Computer & Information Security (372-1-460-1)

Cryptographic Tools

Dept. of Software and Information Systems Engineering, Ben-Gurion University

Prof. Yuval Elovici

Spring, 2024

Cryptography

- Is
  - A powerful tool for protecting information
  - The basis for many security mechanisms
  - Mostly based on Math, logics and Statistics
- Is not
  - The solution to all security problems
  - Reliable unless implemented properly
  - Reliable unless used properly

Use cases

תרחישים שמשתמשים בתורת ההצפנה:

- UC1: Secure communication – לדוגמא, כניסה לאתר הבנק, רואים מנעול בדפדפן והוא מסמל שנפתח ערוץ מוצפן.
- UC2: Protect files via encryption – בקבצים, עושים הבחנה בין מידע בתנועה (גלישה באתר) או במנוחה (מידע שעל המחשב). יש מוטיבציה להצפין את המידע בתנועה.
- UC3: Secure cloud computing – שימוש בענן, יש סיכון גדול שמפעיל הענן יוכל לגשת לענן שלנו - צריך להשתמש בטכניקות להבטחה (טכניקות חישוביות).
- UC4: Blockchain (bitcoin) - כל זה מבוסס על עולם cryptography.

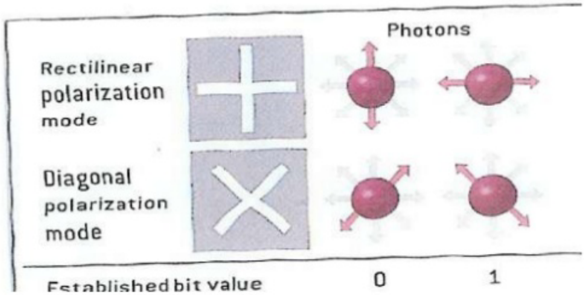
הכל התחיל מאלגוריתמים ידניים ומכניים.

Cryptography techniques - history

- כל הקריפטוגרפיה ממומשת בתוך המחשב.

- Manual
- Mechanical
- Electro-mechanical
- Electronic
- Quantum cryptography
  (base on physical rules, uncertainty theorem etc.)

Photon Polarization



יש תוקפים אוהבים שאנשים משתמשים במזעור הלבנה.

Cryptography - goals

- Confidentiality
- Integrity
- Authenticity
- Non-repudiation
- Access control
- Make attacks more difficult

להתכתש ולהגיד שלא אני התאמתי על זה.

- נראה שלכל אחד יהיו הרשאות שונות וכל אחד יכול לעשות דברים ספציפיים.
לדוגמא אנחנו תלמידים, מורה יכול לפרסם/לשנות ציון וסטודנט לא.
לדרישה זו אין בהכרח פתרון קריפטוגרפי.

Definitions

- Cryptology
  - Cryptography (Kryptos = hidden, Graphia = writing)
  - Cryptanalysis
- Input: Plaintext
- Cryptosystem
  - Algorithm
  - Key/Key space
- Output: Ciphertext

Auguste Kerckhoffs (1835)

שני עקרונות מאוד חשובים בעולם ההצפנה.

"The system should be, if not theoretically unbreakable, **unbreakable in practice**"

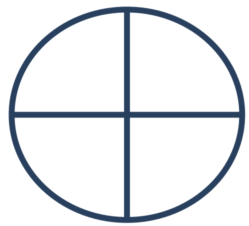
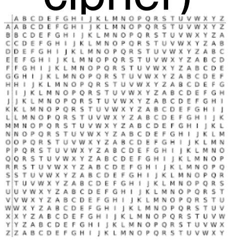
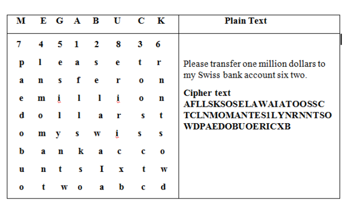
1 - המע' צריכה להיות אונבריקבל מבחינה פרקטית, אם יש לי סוד ואני רוצה להסתיר אותו אז אם אני יכול להסתיר את זה ברמה שכל המחשבים בעולם ינסו לפרוץ ויצליחו לפענח רק אחרי 8 ימים לדוגמא אז ההצפנה שלי טובה.

- Kerckhoffs principle: a cryptosystem should be secure even if **everything about the system**, except the secre key, **is public knowledge**.

2 - עקרון יותר חשוב! אסור לאדם שמנסה להגן על עצמו להניח שהאוי' יודע האלגוריתם מבחינת התוקף. ניתן לי יתרון. אני צריך להרגיש בטוח גם אם אפרסם את אלגוריתם ההצפנה שלי. הכוח של האלגוריתם נובע מאי ידיעת המפתח.

כלומר גם מודיעות האלגוריתם הם לא ידליפו לפרוש כי אין להם את המפתח.
הכוח האמיתי הוא במפתח ופחות באלגוריתם. המפתח מבקר את תהליך ההצפנה.

Cryptosystems' Main Categories

Substitution	Transposition	One time pad	Symmetric (Single Key)	Asymmetric (Two Keys) (PKE)
Mono alphabetic (Ceaser Cipher) 	Scytale Rod 	Vernum Cipher 	Block Cipher (DES, Triple DES, AES, RC5, RC6)	RSA, Diffie – Helman DSS ECC
Poly alphabetic (Vigenere cipher) 	Basic Transposition 		Stream Cipher (RC4)	

אנגלית: substitution כי מחליפים אות באות.

Basic cryptosystems - Substitution

מוסיפים מיקום של כל האותיות באנגלית,
לדוגמא נגיד את המיקום ב-3 אז A הופך ל-D.
המפתח הוא כי- offset כלומר מספר אלפביתי.

- Caesar cipher
- Mono-alphabetic cipher
- key = the offset
- Simple to break

טריוויאלי לפרוץ את זה כי אפשר לנסות
את כל 26 המפתחות.

Basic cryptosystems - Substitution

- Vigenere cipher (1586) – polyalphabetic cipher
- Key = boy (X axis)
- Plaintext = hello (Y axis)
- Cipher = isjmc
- More complicated to break
  - The frequency of the letters appears in the cipher is different

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

האלגוריתם הכי טריוויאלי להצפנה - נחזיק טבלה גדולה של אותיות A-Z-מ ובעמודה השניה נכתוב אלגוריתם
ששם את האותיות בצורה רנדומלית. הטבלה הזו היא מפתח ההצפנה שלי. כה נקרא:

Monoalphabetic Substitution

* לא יכול להיות שיותר מאות אחת תקבל את אותה אות. לדוגמא לא יכול להיות מצב שגם A וגם B
יקבלו את אותה אות הצפנה.

- So 26! ~= 4 x 10^{26} possible mappings
- Testing 10^{6}/sec would take 10^{13} years
- Is it secure?
- All natural languages have statistical properties (most common letters, digrams, trigrams, word endings, doubled letters, words)

האלגוריתם הזה משמר התפלגות.

Letter Frequency

Letter	%	Letter	%
E	13.0	A	7.3
T	9.3	S	6.3
N	7.8	D	4.4
R	7.7	H	3.5
I	7.4	L	3.5
O	7.4	C	3.0

לכל אות יש
הסתברות מופע
שונה מטקסט וייחודי.

זה נכון לגבי השפה האנגלית
באופן כללי לכן אפשר לעשות
סטטיסטיקה ולראות לדוגמא
שהאות y מופיעה 13%
פעמים ואז נבין שזה e
ולא y.

Frequency of Usage

Digrams	Trigrams	Doubles	Endings	Words
•th	•the	•ll	•e	•the
•he	•and	•tt	•t	•of
•at	•tha	•ss	•s	•are
•st	•hat	•ee	•d	•I
•an	•ent	•pp	•n	•and
•in	•ion	•oo	•r	•you
•ea	•for	•rr	•y	•a
•nd	•tio	•ff		•can
•er	•has	•cc		•to
•en	•edt	•dd		•he
•re	•tis	•nn		•her
•nt	•ers			•that
•to	•res			•in
•es	•ter			•was
•on	•con			•is
•ed	•ing			•has
•ti	•men			•it
				•him
				•his

Decrypting a Substitution Cipher

- Count relative frequency of letters, digrams, trigrams, endings, doubles, and words in the ciphertext
- If you have enough encrypted text, it can be analyzed and broken by high-speed computers
- But must have a body of encrypted text of sufficient size to permit analysis

Transposition Ciphers

- Repeated letters in the plaintext results in repeated letters in the ciphertext

- Need to break the repetition relationship between letters in the plaintext and the cipher

? transposition cipher

Basic cryptosystems - Transposition

- **Scytale Rod** (Sparta 400BC)

- The ancient Greeks, and the Spartans used this cipher during wars.
- Key = Both sender and receiver use rod with the same diameter.
- The plaintext and the cypher were written on the same leather.

The plaintext could be: "Help me I am under attack".

	H	E	L	P	M
E	I	A	M	U	
N	D	E	R	A	
T	T	A	C	K	

the ciphertext becomes, "HENTEIDTLAEAPMRCMUAK" after unwinding.

Basic cryptosystems - Transposition

רצף המספרים משמש האלגוריתם הוא סוג של מפתח.
כשנשלח אות הראשונה נשלח לפי מספרי הרצף

M	E	G	A	B	U	C	K	Plain Text
7	4	5	1	2	8	3	6	
p	l	e	a	s	e	t	r	Please transfer one million dollars to my Swiss bank account six two.
a	n	s	f	e	r	o	n	
e	m	i	l	l	i	o	n	**Cipher text**
d	o	l	l	a	r	s	t	**AFLLSKSOSELAWAIATOOSSC**
o	m	y	s	w	i	s	s	**TCLNMOMANTESILYNRNNTSO**
b	a	n	k	a	c	c	o	**WDPAEDOBUOERICXB**
u	n	t	s	I	x	t	w	
o	t	w	o	a	b	c	d	

המקבל יודע את רצף המספרים
וכשיקבל את ההודעת הוא יפלים
את הטקסט ויקבל את ההודעה.

One-Time Pad

- Vernam cipher (Gilbert Vernam), 1917
- Mathematically unbreakable –
  - has a property termed perfect secrecy;
  - ciphertext gives absolutely no additional information about the plaintext
- Generated randomly
- Same length as the plaintext
- Never reused

One-Time Pad

מפתח key='cat'= 01100011 01100001 01110100

⊕

המילה שנרצה להצפין P='dog'= 01100100 01100100 01100111

הצופן. Cipher= 00000111 00000101 00010011

'd'=100_{10}=01100100_2
'o'=111_{10}=01101111_2
'g'=103_{10}=01100111_2

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

נעשה קסור של Cipher עם המפתח וכך
נקבל את המידע המקורי (ההודעה).
—אם גודל המפתח [illegible] גדול/שווה מגודל

One-Time Pad

- Choose a random bit string as a key
- Convert plaintext into bitstring
- Compute exclusive-or of the two strings
- Ciphertext contains no redundancy information because every combination is equally likely
- Key must be at least as long as message-overhead

One-Time Pad

- Challenges:
  - Key can not be memorized (too long)
  - Difficulties to create natural Random
  - Large over head
- Why it works?

$$\Pr_{k \Leftarrow \mathrm{K}}[E_k(m_1) = c] = \Pr_{k \Leftarrow \mathrm{K}}[E_k(m_2) = c]$$

Secure cryptosystem

- Problems
  - Statistical patterns
  - Redundancies
  - Key smaller than plaintext
  - Key reused
- Solutions
  - substitutions
  - Transposition
  - Compression
  - Expansion
  - Padding
  - Key mixing
  - Initialization vector (IV)

