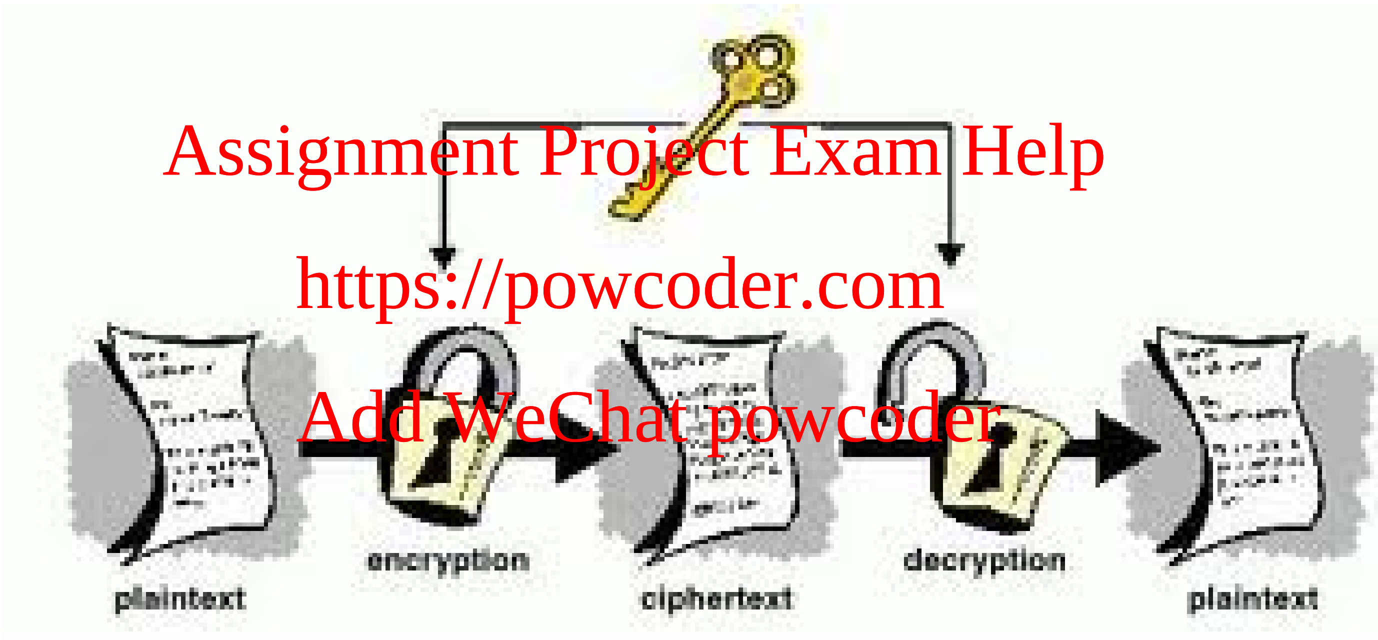
EECS 3482
Introduction to Computer Security



Instructor: N. Vlajic, Winter 2020

Learning Objectives

Upon completion of this material, you should be able to:

- Explain the difference between classical and modern day cryptography.

- List & describe several representative examples of classical encryption.

- Describe the evolution of <u>symmetric</u> cryptography – from DES to 3DES and AES, and their current day uses.

- Explain the basics of <u>asymmetric</u> cryptography, and current day uses of Diffie Hellman and RSA encryption algorithms.
- Discuss the use of public-key cryptography for purposes of message integrity, authentication & digital signatures.

