

# אבטחת מידע



מבחן סיום – סמסטר א' – שנה 3.

פברואר 2009

יוסי מתתיהו

## אבטחת מערכת ואבטחת רשת - מטרות

אימות זהות השולח.

■ Authentication -

מניעת גניבה של מידע ברשת או מניעת ניתוח של המידע אשר מעבירים

■ Confidentiality -

ברשת, כלומר, יתכן שהפורץ לא הצליח לפרוץ את המידע אך הוא לומד ממנו

דפוס התנהגות.

התמודדת אפשרית עם מקרה כזה היא שליחת מידע סרק ("זבל").

חיסרון של שיטה זו - יקר - הגודל של המידע גודל...

■ Data Integrity

■ Availability

■ Non-Repudiation - מניעת הכחשה :

כאשר אני שולח מידע איך אני יכול לדעת אם מה ששלחתי זה מה שהצד

השני קיבל?

תשובה: בעזרת חתימה.

■ הערה: **אם כך מה ההבדל בין Authentication ל- Non-Repudiation ?**

**במקרה של אימות - אימות של המידע נעשה אצל הצד המקבל.**

**במקרה של מניעת הכחשה - אימות המידע נעשה ע"י צד שלישי - "שופט".**

### להצפנה ברשת צריך:

• פרוטוקולי תיאום בין שני הצדדים.

• נוהלי הפעלה.

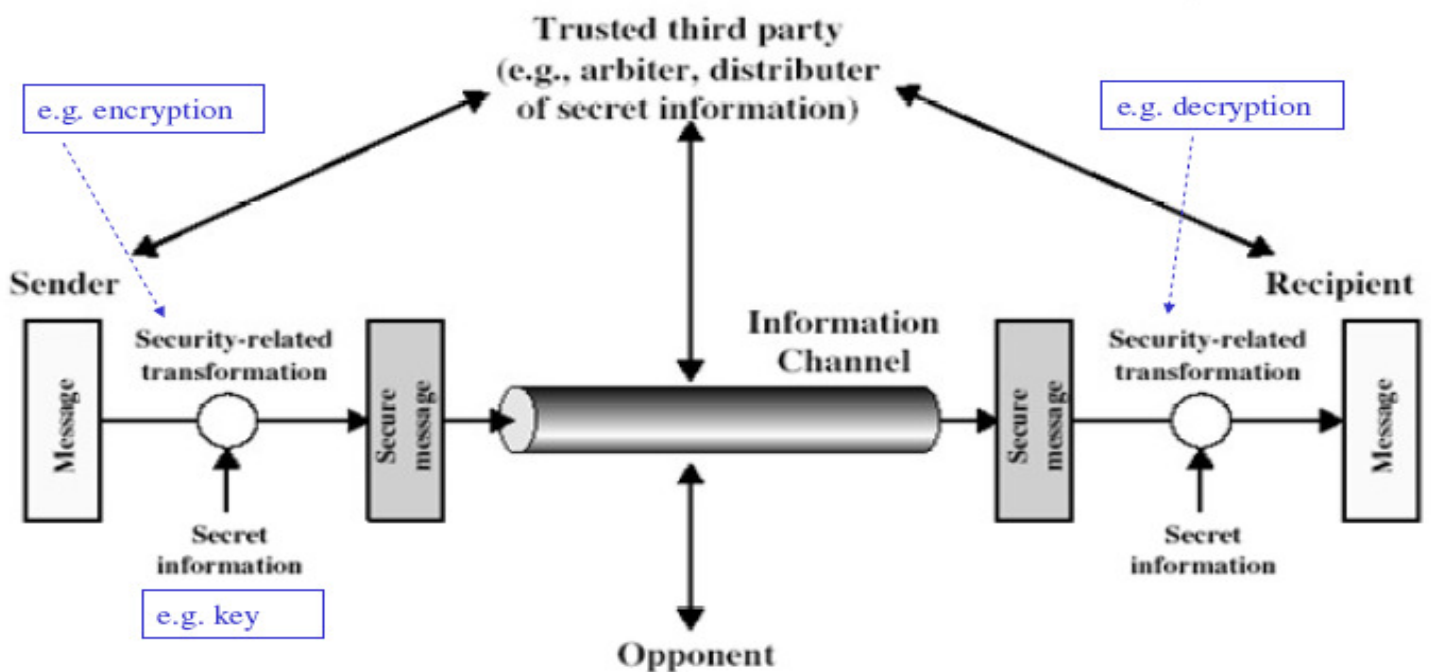
• שיטות הפצת מפתחות.

• ניפוח תנועה.

### כלים:

הערות	אבטחת רשת	אבטחת מערכת
Encryption	הצפנה	
	פרוטוקולי תאום	הגנות במערכת הפעלה
	נוהלי הפעלה	סיסמאות כניסה
	שיטות הפצת מפתחות	אנטי-ווירוס וכלים דומים
	ניפוח תנועה	הדסת תוכנה טובה

### מודל של אבטחת רשת



## סיכון ותקיפה

- תקיפה (attack) היא פעולה של יריב הפוגעת באחד מהיבטי האבטחה של מערכת מידע
- סיכון (threat) הוא כל דבר היוצר פוטנציאל תקיפה
- יש שני סוגי תקיפה:
  - תקיפה פעילה (active) - התוקף יכול לכתוב/למחוק בערוץ
  - תקיפה סבילה (passive) - התוקף יכול רק לקרא מהערוץ
- מטרת מערכת אבטחת מידע:
  - למנוע סיכונים של תקיפה
  - אם אי אפשר, לפחות לגלות תקיפות ולתעד אותן

---

## מטלות של מנהל אבטחת רשת

- לתכנן אלגוריתם של הצפנה / פענוח
- לארגן מערכת לייצור מפתחות סודיים למשתמשים
- להגדיר שיטות/נהלים להפצה בטוחה של מפתחות
- לקבוע כללים לשמירת המידע הסודי באתרים
- לקבוע נהלים לתכיפות החלפת המידע הסודי באתרים
- להגדיר פרוטוקולים להפעלת ההצפנה

---

### מושגים בהצפנה:

**Plaintext** - הודעה מקורית.

**Cipher text** - הודעה מוצפנת.

**Cipher** - חוק ההחלפה (ההצפנה).

**Key** - מפתח להצפנה.

**Encipher** - הצפנה.

**Decipher** - פענוח.

**אופן ההצפנה** -  $f(k, p) = c \Leftrightarrow p = f^{-1}(k, c)$

## שתי שיטות הצפנה:

### ■ (1) הצפנה סימטרית - Symmetric Encryption:

■ לשולח ולמקבל יש את אותו המפתח.