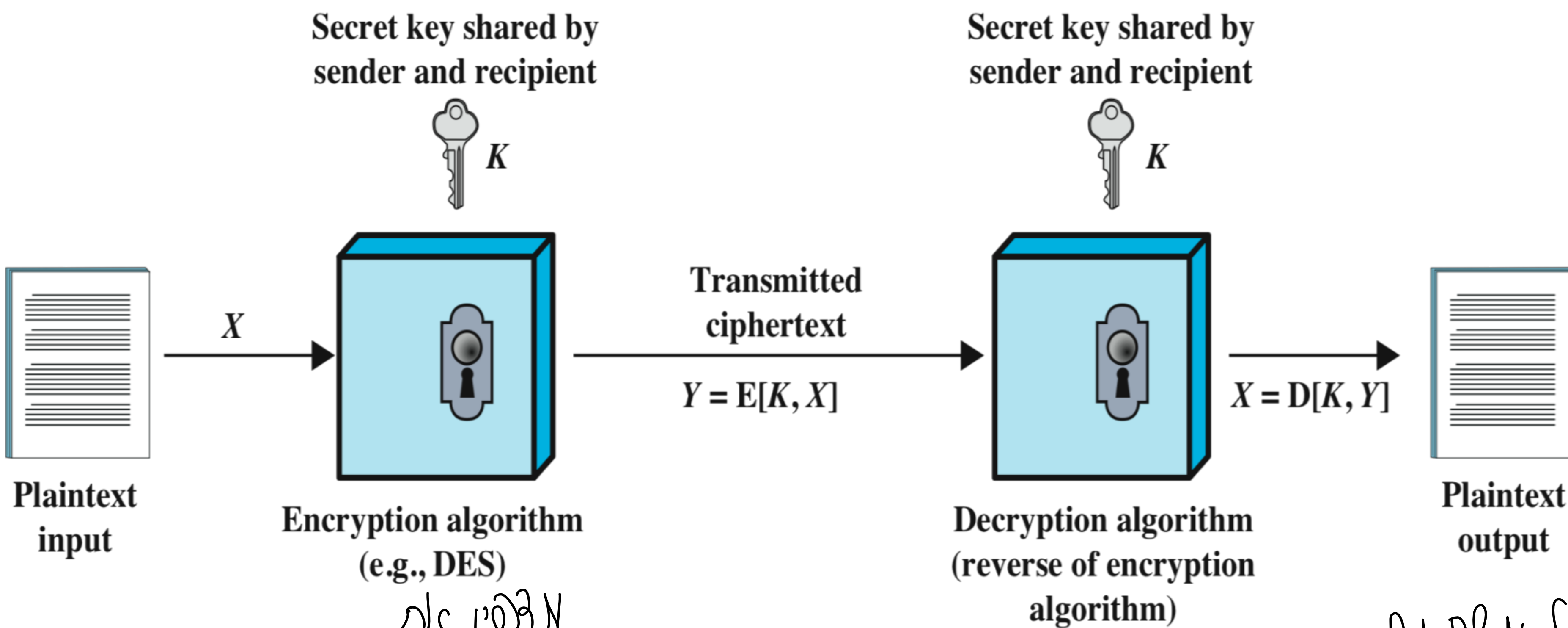
Symmetric Encryption

- The universal technique for providing confidentiality for data at rest/motion
- Also referred to as conventional encryption or single-key encryption
- Two security requirements:
  - Need a strong encryption algorithm
  - Sender and receiver must agree on the secret key in a **secure fashion (handshake)** and must keep the key secured.

איך זה עובד?

Symmetric Encryption [43]

אחרים לא יכולים לפענח.



מצפין את המידע ע"י מפתח שהסכום עם הצד השני.

המקבל משתמש באלגוריתם פענוח שמתאים לאלגוריתם ההצפנה.

- מי שמאזין בין השולח למקבל ואין לו את מפתח ההצפנה לא יכול לפענח.

Symmetric Encryption

- Two classes
  - Block (RC5, RC6, Blowfish…)
  - Stream (RC4)

Cryptanalytic Attacks

כאשר תוקף רוצה לתקוף הוא מתייחס לסוג האלגוריתם ומנסה למצוא
קלשהי שהוא יכול לנצל לצורתו.
וש שתי הנחות של קלשהו:
1- קלשהו מינוש
2- קלשהו

- Rely on:
  - nature of the algorithm
  - some knowledge of the general characteristics of the plaintext
  - some sample plaintext-ciphertext pairs
- Exploits the
  - characteristics of the algorithm to attempt to deduce a specific plaintext or the key being used
- If successful
  - all future and past messages encrypted with that key are compromised

Brute-Force Attack

- Try all possible keys on some ciphertext
- Try until an intelligible translation into plaintext is obtained
- On average half of all possible keys must be tried to achieve success

Average Time Required for Exhaustive Key Search

- Assuming the decryption takes
  - 1 key in 1 micro-second
  - 1M key in 1 micro second

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

Comparison of three Popular Symmetric Encryption Algorithms

- Block cipher - most commonly used in symmetric encryption
- Input = Fixed-size blocks
- Output = Fixed-size equals to the input

	DES	Triple DES	AES
Plaintext block size (bits)	64	64	128
Ciphertext block size (bits)	64	64	128
Key size (bits)	56	112 or 168	128, 192, or 256

DES = Data Encryption Standard
AES = Advanced Encryption Standard

Data Encryption Standard (DES)

- Adopted by NIST in 1977
- Uses
  - 64 bit plaintext block
  - 56 bit key to
  - produce a 64 bit ciphertext block
- DES is the most studied encryption algorithm in existence
  - Electronic Frontier Foundation (EFF) announced in July 1998 that it had broken a DES encryption using the “DES cracker” (250K$ hardware)

How does DES work? (Feistel Network) [597]

- Plaintext is a block of length 2w bits
- K is the Key (seed) from which the subkeys ki are being generated using a generation algorithm.
- Plaintext is divided into two halves of length w: L0 and R0.
- F is a function that combines between Ri and Ki
- N –rounds, and the steps are:

Split the plaintext block into two equal pieces, (L_0, R_0)

For each round $i = 0, 1, \ldots, n$, compute

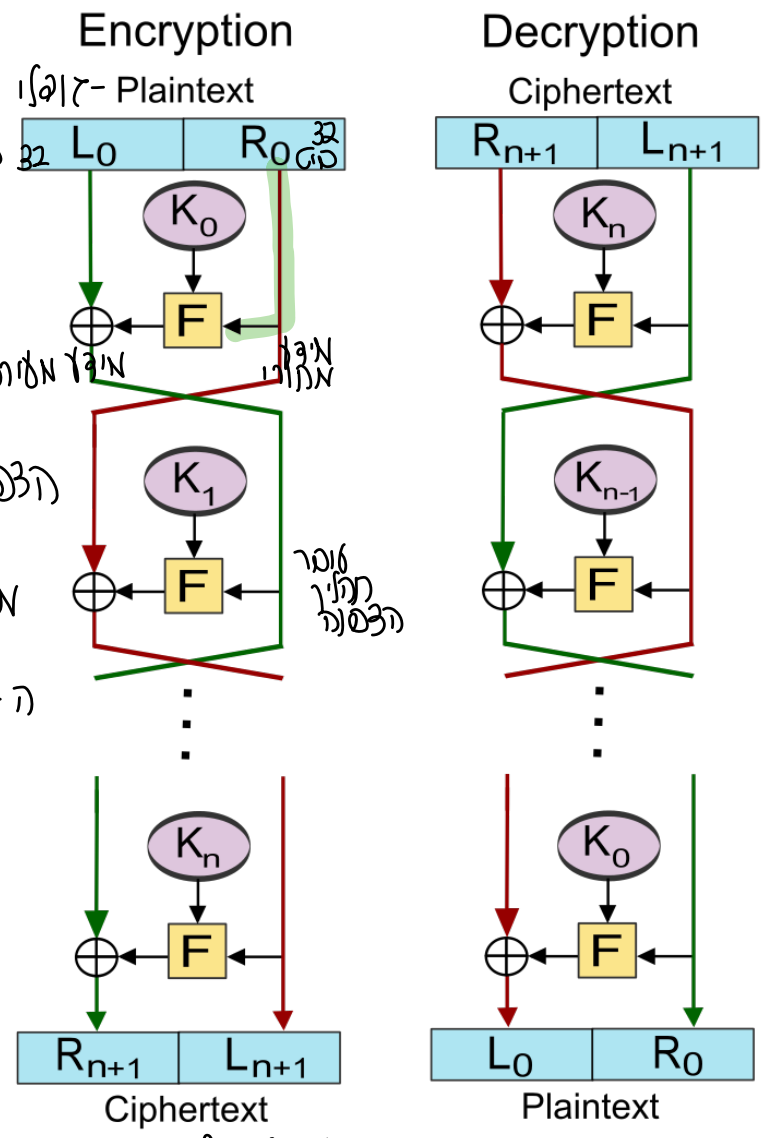
$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \oplus \mathrm{F}(R_i, K_i).$$

Then the ciphertext is (R_{n+1}, L_{n+1}).

Decryption of a ciphertext (R_{n+1}, L_{n+1}) is accomplished by computing for $i = n, n-1, \ldots, 0$

$$R_i = L_{i+1}$$
$$L_i = R_{i+1} \oplus \mathrm{F}(L_{i+1}, K_i).$$

Then (L_0, R_0) is the plaintext again.



Triple DES (3DES)

- Repeats basic DES algorithm three times using either two or three unique keys
- First standardized for use in financial applications in ANSI standard X9.17
- Attractions:
  - 168-bit key length overcomes the vulnerability to brute-force attack of DES
  - underlying encryption algorithm is the same as in DES
- Drawbacks:
  - algorithm is slow in software
  - uses a 64-bit block size

3DES – Scheme [630]

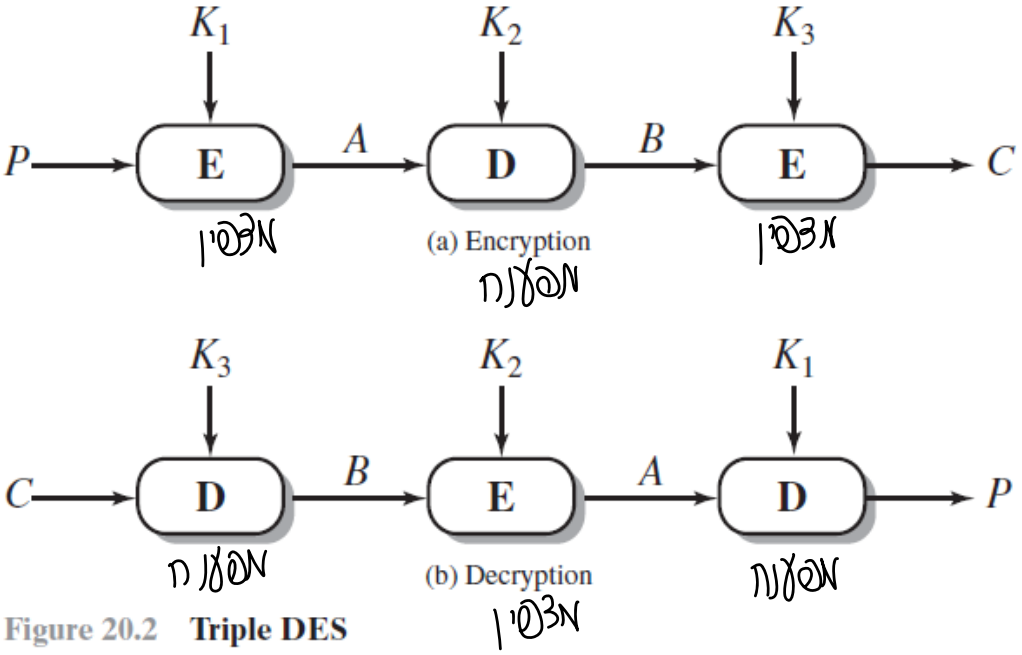


Figure 20.2 **Triple DES**

3DES uses three keys and three executions of the DES algorithm. The function follows an encrypt-decrypt-encrypt (EDE) sequence (Figure 20.2a):

$$C = \mathrm{E}(K_3, \mathrm{D}(K_2, \mathrm{E}(K_1, p)))$$

where

C = ciphertext
P = plaintext
E[K, X] = encryption of X using key K
D[K, Y] = decryption of Y using key K

Decryption is simply the same operation with the keys reversed (Figure 20.2b):

$$P = \mathrm{D}(K_1, \mathrm{E}(K_2, \mathrm{D}(K_3, C)))$$

Advanced Encryption Standard(AES)

אלגוריתם אמין וחזק יותר. גודל הבלוק הוא 128 ביט, תומך במספר אורכי מפתחות
בגדלים שונים.