Required Reading

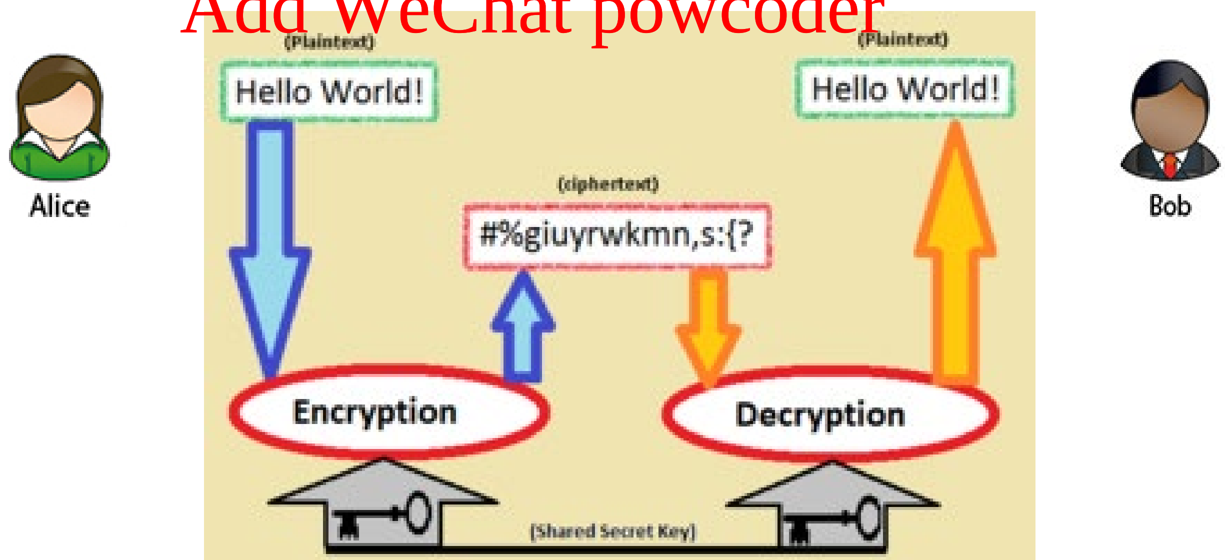
Computer Security, Stallings: Chapter 2

Sections 20.2, 20.3

Sections 21.4, 21.5

Introduction

- **Cryptography** – process/technique(s) of converting data into unintelligible form in order to ensure: confidentiality, data integrity, and authentication
  - requirement 1: no data should be lost during encryption
  - requirement 2: decryption should ensure perfect data recovery