■ רוב האלגוריתמים הקלאסיים מבוססים (עד שנות ה-70) על הצפנה סימטרית.

■ השיטה עדיין נפוצה.

### ■ דרישות הצפנה סימטרית:

אלגוריתם הצפנה חזק.

מפתח הידוע רק לשולח ולמקבל.

סימנים מוסכמים:

מניחים שאלגוריתם ההצפנה ידוע.

ערוץ תקשורת מאובטח.

■ עפ"י שיטת ההצפנה הסימטרית, 2 הצדדים מחזיקים את אותו המפתח, וכך שניהם יכולים לשלוח

מסרים מוצפנים ולפתוח אותם.

■ החיסרון בשיטה זו הוא שאם המפתח מתגלה לצד שלישי, ניתן לפענח את המסרים.

### ■ (2) הצפנה במפתח פומבי - Public Key Encryption:

■ עפ"י שיטת ההצפנה הא-סימטרית (שעליה מבוסס ה-SSL), קיים מפתח ציבורי שנותן השירות מחלק

אותו לכל מי שמעוניין.

■ הוא מחזיק בנוסף מפתח שונה שמיועד לפענח את ההצפנות המבוצעות ע"י המפתח הפומבי שמחולק

לכולם.

■ ברגע שמשמש מרוחק מנסה לתקשר עם נותן השירות, הוא מקבל ממנו את המפתח הפומבי, מצפין

בעזרתו מפתח פרטי, שהוא ייצר בעצמו ורק הוא מכיר אותו, ואז שולח את המפתח הפרטי, כשהוא מוצפן

ע"י המפתח הציבורי, שאותו רק נותן השירות יודע לפענח.

■ ברגע שנותן השירות קיבל את ההצפנה, הוא יכול לפענח ולקבל את המפתח הפרטי, ואז ניתן לתקשר

בצורה סימטרית, כי ל-2 הצדדים יש את המפתח.

■ לדוגמא: לקוח מתחבר לאתר בנק. בעמוד ההתחברות הדפדפן שולח ברקע את המפתח הציבורי של

הבנק. הלקוח משיב לו את המפתח הפרטי שנוצר ע"י הדפדפן מוצפן בעזרת המפתח הציבורי, ואז ניתן

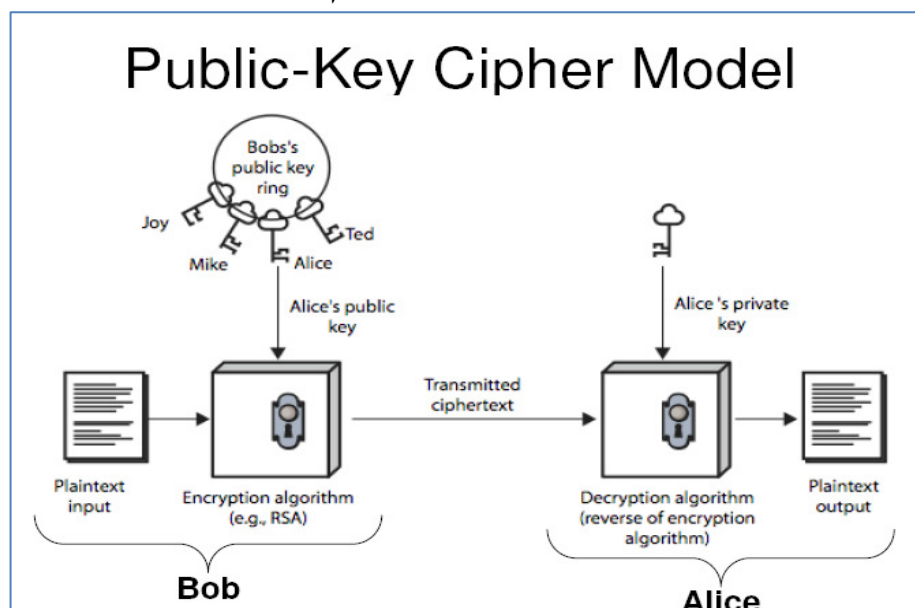
לבצע חיבור מאובטח כי לשני הצדדים יש את אותו המפתח אשר הלקוח ייצר. משלב זה כל ההתחברות

(המכילה שם משתמש וסיסמא) ולאחר מכן גם פעולות בנקאיות מוצפנות ונשלחת לבנק בבטחה.



דרישות ההצפנה :

1. אם המפתח הפומבי ידוע, לא ניתן להסיק את המפתח הפרטי.
2. המפתח הפרטי ידוע רק למשתמש עצמו.
3. לשולח חייב להיות המפתח הפומבי המתאים של הצד המקבל.



**אומנות ההצפנה - שיטות לבניית מפתח :**

- [Substitution](#)

- [Transposition](#)

- [Product](#)

כל אות נשאר באותו מקום בתוך המחרוזת רק מוצפנת.

כל אות נשאר אותו דבר רק במיקום אחר במחרוזת.

שילוב של Substitution ו Transposition , כלומר, כל אות נמצאת במקום

אחר במחרוזת וגם מוצפנת.

מפתח ח.ח.ע - הצפנה סימטרית.

שימוש במפתחות :

מפתח דו ערכי – הצפנה ע"י מפתח פומבי ( + מפתח פרטי).

Stream syfer – כל אות מוצפנת לחוד.

Block syfer – כל בלוק מוצפן בנפרד.

