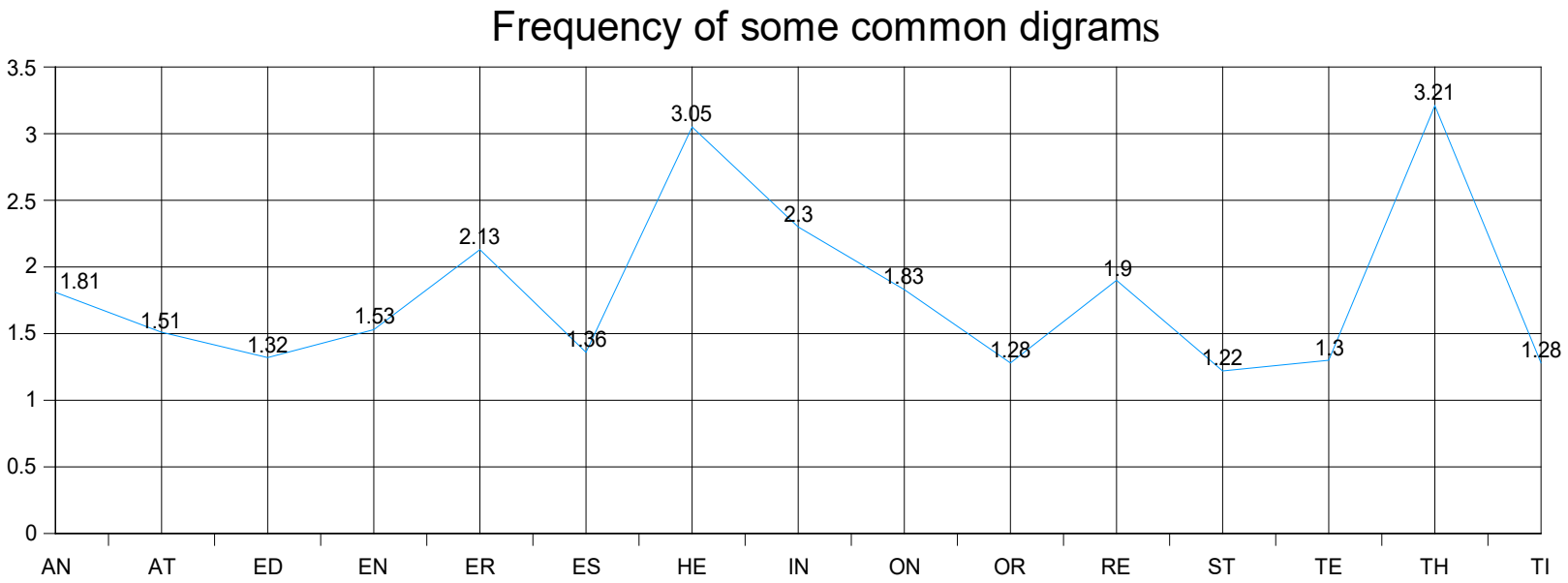
Substitution Cipher

Cryptanalysis



Substitution Cipher

Cryptanalysis



VERNAM One-Time Pad (OTP): World's Best Cipher

$$\text{Plaintext} = \{p_0, \ldots, p_{n-1}\}$$

$$\text{One - time pad stream} = \{otp_0, \ldots, otp_{n-1}\}$$

$$\text{Ciphertext} = \{c_0, \ldots, c_{n-1}\}$$

$$where:$$

$$c_i = p_i \oplus otp_i \forall 0 < i < n$$

$$C = A \oplus B$$

$$C \oplus B = A$$

VERNAM One-Time Pad (OTP): World's Best Cipher

- Vernam offers perfect information-theoretic security,
  - For any m0, m1, Pr[E(k, m0) = c] = Pr[E(k, m1) = c]
  - Intuitively, ciphertext reveal no "info" about plaintext

but:

- How long does the OTP keystream need to be?
- How do Alice and Bob exchange the keystream?