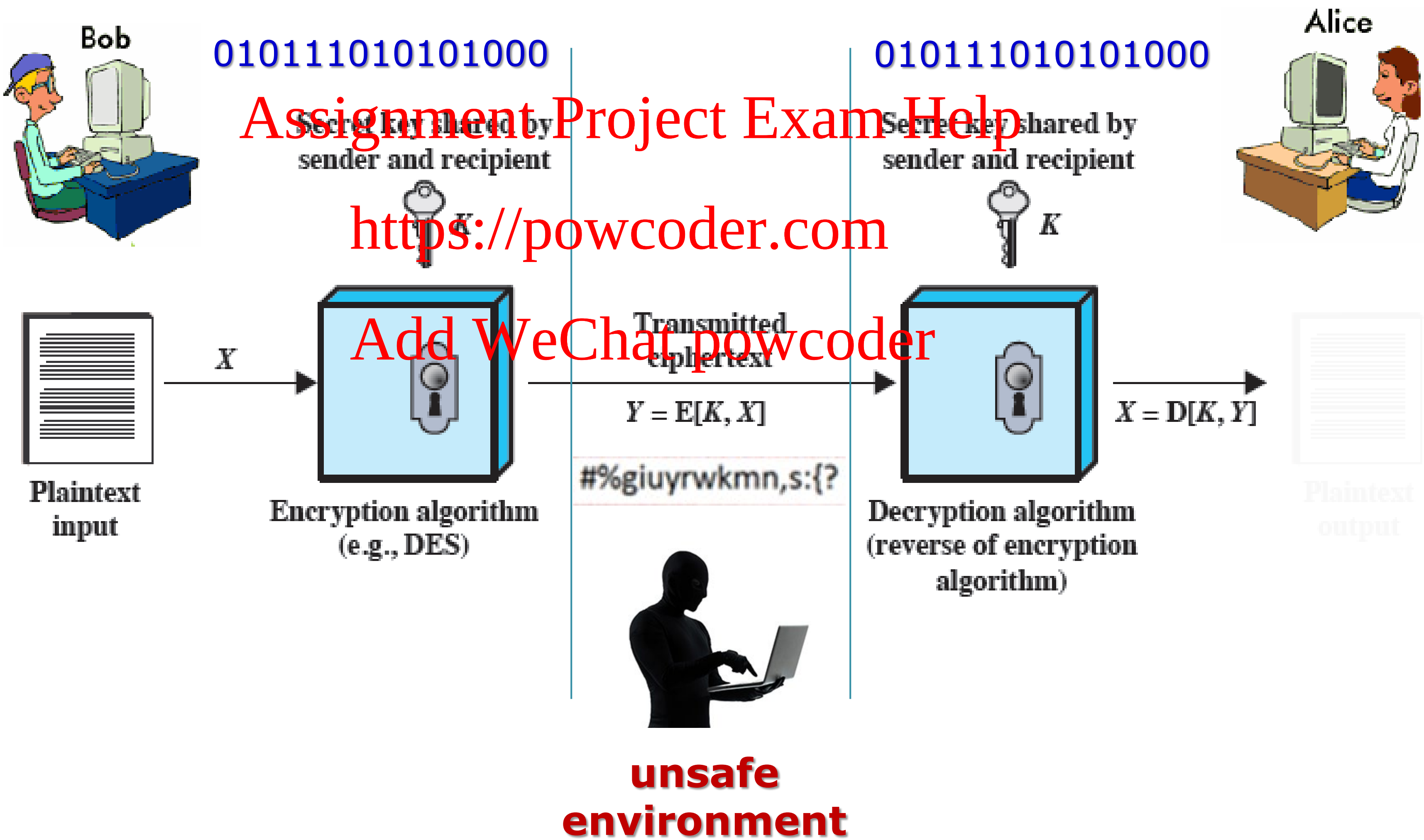


Introduction (cont.)

- **Elements of Encryption System**
  - **plaintext** – original message that should be ‘protected’
  - **encryption algorithm** – performs various substitutions and transformations on plaintext
  - **key** – variable data that is input into encryption algorithm together with plaintext
    - determines exact substitutions and transformations performed on plaintext
  - **ciphertext** – scrambled message produced as output
  - **decryption algorithm** – encryption algorithm run in reverse

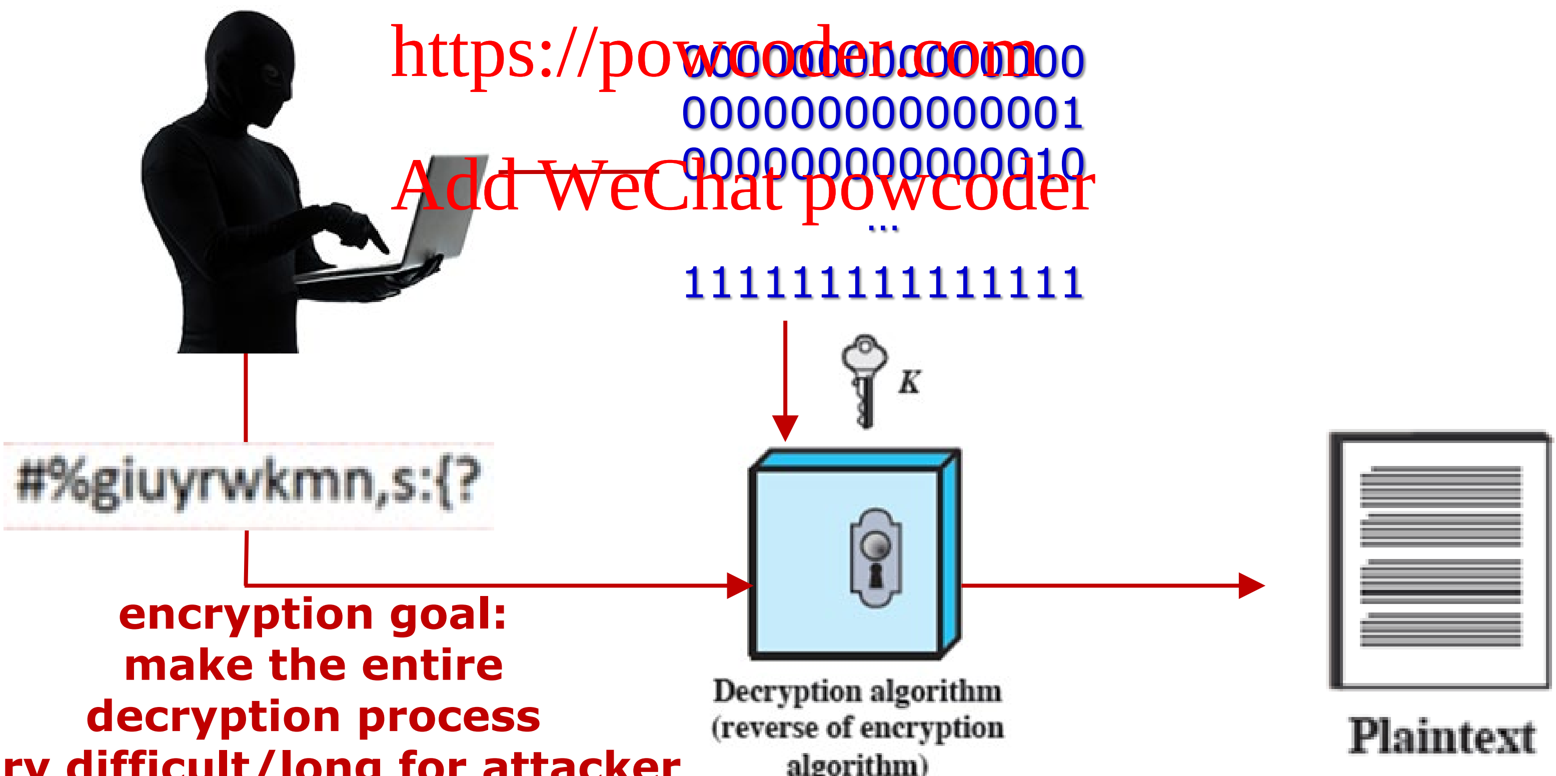
Introduction (cont.)