### **פענוח (ניתוח) cryptanalysis**

המטרה : פענוח מפתחות ולא רק הודעות.

Cryptanalysis – התקפה חכמה.

Brute force attack – להשתמש בכל אפשרויות המפתחות.

ניתן לסווג את ההתקפות לפי המתקין.

2 סוגי אבטחה :

1. אבטחה מוחלטת = לא ניתן לפרוץ את ההצפנה.

2. אבטחה מותנית = ניתן לפרוץ את הקוד כאשר המידע הופך ללא רלוונטי.

## צפנים קלאסיים

### צופן קיסר – Caesar cipher

הטקסט הגלוי = אותיות קטנות.  
הטקסט המוצפן = אותיות גדולות.

■ הרעיון הינו לקבוע מספר כלשהו וממנו הזזת כל אות (K) 3 מקומות.

■ לדוגמא האות M תהיה  $M + 3 = P$  משמע P.

■ Meet me after the toga party = PHHTPHDIWHUWKHWRJDSUWB

■ הרווח אינו לקח בחשבון (נזרק לפח), זאת על מנת להימנע מזיהוי תבניות וזיהוי קל של הצופן.

■ K יכול להיות בין 1-25 (26 יחזיר אותנו לטקסט המקורי).

■ זהו צופן יחסית קל (ניסוי של 25 אפשרויות ניסוי וטעייה). [נרודפורט]

$$C = E(p) = (p + k) \bmod 26 \quad \text{הצפנה}$$

$$p = D(c) = (c - k) \bmod 26 \quad \text{פענוח}$$

### צופן מונו אלפבית – mono alphabetic cipher

■ ערבוב כל האותיות וקביעה מחדש ע"י ייצוג של אות ע"י אות אחרת.

■ הצבה של אות תמורת אות בסדר אקראי.

■ המפתח הינו סדרה של 26 אותיות.

■ המגבלה בצופן זה הינה שהאותיות יהיו שונות ואף אות לא תחזור על עצמה.

Plain: abcdefghijklmnopqrstuvwxyz

Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN

Encipher:

Plaintext: ifwewishtoreplaceletters

Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA

■ מספר האפשרויות בצופן זה הינו **26!**

■ זהו צופן יחסית חלש.

■ צירופי האותיות באנגלית ושכיחותן של אותיות לעומת אותיות אחרות הופכות את קוד זה ללא

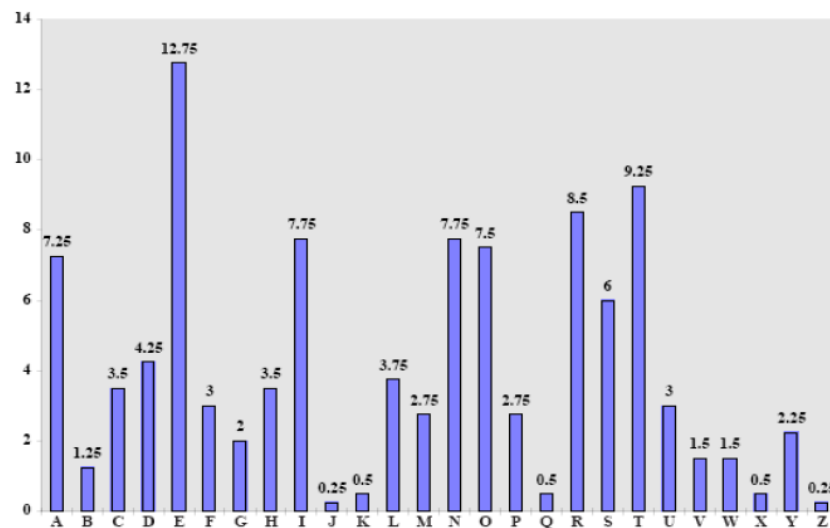
קשה במיוחד.

■ במפתח: אות שהוכנסה בצופן עצמו לא תספר פעמיים (נדלג עליה)

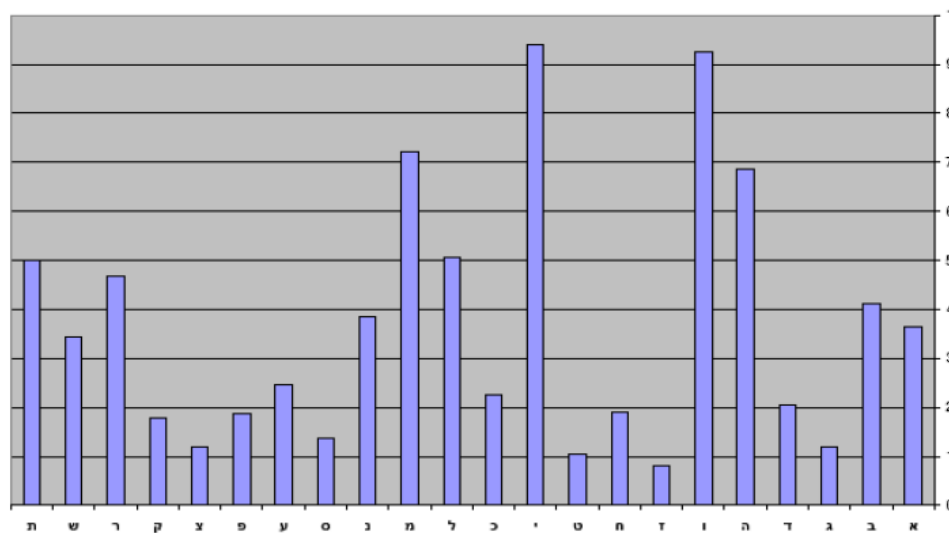
■ הסבר:

CARDIOVASCULAR .....

a b c d e f g h i j k l





כדי להקשות ניתן לתת לאות כלשהי מספר ערכים. (דוגמת האות E).  
**שכיחות אותיות עבריות בעתונות מדעית**







## צופן Play fair cipher

*eg*  *FI*  
shift right


*yw*  *GN*  
shift down

*ck*  *DE*  
cross

*fs*  *IP*  
cross

*dk*  *KT*  
shift down

✖ בחרים מילה כלשהי כמפתח ולאחריה משלימים את המטריצה בא"ב.