- NIST call for proposal for a new Advanced Encryption Standard (AES) in 1997
- Selected Rijndeal in Nov. 2001 (also known as AES)
- Advantages:
  - Faster than 3DES
  - security strength equal or better than 3DES
  - symmetric block cypher
  - Larger Blocks: 128bit blocks
  - Longer keys: **128**/192/256 bit key

Symmetric Encryption Types

- Block Cipher
  - Processes the input one block of elements at a time and produces an output block for each input block.
  - Implemented via different modes.
- Stream cipher
  - Encrypts plaintext one byte at a time (also other units)
  - Processes the input elements continuously

Modes of Operation [612]

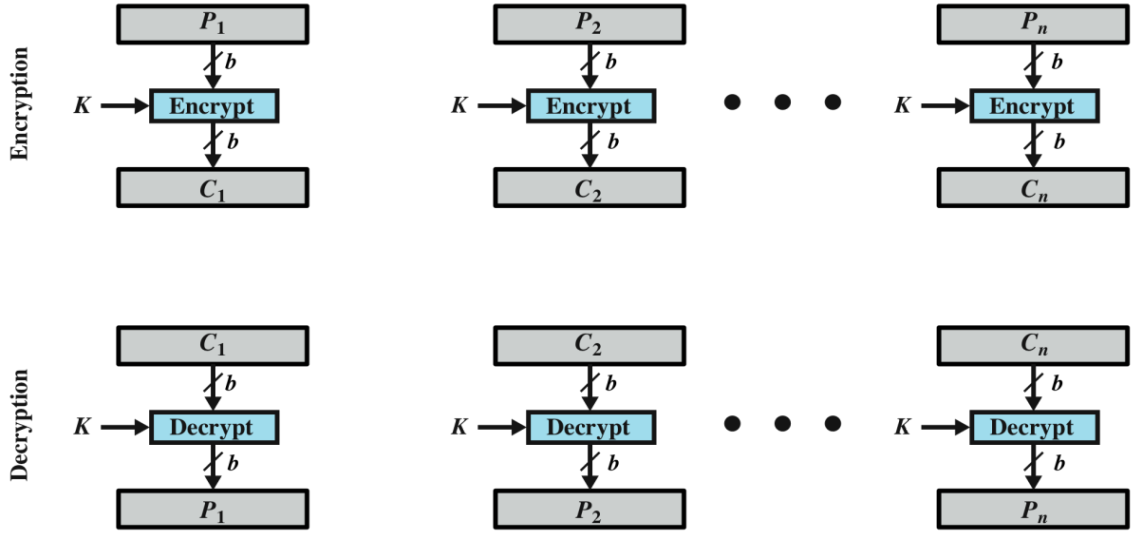
Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of 64 plaintext bits is encoded independently using the same key.	•Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of ciphertext.	•General-purpose block-oriented transmission •Authentication
Cipher Feedback (CFB)	Input is processed *s* bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	•General-purpose stream-oriented transmission •Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding DES output.	•Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	•General-purpose block-oriented transmission •Useful for high-speed requirements

ECB

Block Cipher Encryption

הצורה הפשוטה , לוקחים את ההודעה הארוכה ומחלקים לבלוקים וכל בלוק בנפרד.

- Processes the input one block of elements at a time and produces an output block for each input block
- Plaintext is handled using Electronic Code Book (ECB) For lengthy messages ECB mode is not secure especially if:
  - it is known that messages starts with certain predefined fields
  - keys are being reused



Practical Security Issues

- Typically symmetric encryption is applied to a unit of data larger than a single 64-bit or 128-bit block
- Electronic codebook (ECB) mode is the simplest approach to multiple-block encryption
  - each block of plaintext is encrypted using the same key
  - cryptanalysts may be able to exploit regularities in the plaintext
- Modes of operation
  - Alternative techniques developed to increase the security of symmetric block encryption for large sequences
  - Overcomes the weaknesses of ECB
  - To increase security for lengthy messages we use Stream Cipher

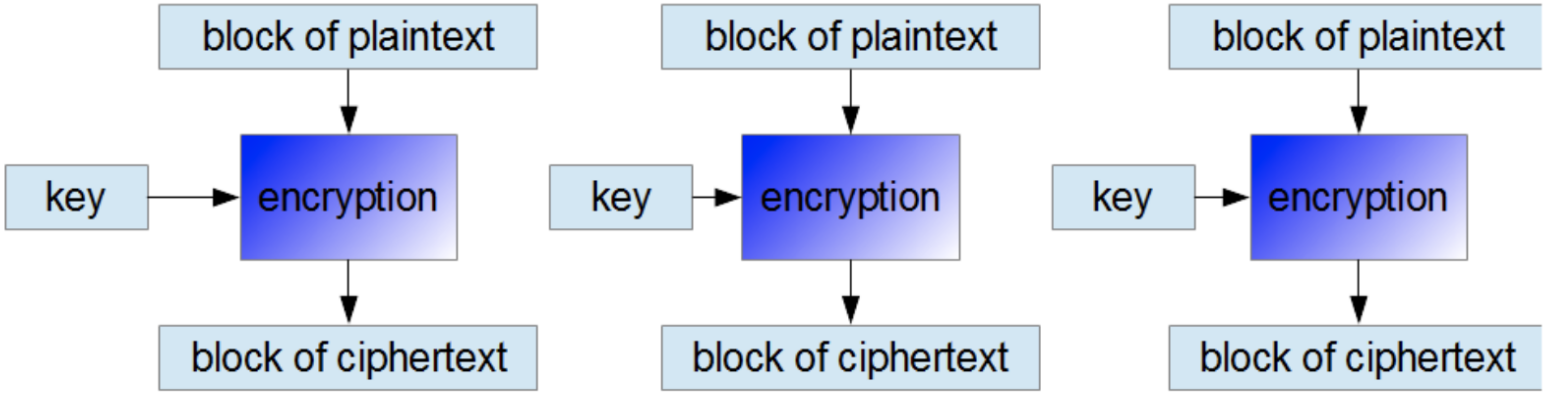
Electronic Codebook (ECB)

- Simplest mode
- Plaintext is handled *b* bits at a time and each block is encrypted using the same key
- "codebook" because have unique ciphertext value for each plaintext block

בעיה: אם plaintext מהה לאחד אחר לא יכול לקרות משהו מעניין.

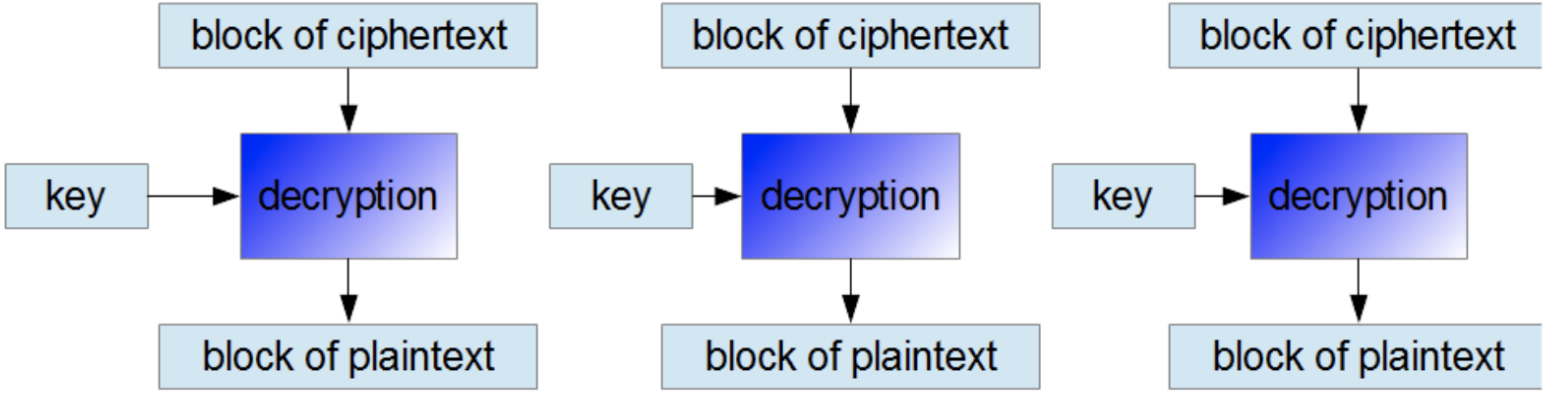
Electronic Codebook (ECB)

מצפינים אחד אחרי השני.



Encryption in the ECB mode

-המפתח והטקסט מגיעים, יש מנגנון שרה.



Decryption in the ECB mode

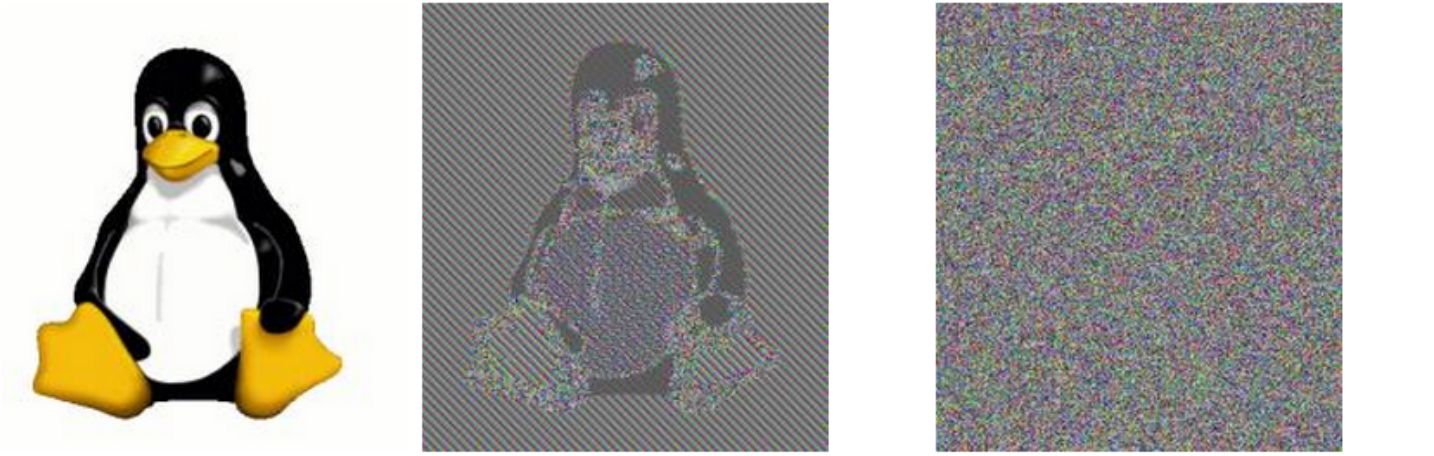
סה"כ תלוחה של בלוקים, נצפין כל בלוק בנפרד.