- **Elements of Encryption System (cont.)**



Introduction (cont.)

- **Process of Breaking a Cipher**
  - in modern cryptography encryption/decryption algorithm is not a secret
  - hacker probes various keys on the captured ciphertext



**encryption goal:
make the entire
decryption process
very difficult/long for attacker**

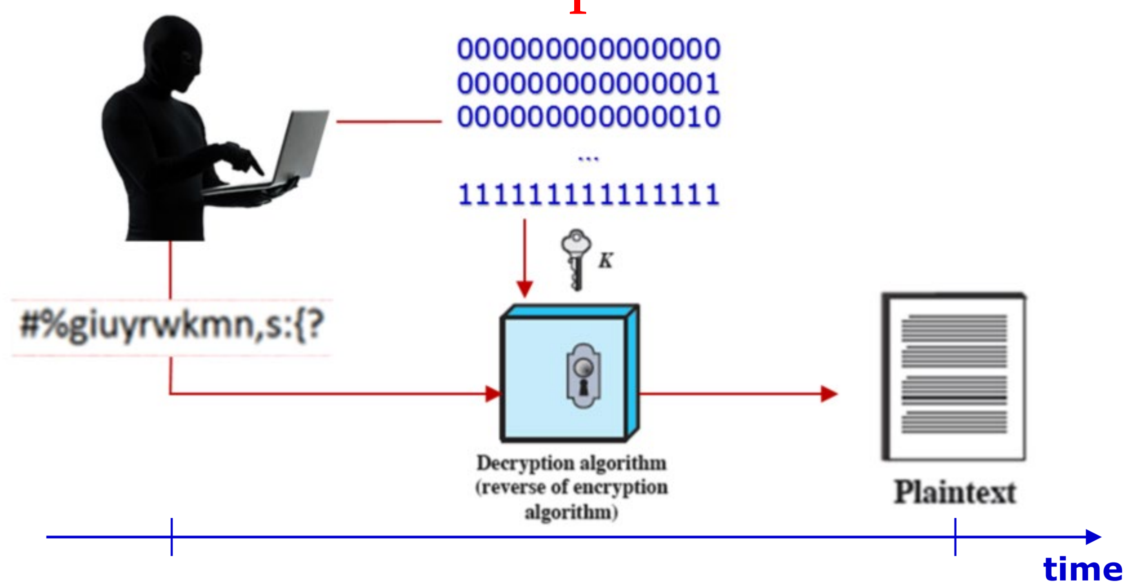
Introduction (cont.)

- **Factors that Influence Success of Crypto-Attack**
  - time to perform one decryption – $t_{one\text{-}decryption}$
  - number of keys to <u>**try**</u> – n_{keys}

$$\text{crypto-attack speed} = \boxed{n_{keys}} \times \boxed{t_{one\text{-}decryption}}$$

number of tried keys

depends on processor speed of attacker's machine



Introduction (cont.)

$$\textbf{crypto-attack speed} = \boxed{n_{keys}} \times t_{one\text{-}decryption}$$

BEST case for hacker:
$n_{keys} = 1$

WORST case for hacker:
$n_{keys} = 2^N$

$$\overbrace{\begin{array}{c}000000000000000\\000000000000001\\000000000000010\\\ldots\\111111111111111\end{array}}^{\text{N bits long keys}}$$

Introduction (cont.)