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

*ee* → *exe*

✖ אם אותה האות מופיעה פעמיים: *ee*, אפשרות זו מבוטלת ומוכנסת באמצע באות X:

✖ אם נשארת אות אחת בסוף – מוסיפים לה X.

✖ בהצפנה נעים ימינה ולמטה.

✖ בפענוח נעים שמאלה ומעלה (כיוונים הפוכים).

✖ דוגמא:

$$choose = \underbrace{ch}_{ch} \underbrace{ox}_{ox} \underbrace{os}_{os} e = \underbrace{ch}_{ch} \underbrace{ox}_{ox} \underbrace{os}_{os} \underbrace{ex}_{ex} = \underbrace{FI}_{ch} \underbrace{XW}_{ox} \underbrace{KR}_{os} \underbrace{WV}_{ex}$$

### PLAYFAIR Example

Given key phrase: NEW AMSTERDAM

a) Encode: "choose a new cipher" →

b) Decode: "VTSVB NWVTF RLHDA DRTGZ" →

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

**חסרון - שיטת פריצה:** באמצעים הקיימים היום, ניתן לבנות גם טבלאות שכיחויות לזוגות של אותיות – ולכן קל לפריצה.

## Poly alphabetic ciphers

● הרעיון הינו שימוש במספר אלפביתיים שמוחלפים תוך כדי ההצפנה.

אות מוצפנת  
מפתח

● מגדירים 26 אלפביתיים :

### Vigenere Table

	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

*key* = *DECEPTIVE*

*plain text* = *b a d l u c k a l l*

*b by D alphabet* = *E*

*a by E alphabet* = *E*

*d by C alphabet* = *F*

*l by C alphabet* = *P*

L אורך, לא ידוע בד"כ. בדוגמא שלנו L=9.

*GDAK* *GDAK*  
}  $t_2$

*t*: 36 48 30 131

נצפה ש- L יחלק בכמעט כולם -6.

● כאשר יש מקרה בו האותיות חוזרות על עצמן, נחפש GCD מחלק משותף מקסימאלי

ובאמצעותו נבין את אורך המפתח.

**VIGENERE קוד****שבירת VIGENERE:**

קוד זה לא נשבר למעלה מ 100 שנה עד מלחמת העולם השנייה, ע"י קצין בצבא הגרמני (קריסקי).

**כיצד נשבר:**

נניח כי אנו יודעים כי המפתח הוא בגודל 4 ואז נחלק את כל ה CHIPHER שלנו לרביעיות, מה שנמצא באותו טור מוצפן ע"י אותו אות ועפ"י טבלת שכיחויות נוכל למצוא את ה PLAINTEXT.

**כיצד נמצא את אורך המפתח? מהו ה K?**

נסתכל על ה CHIPERTEXT ונחפש צירופים שחוזרים על עצמם מספר רב של פעמים

(דוגמת the או ing):

KEY	C	O	A	L	C	O	A	L	C	O	A	L
Plaintext	t	h	e	t	h	e	...	...	t	h	e	...
Cipher text	V	V	E	T	S	G	...	...	V	V	E	..

סוג הצירוף עלול לחזור על עצמו עקב אילוצי שפה או סוג המקור של הטקסט.

נניח לדוגמא כי the קיים בקוד 20 פעמים - הסיכוי שיחזור על עצמו הוא 5 כלומר קיים יחס של 1 ל 4 שהצירוף The יהיה לדוגמא VVE. כעת נמדוד את כל המרחקים בין הצירופים הדומים ונכתוב אותם

מרחק	סוג
12	VVE
18	VVE
32	VVE
31	VVE
19	VVE

בטבלה:

כעת נבדוק מבין השורות בטבלה איזה מחלק משותף למספר הרב ביותר של השורות.

ובכך נדע מהו האורך של ה KEY.

$2 \times 3 \times 7$

$2 \times 3^2$

$2 \times 3 \times 5$

$2 \times 3 \times 11$

$2 \times 5^2$

$3^2 \times 5$

הפרשים	מיקום	דירוג
42,18	20,62,80	XDL
30,66	12,43,109	ANBGD
50	39,89	DBAS
45	60,105	GABN

6 מתאים ל 42,18,30,66 – 4 פעמים

5 מתאים ל 30,50,45 – 3 פעמים

10 מתאים ל 30,50 – פעמיים

9 מתאים ל 18,45 – פעמיים

\* מכון ש 6 מופיע הכי הרבה פעמים ננסה לשבור את הצופן עם  $K=6$  ואם לא נצליח ננסה עם  $K=5$ .

אותו אדם שפיצח את הצופן של VIGENERE המציא גם דבר הנקרא AUTOKEY.

**AUTOKEY.**

## AUTOKEY צופן מפתח אוטו

הוא נקבע עפ"י קטע מסוים מתוך ה- PLAINTEXT .

דוגמא: מילת key = coal

wearegoinginto = plain text

KEY	C	O	A	L	W	E	A	R	E	G	O	I
Plaintext	w	E	A	R	E	G	O	I	N	G	T	O
Cipher text	Y	S	A	C	A	.	.	.	.	.	.	.

בחירת מפתח באורך קבוע ואז ממשיכים עם ה-P

יתרון : המפתח משתנה מטקסט לטקסט – יותר קשה לפצח.

כיצד מפענחים : מתחילים כפענוח VIGENERE רגיל ולאחר שעוברים את אורך המפתח הרגיל מעתיקים

את ה- PLAINTEXT שפוענח אל ההמשך של ה-KEY.

קיימת שיטה אחת שהוכח באופן אבסולוטי שאי אפשר לשבור אותה :

## ONE TIME PAD : שיטה שלא ניתנת לפענוח

בשיטה הידנית :

אנו כותבים ספר בעל 100 תווים בכל עמוד אשר נכתבו בסדר מקרי.