Encryption Principles

- A cryptosystem has (at least) five ingredients:
  - Plaintext
  - Secret Key
  - Ciphertext
  - Encryption Algorithm
  - Decryption Algorithm

- Security usually depends on the secrecy of the key, not the secrecy of the algorithms

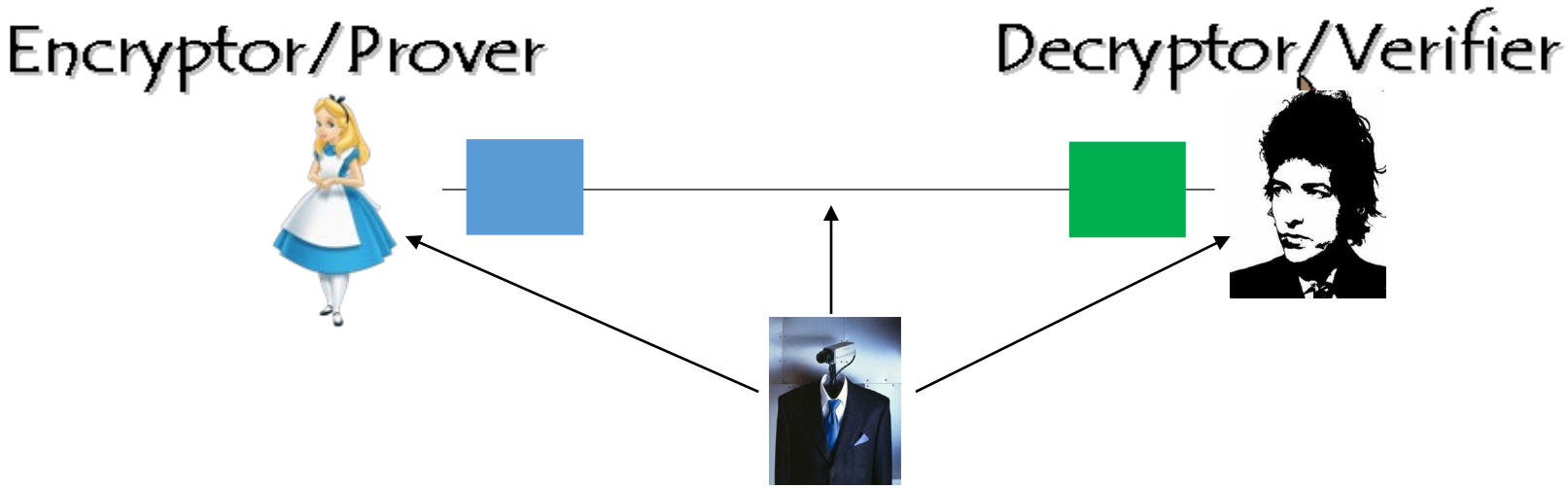
Crypto Basics

Crypto Attacks:

- ciphertext only
- known plaintext
- chosen plaintext
- chosen ciphertext
- brute force

Cryptosystem:

- *P -- plaintext*
- *C -- ciphertext*
- *K -- keyspace*
- *E -- encryption rules*
- *D -- decryption rules*



Average Time for Exhaustive Key Search (for Brute-Force Attacks)

Key Size (bits)	**Number of Alternative Keys**	**Time required at 10^6 Decr/µs**
32	$2^{32} = 4.3 \times 10^9$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	10 hours
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	5.9×10^{30} years

Today, > 80 bits is considered “secure”

Types of Attainable Security

• Perfect, unconditional or "information theoretic": the security is evident free of any (computational/hardness) assumptions

• Reducible or "provable": security can be shown to be based on some common (often unproven) assumptions, e.g., the conjectured difficulty of factoring large integers