ECB Analysis

Security:

– EFB is not secure, since repeated plaintext is seen in repeated ciphertext

– A typical example of this weakness is by encoding a bitmap image – as can be noticed, ECB mode cannot blur the plaintext.

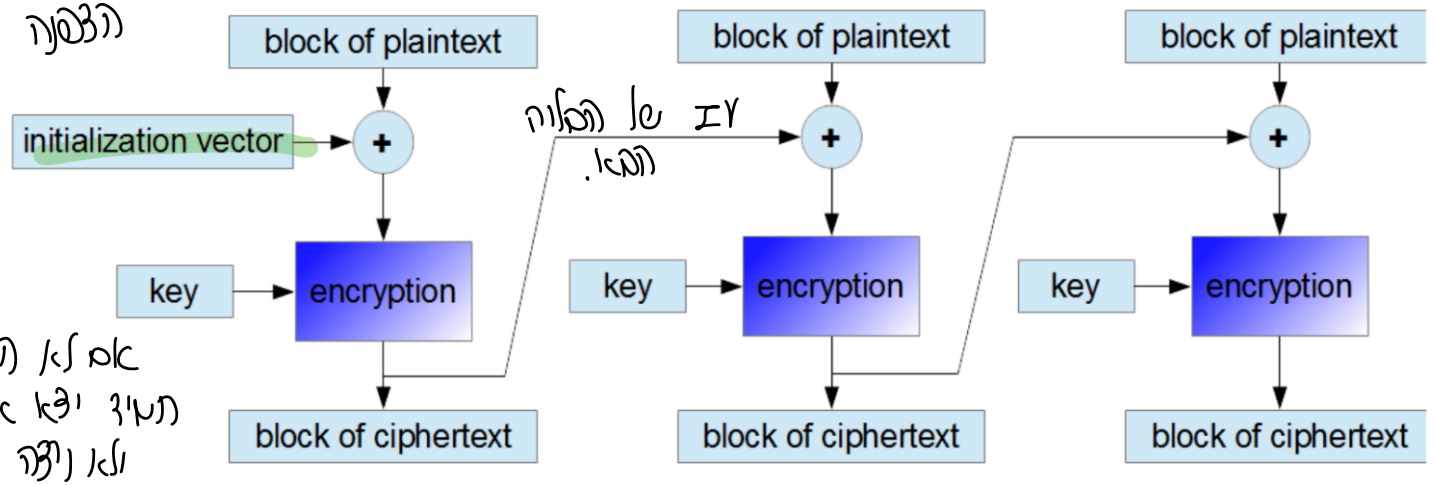


Original *Encrypted using ECB mode* *Modes other than ECB result in pseudo-randomness*

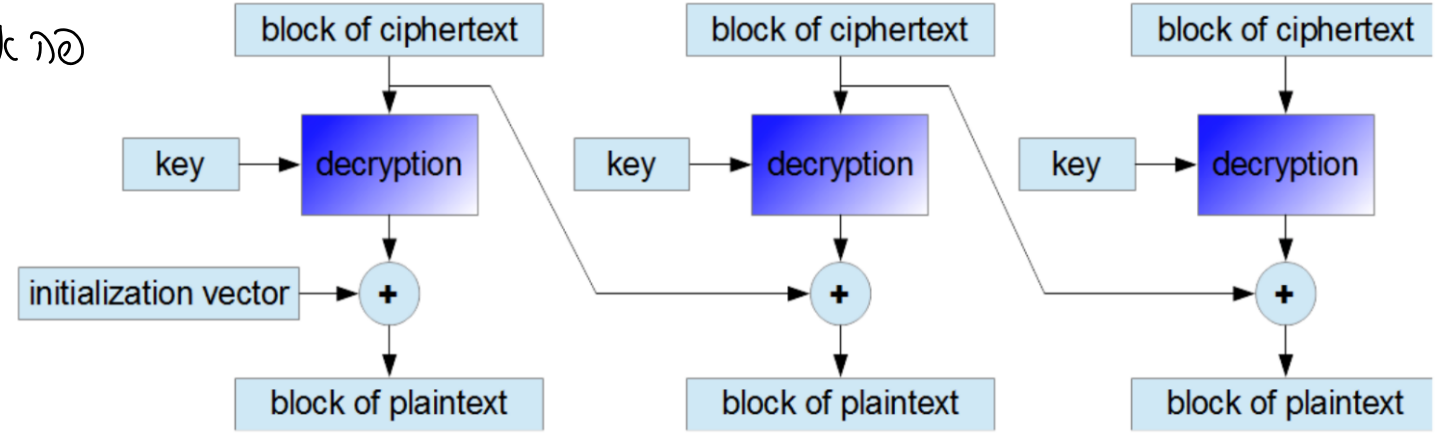
ECB Analysis

- **Speed:** Each plaintext/ciphertext block is encrypted/decrypted separately. Thus, it is possible to encrypt and decrypt by **using many threads** simultaneously.
- **Effect of an error:** Each plaintext/ciphertext block is encrypted/decrypted separately. Thus, an error in **one block** does not affect other blocks.
- **Reoccurrences of key:** the same key are used for all encryption and decryption blocks.

Cipher Block Chaining (CBC)



Encryption in the CBC mode



Decryption in the CBC mode

CBC Analysis

Security: the CBC mode is secure, repeated plaintext will not be seen in repeated ciphertext.

Reoccurrences of key: the same key are used for all encryption and decryption blocks.

Speed:

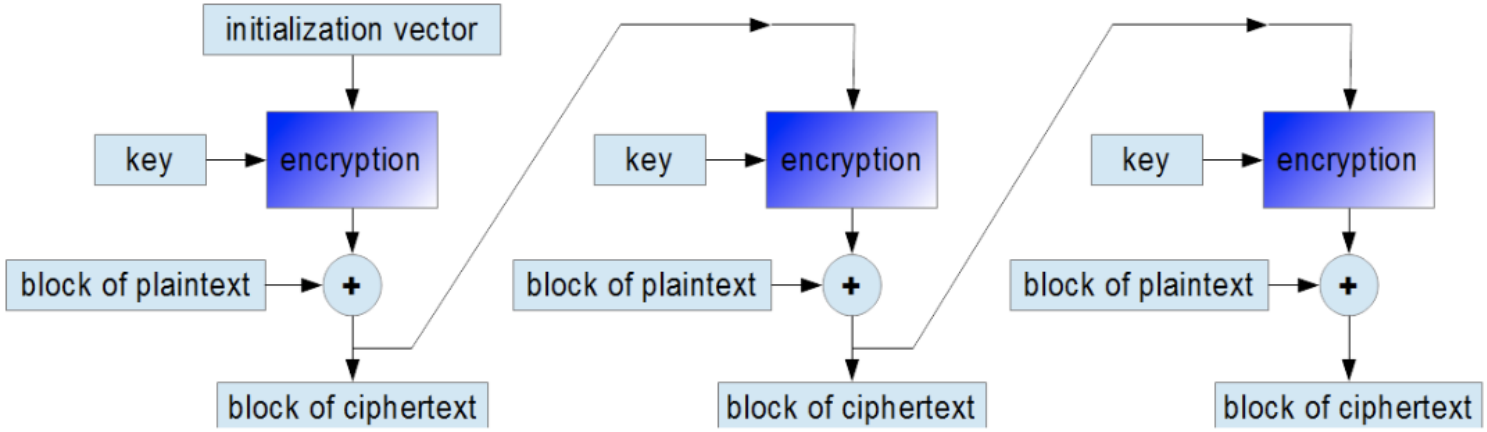
- **Encryption:** Each ciphertext block depends on the previous one. Thus, encryption in CBC mode can only be performed by using **one thread**.
- **Decryption:** Because the receiver knows all the ciphertext blocks just after obtaining the encrypted message, he can decrypt the message using **many threads simultaneously**.

CBC Analysis

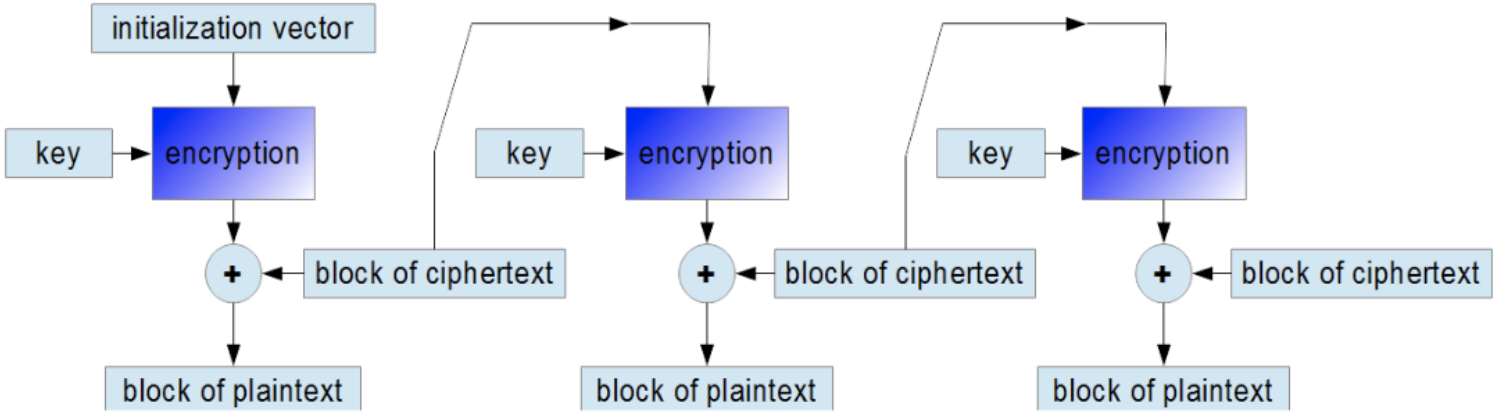
Effect of an error:

- **Encryption:** If one bit of a plaintext message is damaged (for example because of some earlier transmission error), **all subsequent ciphertext blocks will be affected**.
- **Decryption:** If one ciphertext bit is damaged, only **two received plaintext blocks will be damaged**. It might be possible to recover the data.

Cipher Feedback (CFB)



Encryption in the CFB mode



Decryption in the CFB mode

CFB Analysis

- **Security:** The CFB mode is similar to CBC. The main difference is that one should encrypt ciphertext data from the previous round and then add the output to the plaintext bits. It does not affect the cipher security but it results in the fact that **the same encryption algorithm should be used during the decryption** process.

CFB Analysis

Reoccurrences of key: the same key are used for all encryption and decryption blocks.

Speed:

- **Encryption:** Encryption in CFB mode can be performed only by using **one thread**.
- **Decryption:** As in CBC mode, one can decrypt ciphertext blocks **using many threads** simultaneously.