ספר זה קיים בשני עותקים אצל כל צד. לאחר ש-PLAINTEXT מוצפן ע"י VIGENERE בעזרת אחד

מהדפים של הספר , אנו שורפים את הדף של הספר.

אין פה שום הסתברות ואי אפשר להניח כלום! מהטקסט.

בשיטה זו ה- confusion כה גבוה שאין צורך ב-diffusion.

בשיטה הממוחשבת :

אנו כותבים סדרה של ביטים (שהיא הטקסט) בשני עותקים באותה שיטה.

וההצפנה היא פשוט XOR בין המפתח לבין הטקסט

ופענוח היא פשוט XOR בין המפתח לבין ה-CHIPER.

$$p_i \oplus k_i = c_i$$

$$c_i \oplus k_i = (p_i \oplus k_i) \oplus k_i = p_i \oplus (k_i \oplus k_i) = p_i \oplus 0 = p_i$$

פה מסתיימות כל השיטות של ה- SUBSTITUTION.

## שיטת TRANSPOSITION:

אנו כותבים מפתח שהוא מספר אשר נקבע עפ"י חלוקה של הטקס לקבוצות של 4 (מספר עמודות) – לבסוף נשארו 3 מקומות ריקים אז הוספנו XYZ – SPACER.

Key:	<u>3 4 2 1 5 6 7</u>
Plaintext:	a t t a c k p o s t p o n e d u n t i l t w o a m x y z
Ciphertext:	TTNAAPTMTSUOAODWCOIXKNLYPETZ

את הפענוח של ה CHIPHER אנו עושים בצורה ההפוכה – אנו בונים טבלה עפ"י אורך ה CHIPHER ואורך ה KEY. ומתחילים להציב בטבלה עפ"י סדר מספרי של המפתח.

**שיטה טובה יותר:** כתיבה של ה- KEY בצורה של מילה והסדר הוא עפ"י האלפבית. ובנוסף לא משלימים ב XYZ והפענוח מבוצע ע"י אורך הטקסט המוצפן והגודל של המפתח (בונים טבלה ומשאירים חלק ריק

Word key:	<b>M U S T A R D</b>	עפ"י שארית).
Key:	<u>3 4 6 7 1 5 2</u>	
Plaintext:	a t t a c k p o s t p o n e d u n t i l t w o a m	

יתרון הצופן הפוליאלפביתי

1. אותה אות יכולה להיות מוצפנת בכמה צורות לפי מיקומה הסידורי בטקסט: דבר זה גורם

לטשטוש סוגיית השכיחויות

## חבורה GROUP

חבורה  $G$  היא מבנה אלגברי בסיסי הכולל קבוצה עם פעולה בינארית, אשר מקיימת את התכונות הבאות:

- סגירות: לכל  $a, b \in G$  מתקיים  $a \cdot b \in G$ .
  - אסוציאטיביות (קיבוציות): לכל  $a, b, c \in G$  מתקיים  $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ .
  - איבר יחידה (נייטרלי): קיים איבר  $e \in G$  כך שלכל  $a \in G$  מתקיים  $a \cdot e = e \cdot a = a$ .
  - הפיכות: לכל  $a \in G$  קיים  $b \in G$  כך ש  $a \cdot b = b \cdot a = e$ .
- חבורה אבליה (חילופית) היא חבורה שבה מתקיים, בנוסף, תנאי הקומוטטיביות (חילופיות)  $a \cdot b = b \cdot a$ .

$$(G, +), (G, *)$$

$$(a+b)+c = a+(b+c),$$

$$0+a = a,$$

$$(-a)+a = 0$$

שלמים עם כפל אינם חבורה.

$$\cancel{(Z, *)}$$

$$ax = 1 \text{ הופכי} \quad \text{ל-0 אין הופכי לעולם.}$$

קבוצת הטבעיים, ממשים, רציונאליים הם חבורה לחיבור.

$$\cancel{(Q, *)} \longrightarrow (Q \setminus \{0\}, *)$$

$$(\{i, -i, 1, -1\}, *)$$

חבורה קומוטטיבית / חלופית אם כלל החילוף מתקיים בהם (+)  
כנ"ל לגבי כפל למעט קבוצת מטריצות - כפל מטריצות אינו קומוטטיבי.

חבורה היא ציקלית אם יש איבר יוצר / גנראטור כך שהאיבר מתקבל ע"י העלאה בחזקות של הגנראטור:

$$G = \{a, a^2, \dots, a^n\}$$

$$\{i, i^2, i^3, i^4\} = \{i, -1, -i, 1\}$$

יוצר:

(1) החזקות של מספר יוצר בשדה, נותנות את כל האיברים בשדה למעט 0.

(2) החזקה הכי נמוכה שנותנת את המספר 1, היא n-1.

דוגמא: נחפש את המספרים היוצרים של השדה 11:

1: לא יוצר / 1 אף פעם לא יוצר

הערה:  $\square_{11}$  הוא שדה כיוון ש-11 הוא ראשוני.

$$\begin{bmatrix} 2 & 4 & 8 & 5 & 10 & 9 & 7 & 3 & 6 & 1 \end{bmatrix}$$

$$2: 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8, 2^9, 2^{10} \quad \text{יוצר}$$

$$\begin{bmatrix} 3 & 9 & 5 & 4 & 1 & 3 & 9 & 5 & 4 & 1 \end{bmatrix}$$

$$3: 3^1, 3^2, 3^3, 3^4, 3^5, 3^6, 3^7, 3^8, 3^9, 3^{10} \quad \text{לא יוצר / חסרות ספרות רבות (ציקלי 39541)}$$



## חוג RING

חבורה = קבוצה עם פעולה אחת.  
חוג הוא קבוצה עם 2 פעולות.

חוג הוא חבורה לגבי חיבור.  
לגבי כפל בתנאי ש: סגירות  
אסוציאטיבי(כלל קיבוצי,  
ניתן לפתוח סוגריים.

תחום שלמות הוא חוג חילופי עם יחידה כפליית שאין בו מחלקי אפס.