• Ad hoc: the security seems good → "snake oil"…

Take a look at:

http://www.ciphersbyritter.com/GLOSSARY.HTM

Computational Security

- Encryption scheme is *computationally secure* if
  - cost of breaking it (via brute force) exceeds the value of the encrypted information; or
  - time required to break it exceeds useful lifetime of the encrypted information

- Most modern schemes we will see are considered computationally secure
  - Usually rely on having a very large key-space, impregnable to brute force attacks

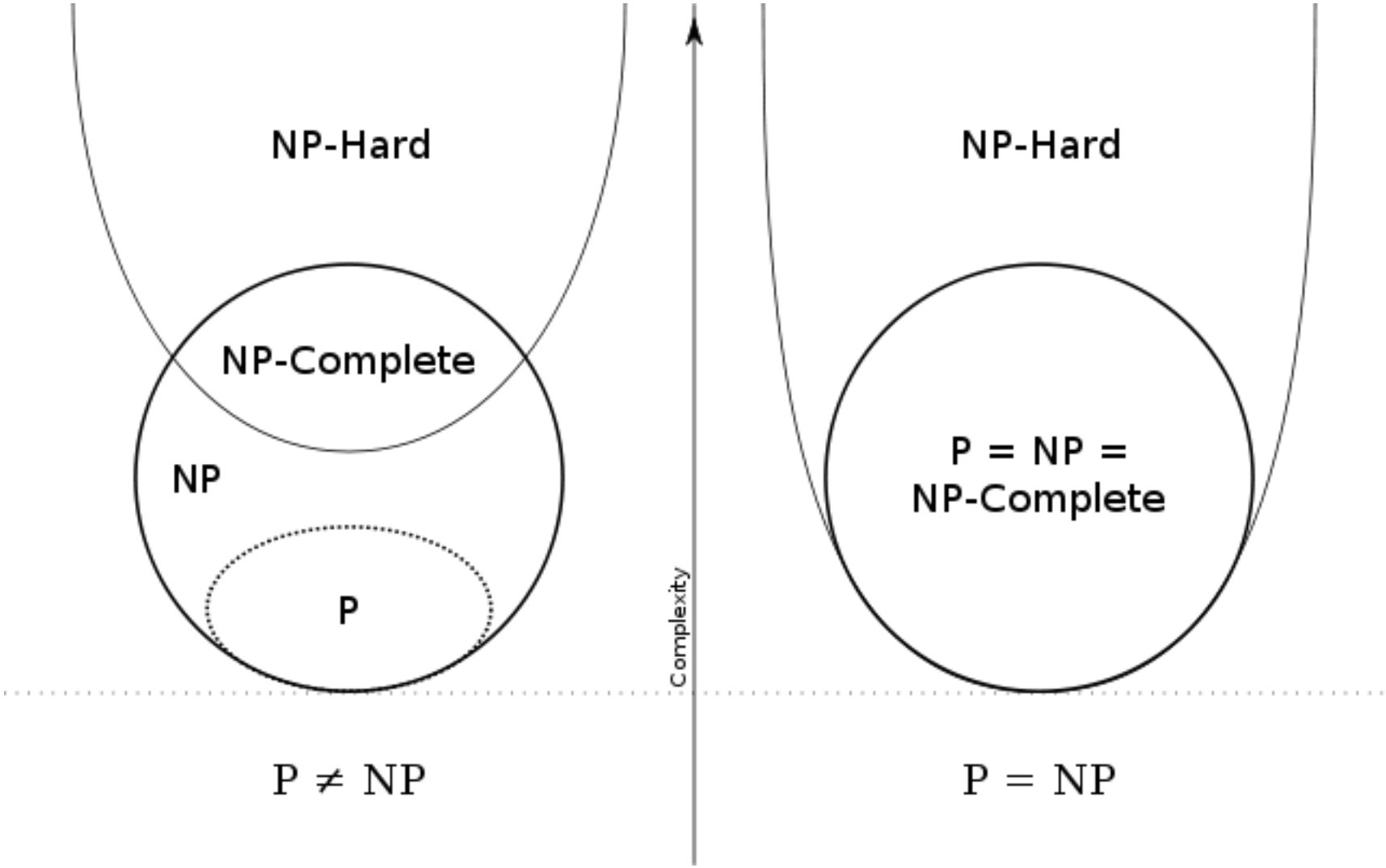
- Most advanced schemes rely on lack of knowledge of effective algorithms for certain hard problems, not on a proven inexistence of such algorithms (reducible security)!
  - Such as: factorization, discrete logarithms, etc.