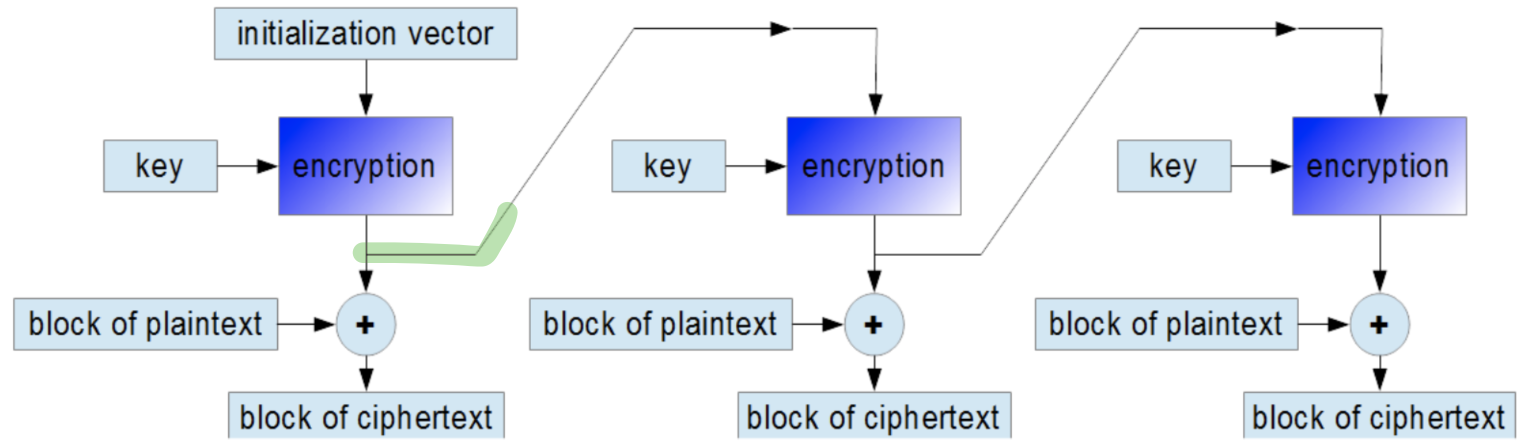
CFB Analysis

Effect of an error:

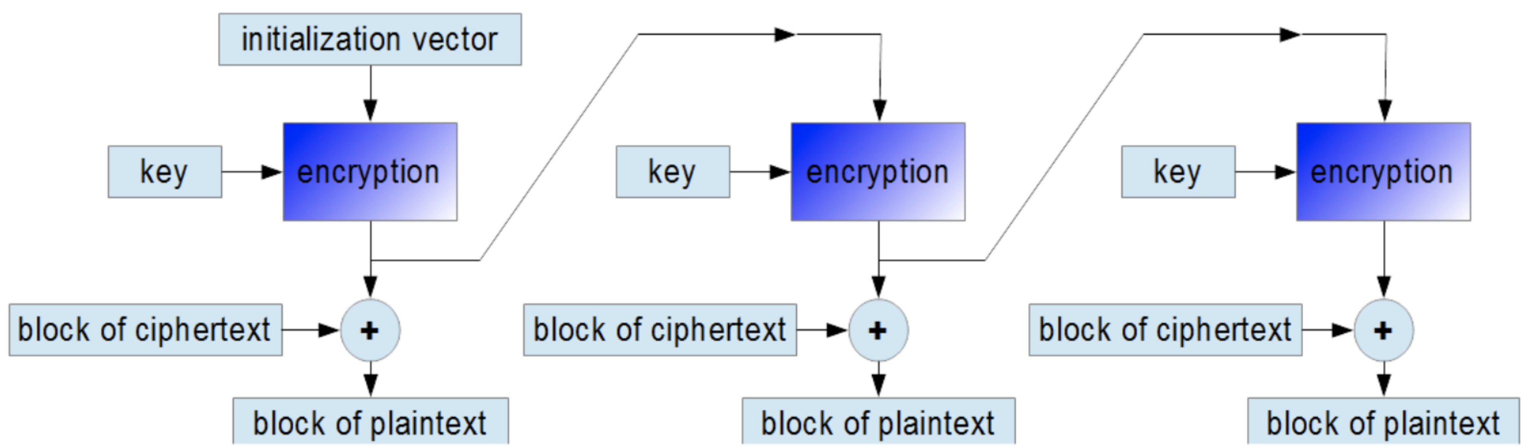
- **Encryption:** If one bit of a plaintext message is damaged, the corresponding ciphertext block and **all subsequent ciphertext blocks will be damaged**.
- **Decryption:** If one ciphertext bit is damaged, only **two received plaintext blocks will be damaged**.

ההבדל מהאלגוריתם הקודם זה שה-IV של הבלוק הבא קורה לפני הקסור.

Output Feedback (OFB)



Encryption in the OFB mode



Decryption in the OFB mode

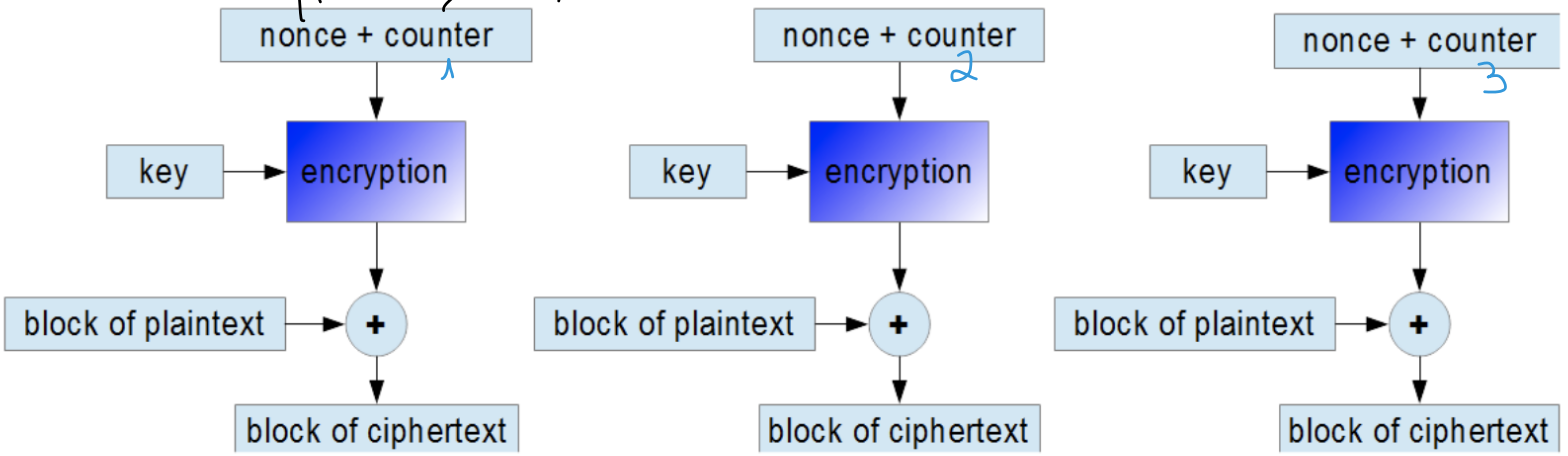
OFB Analysis

- **Security:** The biggest drawback of OFB is that the **repetition of encrypting the initialization vector** may produce the same state that has occurred before. It is an unlikely situation but in such a case the plaintext will start to be encrypted by the same data as previously.
- **Reoccurrences of key:** Algorithms that work in the OFB mode create keystream bits that are used for encryption subsequent data blocks. In this regard, the way of working of the block cipher becomes **similar to the way of working of a typical stream cipher**.

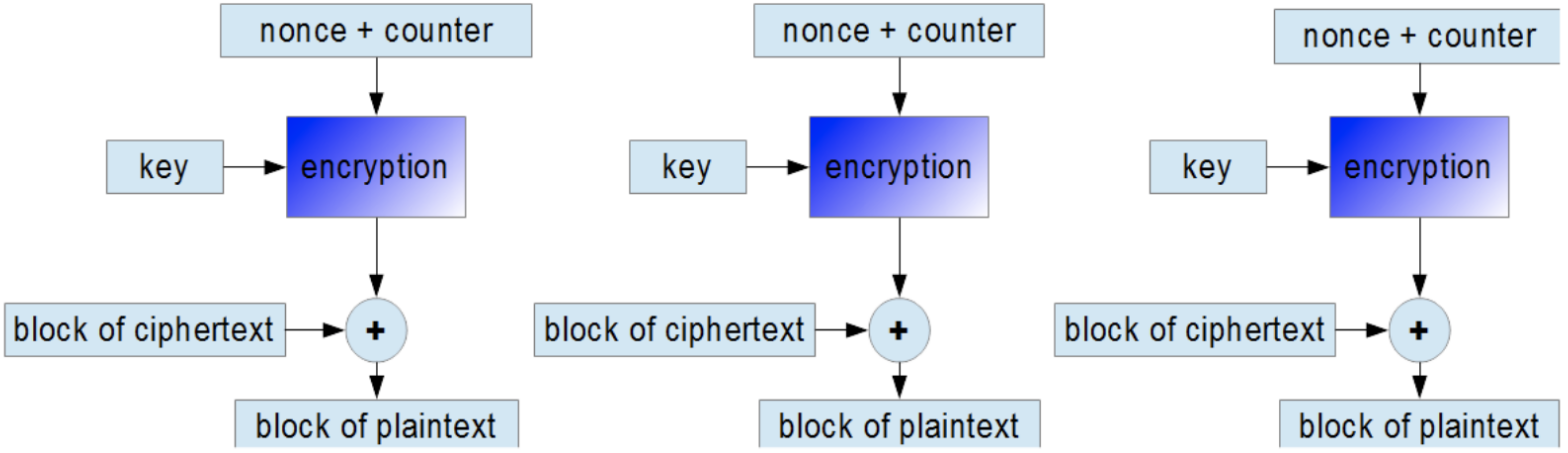
OFB Analysis

- **Speed:** Because of the continuous creation of keystream bits, both encryption and decryption can be performed using only **one thread** at a time. Similarly, as in the CFB mode, both data encryption and decryption uses the same cipher encryption algorithm.
- **Effect of an error:** If one bit of a plaintext or ciphertext message is damaged (for example because of a transmission error), **only one corresponding ciphertext** or respectively plaintext bit is damaged as well. It is possible to use various correction algorithms to restore the previous value of damaged parts of the received message.

Counter (CTR)



Encryption in the CTR mode



Decryption in the CTR mode

CTR Analysis

- **Security:** In this mode, subsequent values of an increasing counter are added to a *nonce* value (the nonce means a number that is unique: *number used once*) and the results are encrypted as usual. The **nonce plays the same role as initialization vectors**.
- **Reoccurrences of key:** Using the CTR mode makes block cipher way of working **similar to a stream cipher**. As in the OFB mode, keystream bits are created regardless of content of encrypting data blocks.

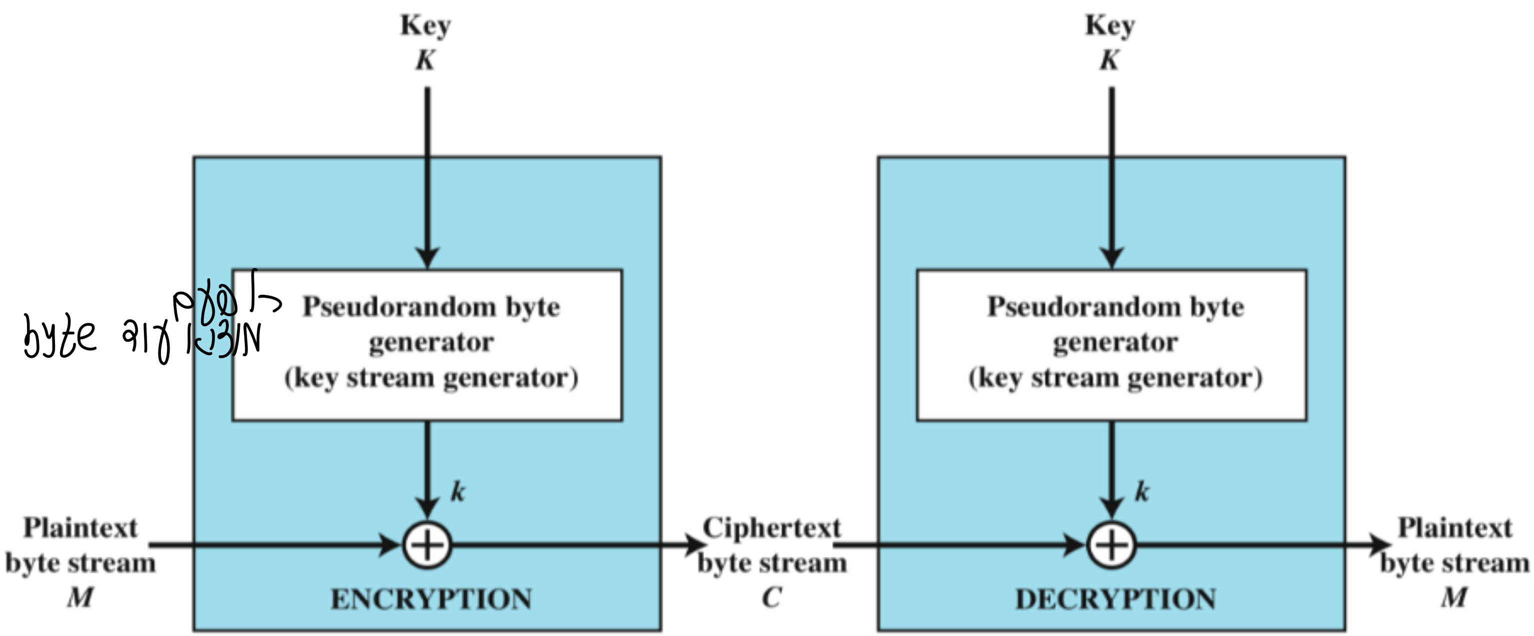
CTR Analysis

- **Speed:** It is one of the most popular block ciphers modes of operation. **Both encryption and decryption** can be performed using **many threads** at the same time.
- **Effect of an error:** **If one bit of a plaintext** or ciphertext message is damaged, only **one corresponding output bit** is damaged as well. Thus, it is possible to use various correction algorithms to restore the previous value of damaged parts of received messages.