תחום שלמות =  
1. אין מחלקי 0.  $ab = 0 \rightarrow a = 0$  או  $b = 0$   
כללים  
שקולים  $ab = ac, a \neq 0 \rightarrow b = c$  2. כלל הצמצום:

## שדה FIELD

שדה: 2 דרישות: 1. חוג שבו הכפל הוא חילופי.  
2. יש בו איבר יחידה. לכל  $a \neq 0$  יש הופכי.  $a^{-1}$

שדה הוא מבנה אלגברי הכולל קבוצה עם שתי פעולות בינאריות, להן אפשר לקרוא "חיבור" ו"כפל", ושני קבועים (שונים) - 0 ו-1.

$$ab = 0$$

שדה הוא תחום שלמות

$$(1) a = 0$$

$$(2) a \neq 0 \rightarrow \text{there is } a^{-1} \rightarrow a^{-1}0 = 0 \rightarrow a^{-1}ab = 1b = b$$

## האופרטור מודולו %

$$a \bmod n = \text{השארית של } a \text{ בחלוק } n$$

$$19 \bmod 7 = 5$$

שארית לעולם תהיה חיובית

$$(a + b)(\text{mod } n) = [a(\text{mod } n) + b(\text{mod } n)](\text{mod } n)$$

Modulo 8 Addition Table      Modulo 9 Multiplication Table

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

x	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8
2	0	2	4	6	8	1	3	5	7
3	0	3	6	0	3	6	0	3	6
4	0	4	8	3	7	2	6	1	5
5	0	5	1	6	2	7	3	8	4
6	0	6	3	0	6	3	0	6	3
7	0	7	5	3	1	8	6	4	2
8	0	8	7	6	5	4	3	2	1

מודולו 9 הם חוג, אך אינם חוג שלמות.

$$3 * 6 \text{ mod } 9 = 0$$

וזאת כיוון שהתוצאה שווה ל-0 ו-3 ו-6 אינם אפסים

תרגיל כיתה:

למצוא יוצר עבור החבורה  $(\mathbb{Z}_8, +)$

יוצר	לא יוצר
$1 \rightarrow \{1, 2, 3, 4, 5, 6, 7, 0\}$	2
$3 \rightarrow \{3, 6, 1, 4, 7, 2, 5, 0\}$	4
$5 \rightarrow \{3, 6, 1, 4, 7, 2, 5, 0\}$	6
$7 \rightarrow \{3, 6, 1, 4, 7, 2, 5, 0\}$	8

$$a \equiv b \text{ mod } n \Leftrightarrow a \text{ mod } n = b \text{ mod } n$$

$$\Leftrightarrow (a - b) \text{ mod } n = 0$$

$$\Leftrightarrow (a - b) = n * t$$

$$[5 \equiv 15 \equiv 55] :: \text{mod } 10$$

יחס שקילות

## מחלקים

B מחלק את A. A שווה כפולה של B

דוגמא: 5 מחלק את 40  $5 \leq 40$  כפול מספר שלם  $40=5t$

$$5 \mid 40 \quad 40 - 5t = 5 * 8$$

מחלקי 36: 1, 2, 3, 4, 6, 9, 12, 18, 36. סה"כ 9 מחלקים

$$\{s^R 3^m\}$$

$$0 \leq k \leq 2$$

$$0 \leq m \leq 2$$

$$\gcd\{98, 21\}$$

$$21:3=7$$

$$7:7=1$$

$$1$$

$$98:2=49$$

$$49:7=7$$

$$7 \cdot 7$$

GCD

$$2^0 * 3 * 7$$

$$2 * 3^0 * 7^2$$

## הגדרה 1.2

קבוצה  $H$  עם פעולת חיבור  $a+b$  ועם פעולת כפל  $a \times b = ab$  נקראת חוג חילופי (Commutative Ring) אם מתקיימים עבורה הכללים הבאים:

$$a + b = b + a \quad (1)$$

$$(a + b) + c = a + (b + c) \quad (2)$$

$$0 \in H \text{ קיים } 0 \text{ המקיים: } a + 0 = a \text{ לכל } a \quad (3)$$

$$\text{לכל } a \in H \text{ קיים } -a \in H \text{ שעבורו } a + (-a) = 0 \quad (4)$$

$$a \times b = b \times a \quad (5)$$

$$(a \times b) \times c = a \times (b \times c) \quad (6)$$

$$1 \in H \text{ קיים } 1 \text{ המקיים: } 1 \times a = a \text{ לכל } a \quad (7)$$

$$a \times (b + c) = (a \times b) + (a \times c) \quad (8)$$

## משפט 1.1

הקבוצה  $Z_n$  עם הפעולות  $\oplus$  ו-  $\otimes$  היא חוג חילופי (כלומר מקיימת את הכללים (1) עד (8))

## הגדרה 1.3

חוג חילופי  $H$  נקרא שדה (Field) אם בנוסף לכללים (1) – (8) של חוג חילופי מתקיים גם:

$$\text{לכל איבר } a \in H \text{ יש איבר הופכי } a^{-1} \in H \quad (9)$$

$$\gcd(98, 21) \rightarrow 98 \bmod 21 = 14 \rightarrow \gcd(21, 14) \rightarrow 21 \bmod 14 = 7 = \gcd(14, 7) \rightarrow 14 \bmod \underset{\text{gcd}}{7} = 0$$

מספרים זרים :  
מספרים A B זרים אם אין להם גורם משותף פרט ל-1.

איבר הפיך ב-  $Z_N$  אם"ס הוא זר.

ב-  $18Z$  המספרים הזרים הם אלו שלא מתחלקים ב- 2 ולא מתחלקים ב- 3.