Complexity Reminder/Re-cap

- **P:** problems that can be solved in polynomial time, i.e., problems that can be solved/decided "efficiently"
- **NP:** broad set of problems that includes P;
  - answers can be verified "efficiently" (in polynomial time);
  - solutions cannot always be efficiently found (as far as we know).
- **NP-complete:** believed-to-be-hard decision problems in NP; they appear to have no efficient solution; answers are efficiently verifiable, solution to one is never much harder than a solution to another
- **NP-hard:** hardest; some of them may not be solved by a non-deterministic TM. Many computational version of NP-complete problems are NP-hard.
- **Examples:**
  - Factoring, discrete log are in NP, not known if NP-complete or P
  - Primality testing was "recently" (2002) shown to be in P
  - Knapsack is NP-complete

For more info, see: **https://www.nist.gov/dads//**

P vs NP

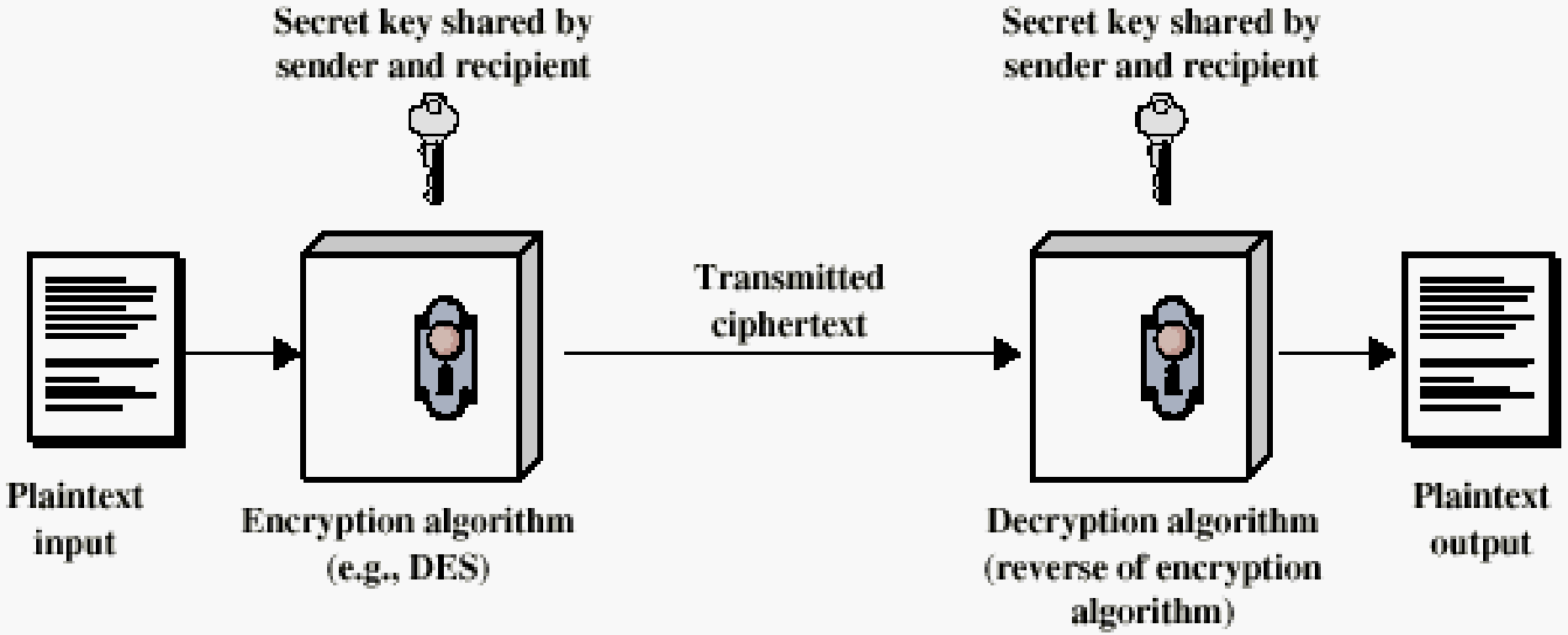

NP-Hard
NP-Complete
NP
P
Complexity
NP-Hard
P = NP =
NP-Complete
P ≠ NP
P = NP

Cryptosystems

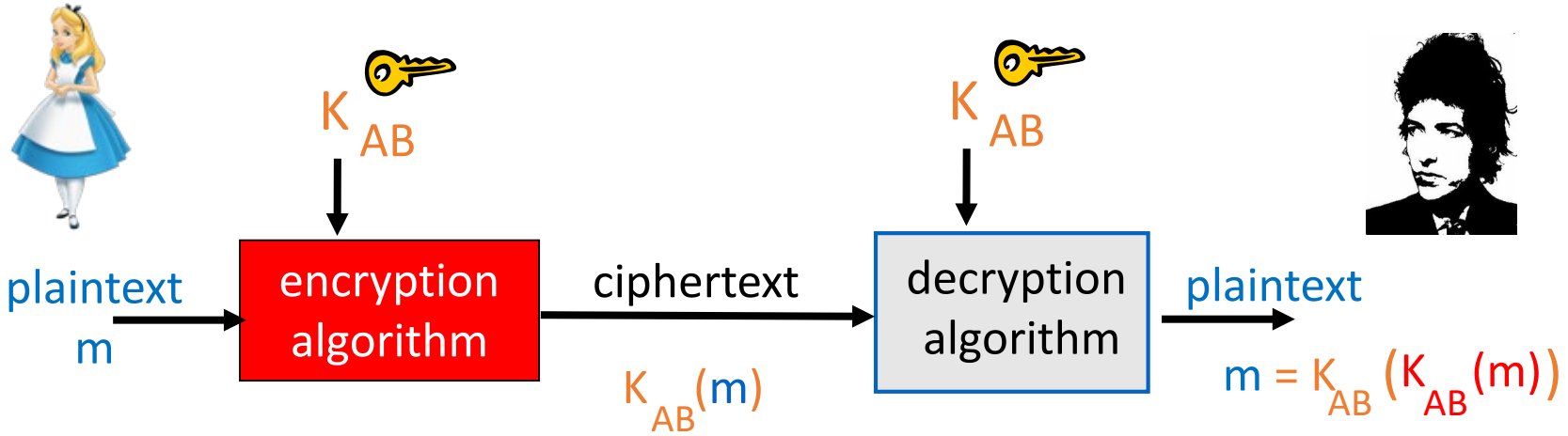
Classified along three dimensions:

- Type of operations used for transforming plaintext into ciphertext
  - Binary arithmetic: shifts, XORs, ANDs, etc.
    - Typical for **conventional/symmetric** encryption
  - Integer arithmetic
    - Typical for **public key/asymmetric** encryption
- Number of keys used
  - Symmetric or conventional (single key used)
  - Asymmetric or public-key (2 keys: 1 to encrypt, 1 to decrypt)
- How plaintext is processed:
  - One bit at a time – “stream cipher”
  - A block of bits – “block cipher”