מייצר רצף של ביטים, אין צורך של padding.

Stream Cipher Encryption – RC4

- Encrypts plaintext one byte at a time (also other units)
- Produces output one element at a time
- Processes the input elements continuously



חייב סנכרון מלא בין הריבוע
הלבן מצד המשדר לצד הקולט.

Stream Cipher Encryption – RC4

- Almost always faster
- Pseudorandom stream is one that is unpredictable without knowledge of the input key
- Keys cannot be reused
- Generate pseudo random keys (key Stream)

Message Confidentiality

VS

Message Authentication

עד עכשיו עסקנו בלהסתיר את ההודעה מאדם לא מורשה.

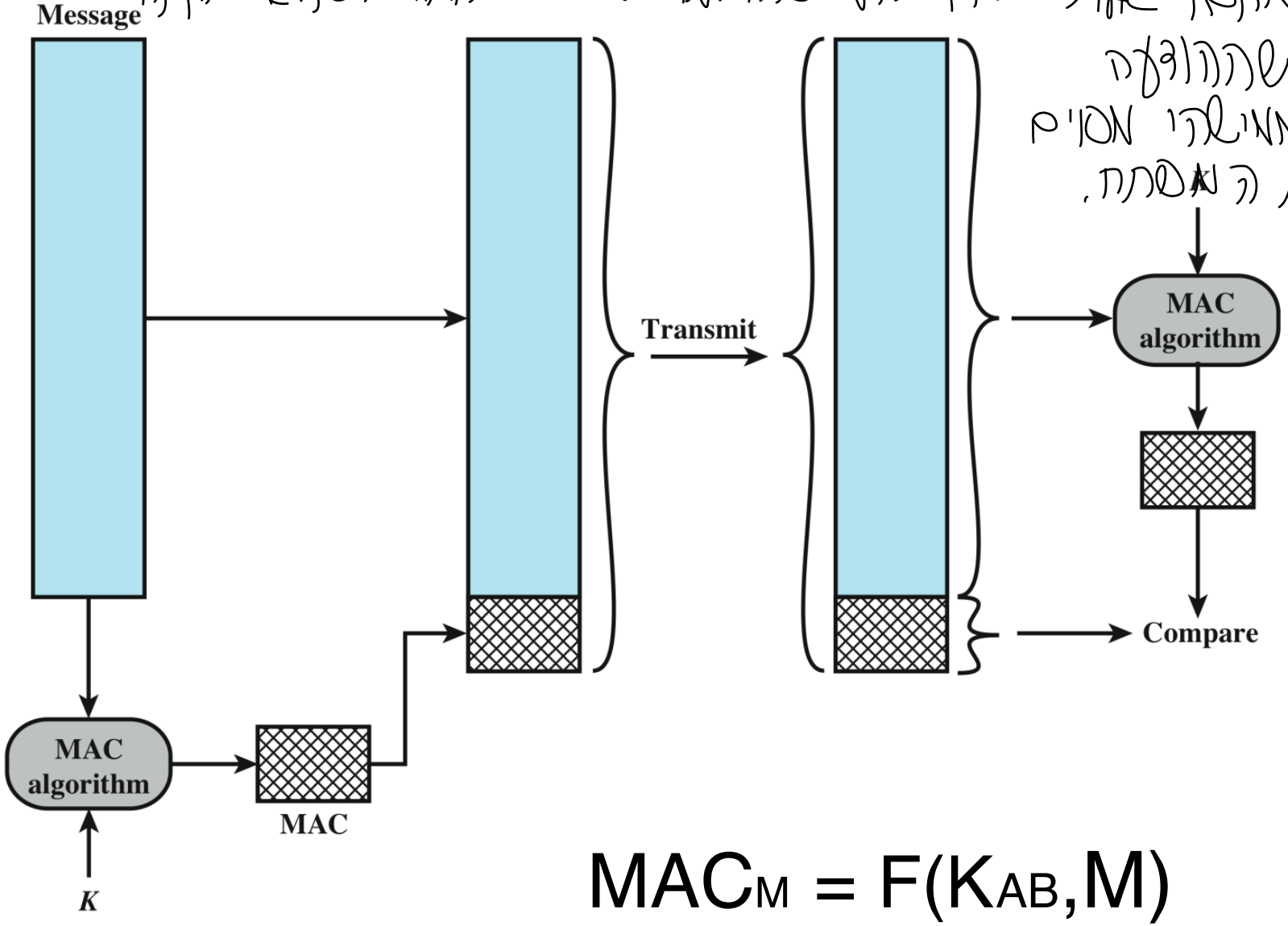
Message Authentication

- Protects against active attacks
- Verifies that the received message is authentic
  - content have not been altered
  - from authentic source
  - timely and in correct sequence
- can use conventional (Symmetric) encryption
  - Only sender and receiver share a key

Message Authentication without Encryption

- Symmetric Encryption only for message authentication is exposed to Block reorder attack (e.g in ECB)
- Thus using a generated tag and appending it to each message protect from this attack as well:
  - Message Authentication Code
  - One Way Hash Function

Message Authentication Codes (MAC)



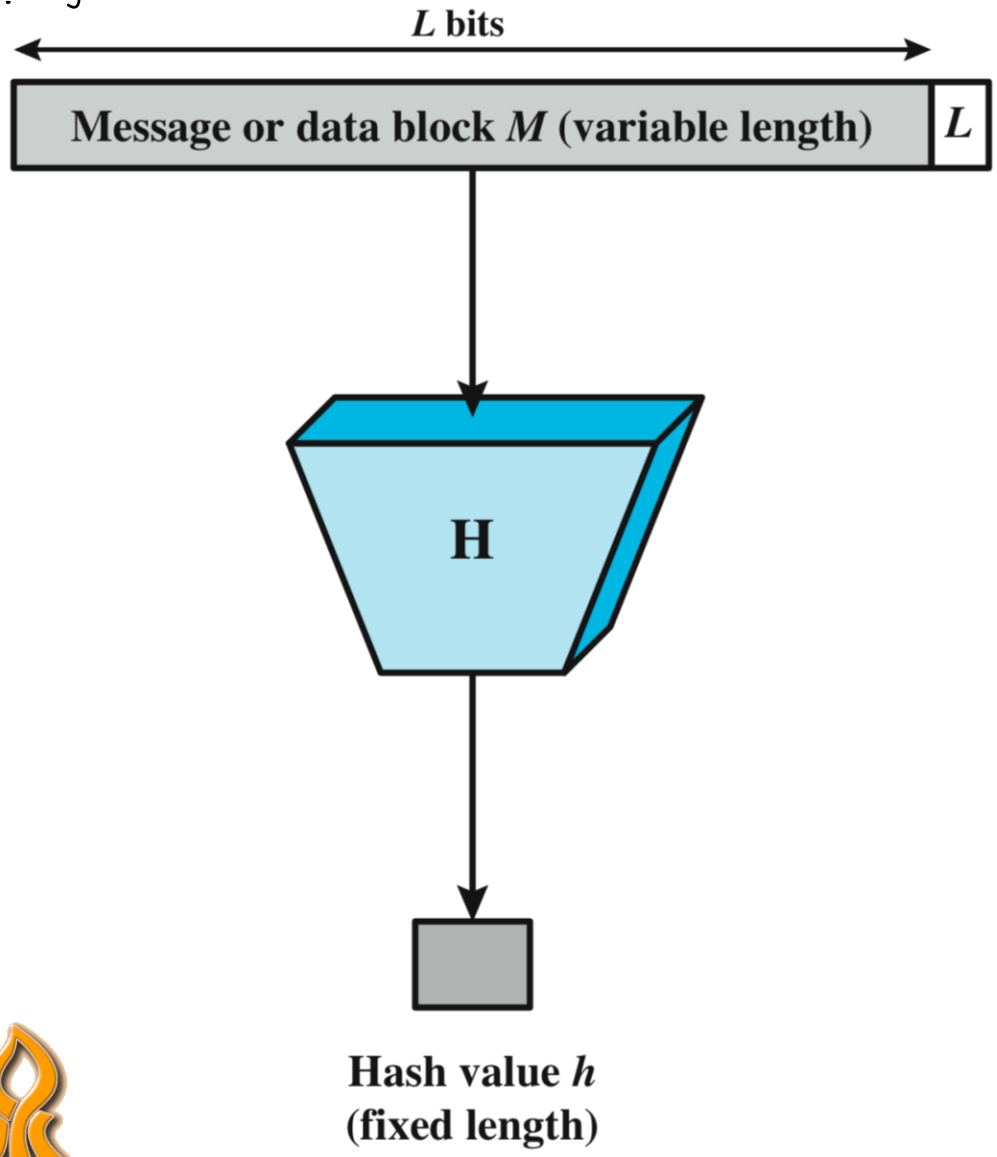
$$MAC_M = F(K_{AB}, M)$$

Message Authentication Codes (MAC)

רק למי שיש את ההצפנה יכול לפענח את הקוד.