- **Factors that Influence Success of Crypto-Attack (cont.)**
  - **brute force attack on ciphertext** – all possible keys are tried until an intelligible translation into plaintext is obtained
  - **with current processing capabilities, 56 bit keys are not considered safe**

N bits $\quad n_{keys} = 2^N \quad t_{one\text{-}decrypt} = 1\ [10^{-6}\ sec] \quad t_{one\text{-}decrypt} = 1\ [10^{-12}\ sec]$

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/µs	Time Required at 10^6 Decryptions/µs
32	$2^{32} = 4.3 \times 10^9$	2^{31} µs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} µs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} µs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} µs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} µs = 6.4×10^{12} years	6.4×10^6 years

Introduction (cont.)

- **Estimation of Processor Speeds Today …**
  - **Moore's Law** – computing power doubles every 18 months (1.5 years)
  - in 1997 it was possible to crack 1 million keys / second

$$\text{cracking power} = 1 \text{ million } \frac{\text{keys}}{\text{sec}} \times 2^{\frac{\text{now}-1997}{1.5}}$$

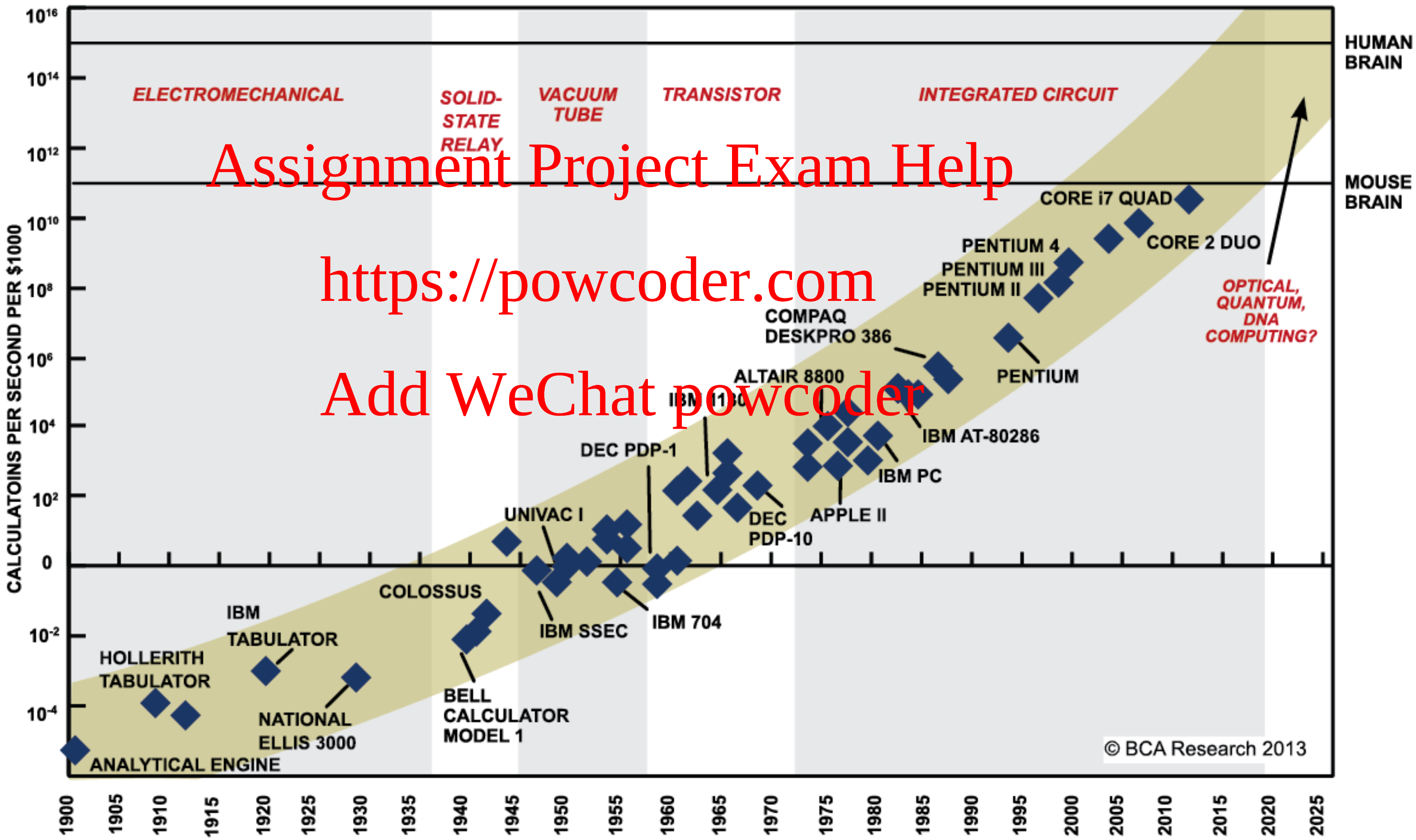
$$\text{cracking power in } 2020 =$$

$$= 1 \text{ million } \frac{\text{keys}}{\text{sec}} \times 2^{\frac{2020-1997}{1.5}} =$$

$$= 41{,}285 \text{ billion } \frac{\text{keys}}{\text{sec}} = \mathbf{41 \times 10^3 \times 10^9} \text{ keys/sec} = \mathbf{41 \times 10^{12}} \text{ keys/sec}$$

Introduction (cont.)

Moor's Law



SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, *THE VIKING PRESS*, 2006. DATAPOINTS BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.

Introduction (cont.)

numbers historically increased

numbers historically decreased

$$\text{crypto-attack speed} = n_{keys} \times t_{one\text{-}decryption}$$

depends on:

- **processor speed of attacker's machine** – decreases due to Moor's l.
- **complexity of crypto algorithm**

defender's goal is to make this speed/time as long as possible.

This is where the emphasis of modern cryptography is.

How about we use/invent very complex algorithms.
Good or bad idea ??

Introduction (cont.)

- **Cryptography vs. Cryptanalysis**

	Cryptography	Cryptanalysis
Defintion	The art or science of encrypting plain messages into cipher text for security of the messages especially while transmission	The art of obtaining plain text from a cipher text without knowledge of key.
Origin	From Greek κρυπτός, "hidden, secret"; and γράφειν, graphein, "writing", or -λογία, -logia, "study", respectively	From Greek kryptós, "hidden", and analýein, "to loosen" or "to untie"