Conventional/Symmetric Encryption Principles



Conventional (Symmetric) Cryptography

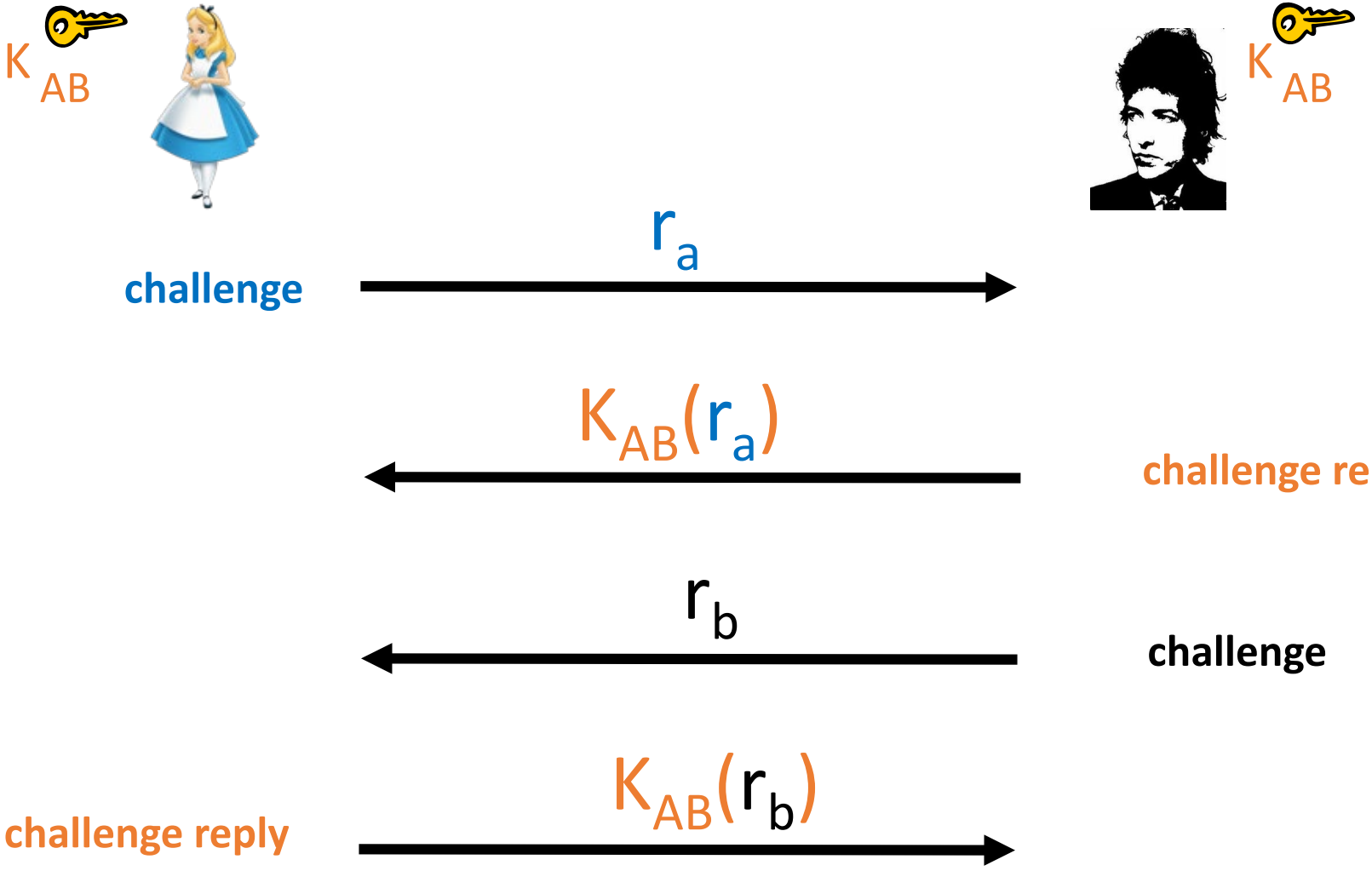


- Alice and Bob share a key K_{AB} which they somehow agree upon (how?)
  - key distribution / key management problem
  - ciphertext is roughly as long as plaintext
  - examples: Substitution, Vernam OTP, DES, AES