- Only the sender and receiver share the secret key K_{AB}
- If the sender find a match between the received MAC and his calculated code then:
  - Message wasn't altered: attacker can alter the message but not the code since doesn't have the secret key
  - Sender authenticity –no one else is able to produce the same MAC due to the secret key
  - Message on correct order - attacker cannot alter the sequence number (X.25,HDLC, TCP) within a message
- Note that MAC algorithms are not required to be reversible

One-Way Hash Function

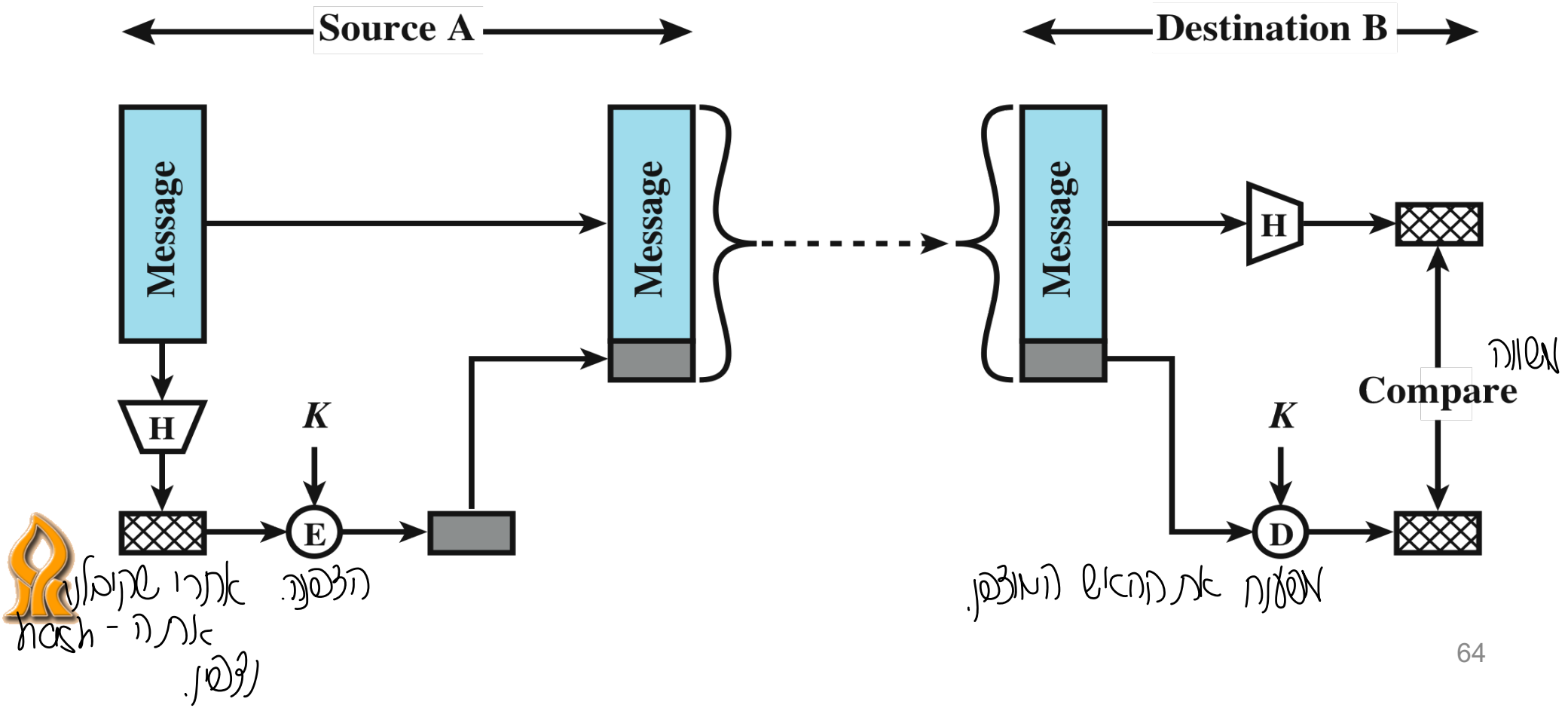


- One way since it is not reversible
- M message with variable length
- H – one way hash function
- H does not use a secret key
- H(M) – fixed message digest that is added to the original message
- We add to H(M) the length of the original message in bits
- Hash is more lightweight than MAC
  - Sender authenticity is not guaranteed

Message Authentication Using a One-Way Hash Function – HMAC (MD2, MD4, MD5, SHA-1(160)/256/512) [53]

* כולם רואים את ההצפנה אבל אף אחד לא יודע את המפתח.

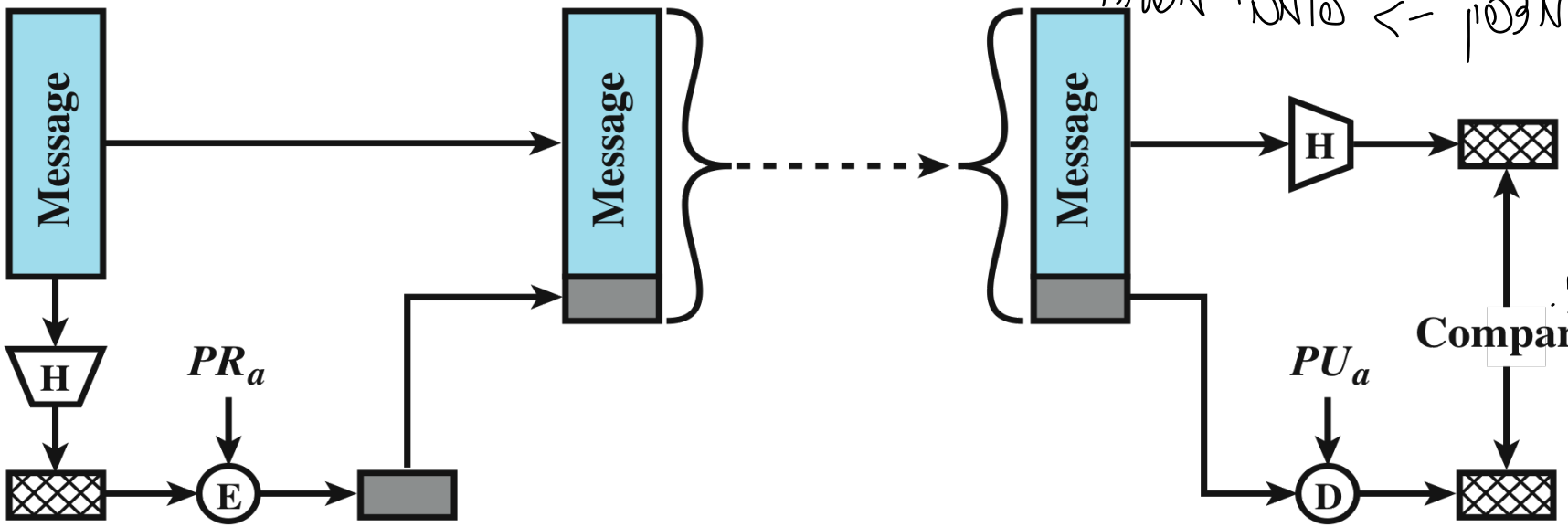
We encrypt the digest only using the symmetric key



Message Authentication Using a One-Way Hash Function – HMAC (MD2, MD4, MD5, SHA-1(160)/256/512) [53]

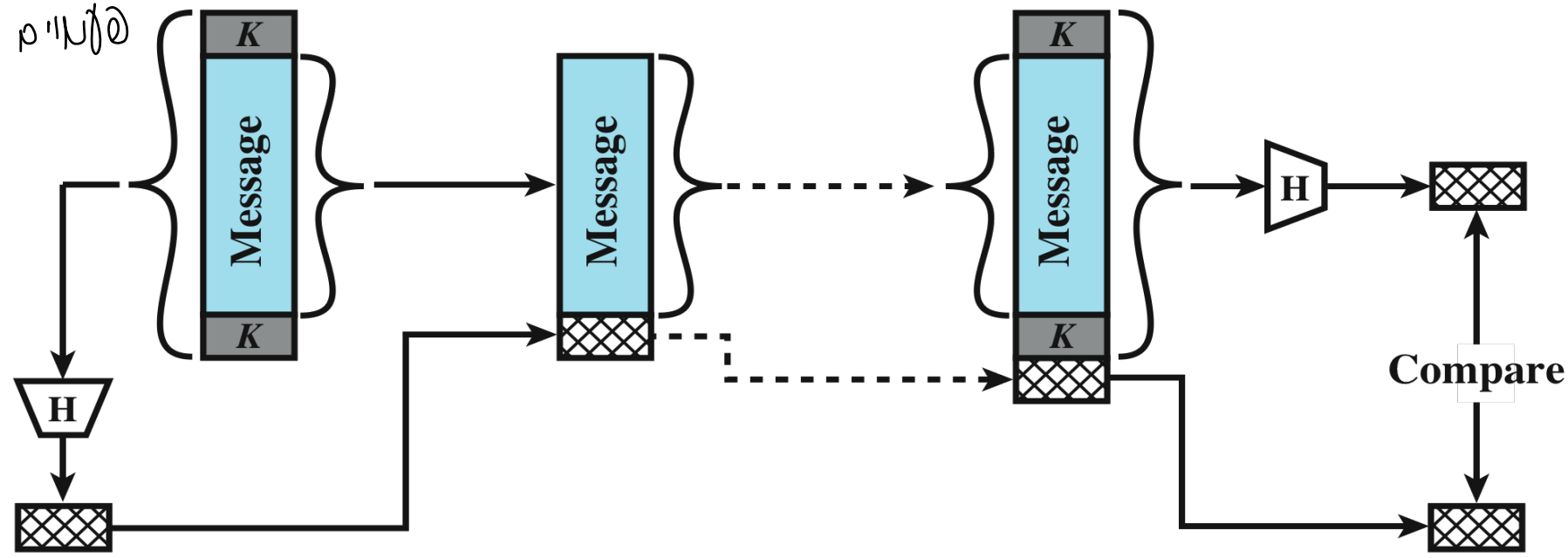
We encrypt the digest only using our private key

- used as digital signature
- no need for key distribution



Message Authentication Using a One-Way Hash Function – HMAC (MD2, MD4, MD5, SHA-1(160)/256/512) [53]

We use secret value K that is agreed on the sender and receiver, the secret is not sent therefore it is difficult for an attacker to create H(m)



Hash Function Requirements

ניתן להפעיל על כל אורך הנקלה

- Can be applied to a block of data of any size
- Produces a fixed-length output
- H(x) is relatively easy to compute for any given x

כלומי ניתן לחשב במהירות סבירה את ערכו של הגיבוב המתאים.

- One-way or pre-image resistant
  - computationally infeasible to find x such that H(x) = h

Hash Function Requirements

- Second pre-image resistant or weak collision resistant
  - Given x, it computationally infeasible to find y ≠ x such that H(y) = H(x)
  - Not alternative message can be found
- Collision resistant or strong collision resistance (Strong Hash)
  - computationally infeasible to find any pair (x, y) such that H(x) = H(y)

א' אפשר ליצר 2 הודעות שונות עליו שיש להם את אותו Hash.

Security of Hash Functions

- Two approaches to attacking a secure hash function:
  - Cryptanalysis
    - exploit logical weaknesses in the algorithm
  - Brute-force attack
    - strength of hash function depends solely on the length of the hash code produced by the algorithm
- SHA (Secure hash function) most widely used hash algorithm – 160 bits