Practitioner	Cryptographer	Cryptanalyst
Focus	Secret writing	Breaking secrets

http://www.differencebetween.info/difference-between-cryptography-and-cryptanalysis

Introduction (cont.)

The Cryptographer's Dilemma

As with many analysis techniques, having very little ciphertext inhibits the effectiveness of a technique being used to break an encryption. A cryptanalyst works by finding patterns. Short messages give the cryptanalyst little to work with, so short messages are fairly secure with even simple encryption.

Substitutions highlight the cryptologist's dilemma: An encryption algorithm must be regular for it to be algorithmic and for cryptographers to be able to remember it. Unfortunately, the regularity gives clues to the cryptanalyst.

There is no solution to this dilemma. In fact, cryptography and cryptanalysis at times seem together like a dog chasing its tail. First, the cryptographer invents a new encryption algorithm to protect a message. Then, the cryptanalyst studies the algorithm, finding its patterns and weaknesses. The cryptographer then sets out to try to secure messages by inventing a new algorithm, and then the cryptanalyst has a go at it. It is here that the principle of timeliness from Chapter 1 applies; a security measure must be strong enough to keep out the attacker only for the life of the data. Data with a short time value can be protected with simple measures.

Security in Computing
By Charles P. Pfleeger, Shari Lawrence Pfleeger

Introduction (cont.)

Every Cryptographer Has to Be a Good Cryptanalyst

Every cryptographer's aim is naturally to design an algorithm that won't supply any practically usable results when cryptanalyzed. This doesn't necessarily mean that it can't be cryptanalyzed at all. It normally means that it would take too long (the encrypted information might become worthless in the meantime), or that it would be too costly to justify the value of the information.

For instance, the encryption methods used at the fronts in World War I had been estimated by the cryptologists to require at least one day's work for the adversary to recover the plaintext. After one day, the encrypted commands had become worthless—the shells had long hit by that time. The catch in the matter could only have been that the adversary deciphered faster than expected [BauerMM].

Cryptology Unlocked
By Reinhard Wobst

Introduction (cont.)

Example: Is the best encryption always necessary?

#%giuyrwkmn,s:{?

data($time_1$) valid for only Δt seconds

Encryption that keep intruder 'busy' for > Δt seconds may be good enough!