Uses of Conventional/Symmetric Cryptography

- Message transmission (confidentiality):
  - Communication over insecure channels
- Secure storage: crypt on Unix
- Strong authentication: proving knowledge of a secret without revealing it:

Challenge-Response Authentication Example



Uses of Conventional/Symmetric Cryptography

- Message transmission (confidentiality):
  - Communication over insecure channels
- Secure storage: crypt on Unix
- Strong authentication: proving knowledge of a secret without revealing it:
  - Eve can obtain chosen <plaintext, ciphertext> pair
  - Challenge should be chosen from a large pool
- Integrity checking: fixed-length checksum for message via secret key cryptography
  - Send MAC along with the message MAC=H(K, m)

Conventional/Symmetric Cryptography

- Advantages
  - high data throughput
  - relatively short key size
  - primitives to construct various cryptographic mechanisms
- Disadvantages
  - key must remain secret at both ends
  - key must be distributed securely and efficiently
  - relatively short key lifetime

Public Key (Asymmetric) Cryptography

- Asymmetric cryptography
- Invented in 1974-1978 (Diffie-Hellman, Rivest-Shamir-Adleman)
  - Both win Turing awards (2002, 2015)!
- Two keys: private (SK), public (PK)
  - Encryption: with public key;
  - Decryption: with private key
  - Digital Signatures: Signing by private key; Verification by public key. i.e., "encrypt" message digest/hash -- $h(m)$ -- with private key
    - Authorship (authentication)
    - Integrity: Similar to MAC
    - Non-repudiation: can't do with secret/symmetric key cryptography
- Much **slower** (~1000x) than conventional cryptography
  - Often used together with conventional cryptography, e.g., to encrypt session keys