Security of Hash Functions

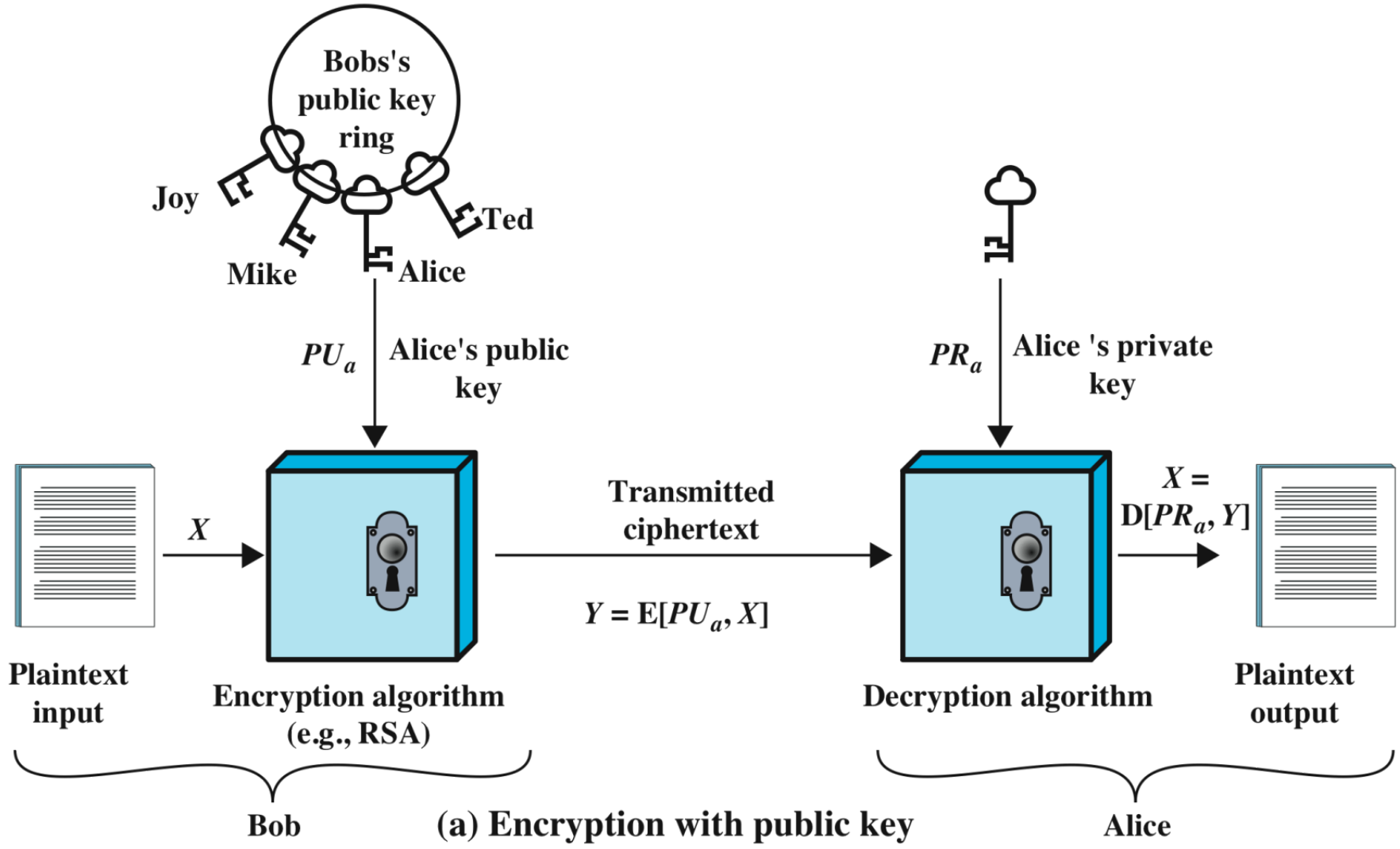
- Additional secure hash function applications:
  - Passwords - hash of a password is stored by an operating system
  - intrusion detection - store H(F) for each file on a system and secure the hash values

שימוש נפוץ בפונקציית hash, לאחסן סיסמאות במערכת מאובטחת.
כל סיסמא שמגיעים מעבירים דרך hash,
מי שתוקף לא יכול לדעת את הסיסמא.

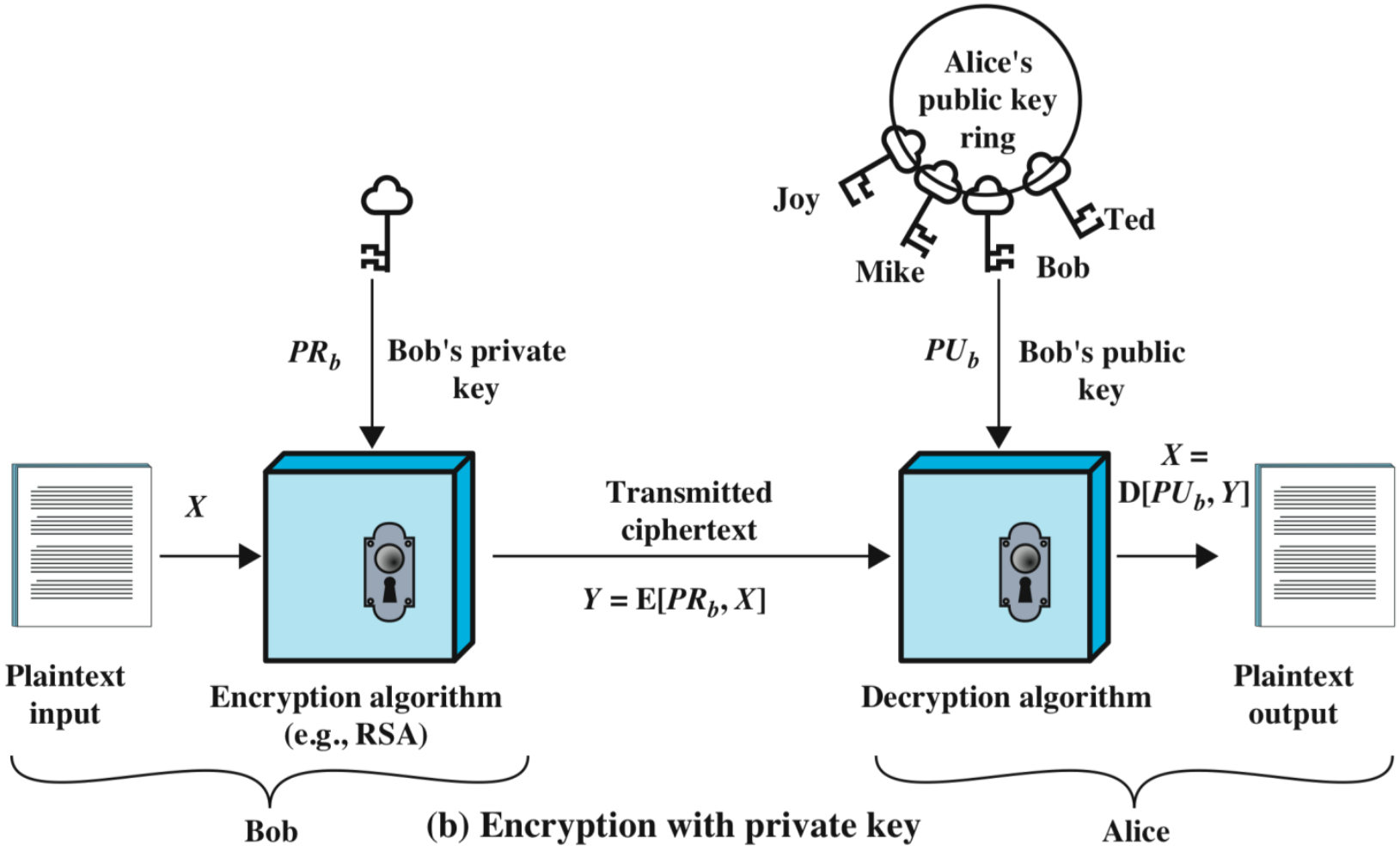
Public-Key Encryption

- Publicly proposed by Diffie and Hellman in 1976
- Asymmetric
  - Uses two separate keys
  - Public key (PUK) and private key (PRK)
  - Public key is made public for others to use
  - There is a relation between the private and public key.
- Some form of protocol is needed for distribution

Public-Key Encryption [58] (Confidentiality)



Private-Key Encryption (Authentication \ data integrity)



(b) Encryption with private key

Requirements for Public-Key Cryptosystems

- Computationally easy to create key pairs
- Computationally easy for sender knowing public key to encrypt messages
- Computationally easy for receiver knowing private key to decrypt ciphertext

Requirements for Public-Key Cryptosystems

- Computationally infeasible for opponent to determine private key from public key
- Computationally infeasible for opponent to otherwise recover original message
- Useful if either key can be used for each role (encryption/Decryption)

Asymmetric Encryption Algorithms

RSA (Rivest, Shamir, Adleman)

- developed in 1977
- most widely accepted and implemented approach to public-key encryption
- block cipher in which the plaintext and ciphertext are integers between [0 , *n*-1] for some *n*
- Was broken when key was 428 bits, currently 1024 is considered strong enough

Diffie-Hellman key exchange algorithm

- enables two users to securely reach agreement about a shared secret that can be used as a secret key for subsequent symmetric encryption of messages
- limited to the exchange of the keys (handshake)
- Not useful for full message encryption

Digital Signature Standard (DSS)

- developed in 1991
- provides only a digital signature function with SHA-1
- cannot be used for encryption or key exchange

Elliptic curve cryptography (ECC)

- security like RSA, but with much smaller keys – thus faster
- Used in e-commerce sites since they conduct very large number of transactions

Applications for Public-Key Cryptosystems [59]

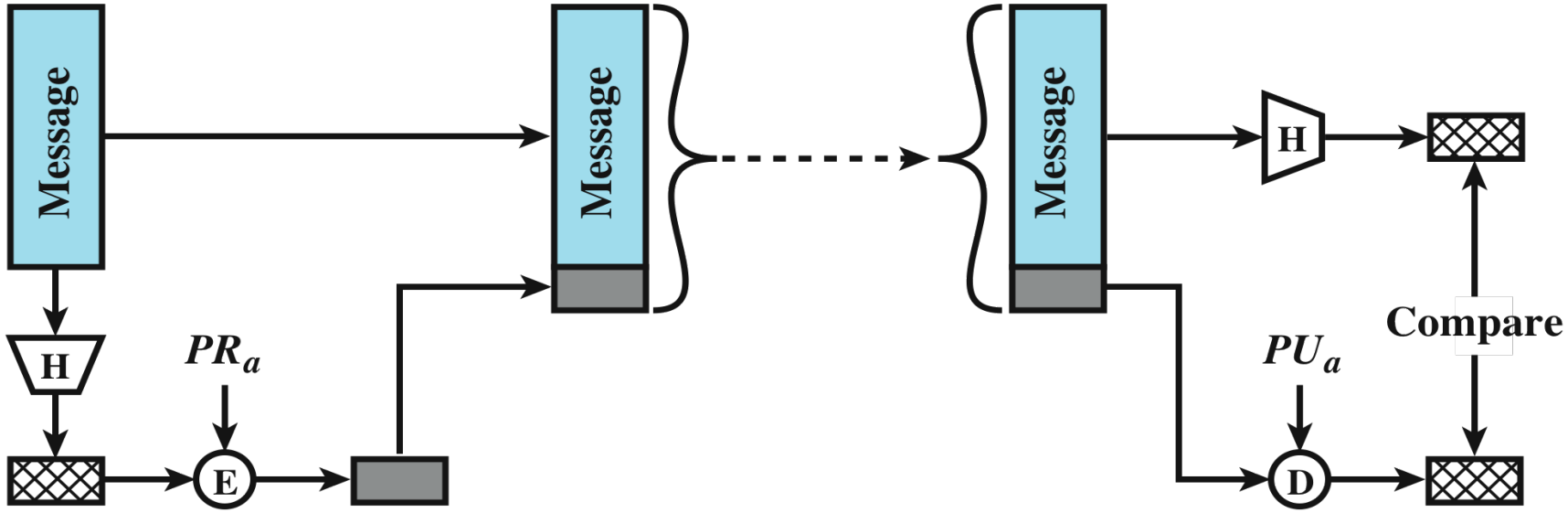
Algorithm	Digital Signature	Symmetric Key Distribution	Encryption of Secret Keys
RSA	Yes	Yes	Yes
Diffie-Hellman	No	Yes	No
DSS	Yes	No	No
Elliptic Curve	Yes	Yes	Yes

Digital Signatures

- Used for authenticating of:
  - Source
  - Data integrity
- Created by encrypting hash code with private key
- Does not provide confidentiality
  - Even in the case of complete encryption
  - Message is safe from alteration but not eavesdropping

Digital Signatures

- encrypt the digest only using private key
  - used as digital signature
  - no need for key distribution



(b) Using public-key encryption

Public Key Certificates

- The weakness of PKE:
  - anyone can forge a publication of the Bob's public key and
  - by the time it is discovered he can read messages aimed at Bob and pretend to be Bob as well
- We need a **trusted third party** – certificate authority CA

Public Key Certificates