אם כל המספרים הפיכים ב-  $Z_N$  אז  $Z_N$  הוא שדה

נמצא יוצר עבור החבורה (generator)  $(\mathbb{Z}_8, +)$

יש את כל המספרים מ-1 עד 8-1 יוצרים בחבורה הנ"ל

$$\left[ \begin{array}{l} 1 \Rightarrow \{1, 2, 3, 4, 5, 6, 7\} \\ 3 \Rightarrow \{3, 6, 1, 4, 7, 2, 5\} \\ 5 \Rightarrow \{5, 2, 7, 4, 1, 6, 3\} \\ 7 \Rightarrow \{7, 6, 5, 4, 3, 2, 1\} \end{array} \right]$$

אין את כל המספרים מ-1 עד 8-1 אינם יוצרים בחבורה הנ"ל

$$\left[ \begin{array}{l} 0 \Rightarrow \{0\} \\ 2 \Rightarrow \{2, 4, 6, 8\} \\ 4 \Rightarrow \{4, 8\} \\ 6 \Rightarrow \{6, 4, 2, 8\} \end{array} \right]$$

היום יום ה'  $4dec$

איזה יום בשבוע הוא  $15feb$ ?

כמה ימים יעברו מעכשיו עד לפברואר???

דצמב' מסתיים ב- 31 דצמבר

ה-1 לינואר = 32 לדצמבר.

ה-2 לינואר = 33 לדצמבר.

ה-31 לינואר = 62 לדצמבר.

$$15/2 = 77/12$$

$$77/12 - 4/12 = 73days$$

$$73 \bmod 7 = 3$$

יום חמישי + 3 = יום ראשון.

מסקנה – התאריך ה- 15 לפברואר יהיה היום ראשון של השבוע

### אלגוריתם אוקלידס:

מוצא את ה- GCD של 2 מספרים נתונים.

כיצד מחשבים את ה- GCD – המחלק המשותף הגדול ביותר.

assume:  $a > b$

$$\gcd(a, b)$$

$1R =$  שארית

$$a = t * b + r_1$$

$\Downarrow$

$$r_1 = a \bmod b$$

$$d \mid a \text{ and } d \mid b \Rightarrow d \mid r_1$$

$$r_1 = a - tb$$

exm:

$$a = 25$$

$$b = 15$$

$$25 \bmod 15 = 10$$

$$\gcd(15, 25) = 5$$

$$\gcd(15, 10) = 5$$

rule:

$$d \mid r, d \mid b \Leftrightarrow d \mid a, d \mid b$$

מחליף את המספר הקטן בגדול ואת הקטן בשארית  
 $ex\_2:$

$$\gcd(81, 72) = ?$$

$$81 \bmod 72 = 9$$

$$\gcd(72, 9)$$

$$72 \bmod 9 = 0$$

$$\gcd(9, 0) = 0$$

התהליך רקורסיבי עד שמגיעים ל-0

דוגמא 4-18

לפי ה-GCD, ניתן לבדוק אם המספרים זרים או לא?

שני מספרים זרים אם אין מחלק משותף למעט 1.

אם בסיומו של התהליך נגיע ל-1 – המספרים זרים. אם לא הם יש להם מחלק משותף = לא זרים.

$$\gcd(n, a) = 1 \quad a \in \mathbb{Z}_n \quad \text{הפיך } a$$

26 = 2 \* 13 זרים – אסור שיתחלקו בפירוק = 2 ו-13 וכל מה שפריק מהם.

כל האי זוגיים למעט 13.

$$a \in \mathbb{Z}_{26} \text{ הפיך} \Leftrightarrow \gcd(26, a) = 1 \Leftrightarrow a \text{ לא מתחלק ב-2 ו-13}$$

כיצד מוצאים מספר הפוך של מספר ב- $\mathbb{Z}_n$

נתון  $a \in \mathbb{Z}_n$

מצא את  $a^{-1}$  (פתרון של  $ax = 1$ )

$$\overbrace{26}^n = 3 * 7 + 5$$

$$7 = 1 * 5 + 2$$

$$5 = 2 * 2 + 1$$

$$5 = 26 - 3 * 7$$

$$2 = 7 - 1 * 5$$

$$7 - 1(26 - 3 * 7) =$$

$$4 * 7 - 26 * 1$$

$$1 = 5 - 2 * 2$$

$$26 - 3 * 7 - 2(4 * 7 - 1 * 26) =$$

$$26 - 3 * 7 - 8 * 7 + 2 * 26 =$$

$$3 * 26 - 11 * 7$$

$$1 = 3 * 26 - 11 * 7$$

$$1 \bmod 26 = (3 * 26 - 11 * 7) \bmod 26$$

$$1 \bmod 26 = (3 * 26) \bmod 26 - (11 * 7) \bmod 26$$

$$1 \bmod 26 = (-11 * 7) \bmod 26$$

$$7^{-1} = -11 = 15$$

15 הוא ההופכי של 7 ב- $\mathbb{Z}_{26}$

20

דוגמא 4-20 1759

אם היינו מקבלים תשובה חיובית אז היא הייתה התשובה ללא משלים ל-26



$$\overbrace{9}^n = 4 * 2 + 1 \quad 1 = 26 - 3 * 7 \quad \gcd(26, 7) = \gcd(7, 5)$$

$$\begin{aligned} 7 &= 1 * 5 + 2 & 2 &= 7 - 1 * 5 = \\ & & 7 - 1(26 - 3 * 7) &= \\ & & 4 * 7 - 26 * 1 &= \gcd(5, 2) \end{aligned}$$

$$\begin{aligned} 5 &= 2 * 2 + 1 & 1 &= 5 - 2 * 2 = \\ & & 26 - 3 * 7 - 2(4 * 7 - 1 * 26) &= \\ & & 26 - 3 * 7 - 8 * 7 + 2 * 26 &= \\ & & 3 * 26 - 11 * 7 & \end{aligned}$$

אלג אוקלידס מורחב:

$$\begin{aligned} 1 &= 3 * 26 - 11 * 7 \\ 1 \bmod 26 &= (3 * 26 - 11 * 7) \bmod 26 \\ 1 \bmod 26 &= (3 * 26) \bmod 26 - (11 * 7) \bmod 26 \\ 1 \bmod 26 &= (11 * 7) \bmod 26 \\ 7^{-1} &= -11 = 15 \end{aligned}$$

$$\gcd(n, a) = 1 \Leftrightarrow \text{הפיך } a \in \mathbb{Z}_n$$

$$\mathbb{Z}_n \text{ שדה } \Leftrightarrow \text{כל } a \in \mathbb{Z}_n, a \neq 0 \text{ הפיך}$$

$$\Leftrightarrow \text{כל } 1 \leq a \leq n-1 \text{ הוא זר ל- } n$$

$$n \Leftrightarrow \text{ראשוני}$$

$$\mathbb{Z}_n \Leftrightarrow n \text{ ראשוני}$$

## דוגמא:

מצא את האיברים ההפיכים ב-  $\mathbb{Z}_{28}$

פתרון:

הפירוק לגורמים של 28 הוא:  $28 = 2^2 \times 7^1$ .