Genesis of Public Key Cryptography: Diffie- Hellman Paper

https://www-ee.stanford.edu/~hellman/publications/24.pdf

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

***Abstract*—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.**

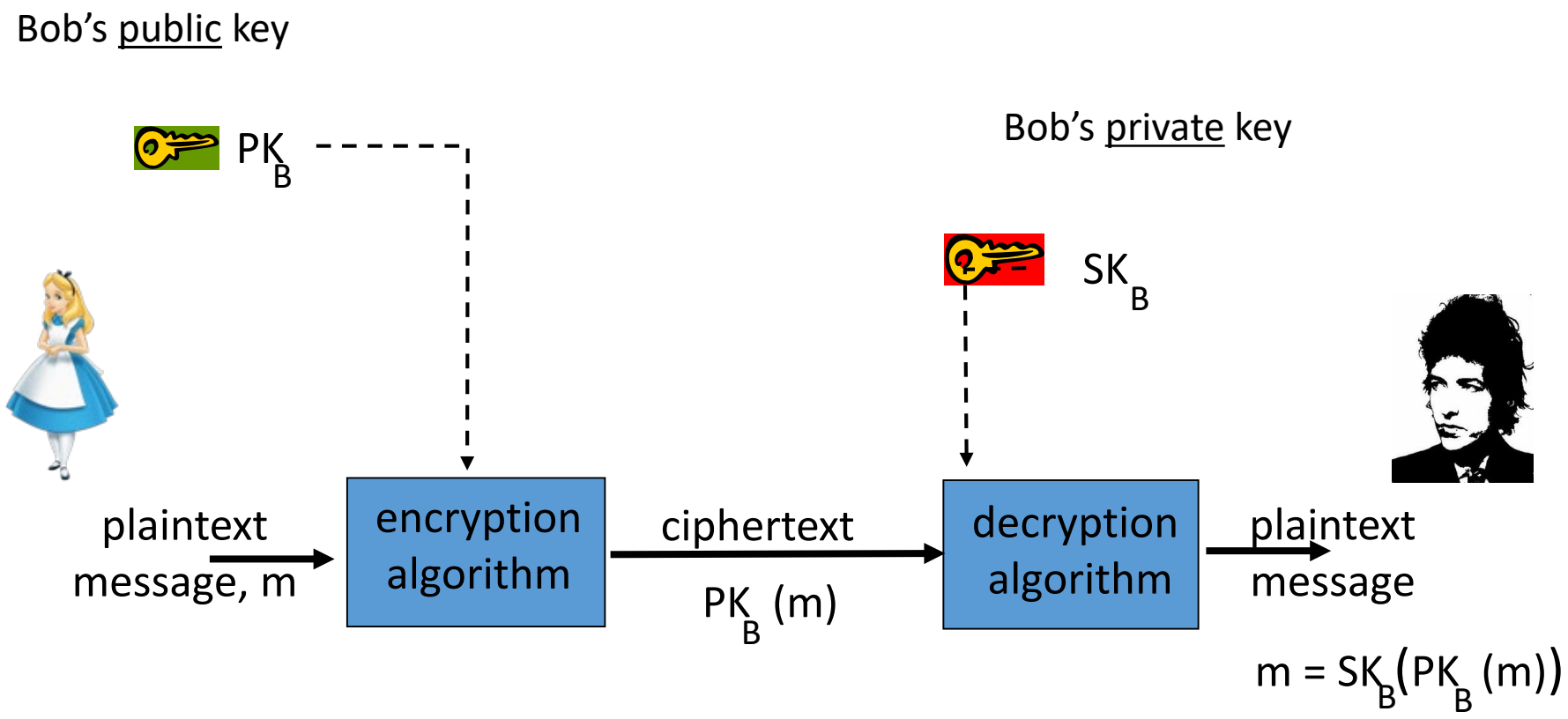
I. Introduction

WE STAND TODAY on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals. In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Section III proposes two approaches to transmitting keying information over public (i.e., insecure) channels without compromising the security of the system. In a *public key cryptosystem* enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infeasible (e.g., requiring 10^{100} instructions). The enciphering key E can thus be publicly disclosed without compromising the deciphering key D. Each user of the network can, therefore, place his enciphering key in a public directory. This enables any user of the system to send a message to any other user enci-

Public Key Cryptography



Uses of Public Key Cryptography

- Data transmission (confidentiality):
  - Alice encrypts m_a using PK_B, Bob decrypts it to obtain m_a using SK_b.
- Secure Storage: encrypt with own public key, later decrypt with own private key
- Authentication:
  - No need to store secret**s**, only need *public* keys.
  - Secret/symmetric key cryptography: need to share *secret* key for every person one communicates with
- Digital Signatures (authentication, integrity, non-repudiation)

Public Key Cryptography

- Advantages
  - only the private key must be kept secret
  - relatively long life time of the key
  - more security services
  - relatively efficient digital signatures mechanisms
- Disadvantages
  - low data throughput
  - much larger key sizes
  - distribution/revocation of public keys
  - security based on conjectured hardness of certain computational problems

Comparison Summary

- Public key
  - encryption, signatures (esp., non-repudiation), and key management
- Conventional/symmetric
  - encryption and some data integrity applications
- Key sizes
  - Keys in public key crypto must be larger (e.g., 2048 bits for RSA) than those in conventional crypto (e.g., 112 bits for 3-DES or 256 bits for AES)
    - most attacks on "good" conventional cryptosystems are exhaustive key search (brute force)
    - public key cryptosystems are subject to "short-cut" attacks (e.g., factoring large numbers in RSA)

Suggested Readings:

Chapters 1 and 2 in KPS book

Optional: Ch 1 in Stinson