המספרים הזרים לו הם המספרים שאינם כוללים בפירוק שלהם את 2 ואת 7, כלומר:  
1, 3, 5, 9, 11, 13, 15, 17, 19, 23, 25, 27.

$$GF(p) = \mathbb{F}_p$$

$GF(p^n) = \mathbb{F}_{p^n}$  מס' ראשוני P.  
 $Z$  הוא שדה של פולינומים ממעלה  $n-1$

$$GF(p^2) = \{ax + b; a, b \in \mathbb{F}_p\}$$

$$GF(p^3) = \{ax^2 + bx + c; a, b, c \in \mathbb{F}_p\}$$

$$GF(8) = GF(2^3) \xrightarrow{N-1} \{ax^2 + bx + c; a, b, c \in \mathbb{F}_2\}$$

$$GF(2) = \mathbb{F}_2 \quad 0, 1 \in \mathbb{F}_2$$

## צופן Hill

$\mathbb{Z}_{26}$

$$A_{n \times n} \in \mathbb{Z}_{26}^{n \times n}$$

$$P = P_1 P_2 P_3 \dots$$

$$P_j = n \text{ letters}$$

$$C_j = AP_j$$

$$AA^{-1} = I$$

המטריצה A חייבת להיות הפיכה ב-  $\mathbb{Z}_{26}$

בדיקה ע"י דטרמיננט שונה מ-0

כלל: A הפיכה  $\Leftrightarrow \det A \neq 0$  שונה מאפס

$\mathbb{Z}_{26}$

דוגמא:

$$A = \begin{pmatrix} 3 & 5 & 1 \\ 4 & 10 & 1 \\ 2 & 3 & 1 \end{pmatrix}, A^{-1} = \begin{pmatrix} 11 & 8 & 7 \\ 8 & 9 & 9 \\ 8 & 9 & 12 \end{pmatrix}$$

$P = \text{arrange}$

$$p_1 = \begin{pmatrix} a \\ r \\ r \end{pmatrix} = \begin{pmatrix} 0 \\ 17 \\ 17 \end{pmatrix}$$

$$C_1 = AP_1 = \begin{pmatrix} 3 & 5 & 1 \\ 4 & 10 & 1 \\ 2 & 3 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 17 \\ 17 \end{pmatrix} = \begin{pmatrix} 102 \\ 187 \\ 68 \end{pmatrix} = \begin{pmatrix} 102 \\ 187 \\ 68 \end{pmatrix} \bmod 26 = \begin{pmatrix} 24 \\ 5 \\ 16 \end{pmatrix}$$

$$p_2 = \begin{pmatrix} a \\ n \\ g \end{pmatrix} = \begin{pmatrix} 0 \\ 13 \\ 6 \end{pmatrix}$$

$$p_3 = \begin{pmatrix} e \\ x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 17 \\ 17 \end{pmatrix}$$

**חסרונות – שיטות פריצה:** אם נתונה מטריצת הצפנה, ניתן למצוא את מטריצת הפענוח בקלות. ואם אנו משיגים גם `plaintext` (טקסט), אזי ניתן למצוא את המטריצות ע"י מערכת משוואות ליניאריות. מכאן אנו למדים שמערכות ליניאריות אינן טובות להצפנה.

- צופן hill שהמטריצה שלו מסדר גבוה (n), מספק diffusion רב, כיוון שכל אות מושפעת מ-n אותיות.
- צופן hill אינו מספק confusion בדרגה גבוהה, כי הוא ניתן לתקיפה והוא למעשה צופן ליניארי.

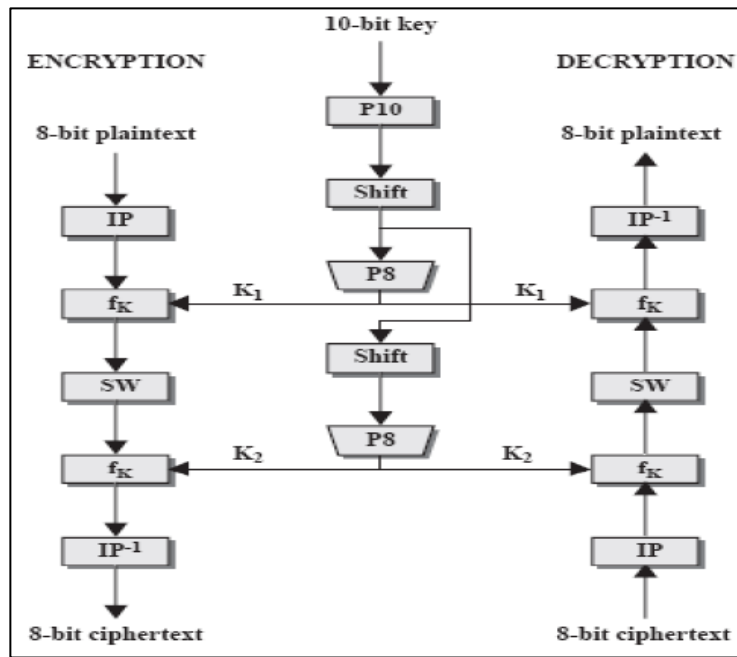
שיטת תקיפה של צופן Hill כאשר התוקף מחזיק n בלוקים מוצפנים ואת המסר המקורי של כל אחד:

1. מסדרים את המסרים המקוריים הידועים כעמודות במטריצה ריבועית B
2. מסדרים את המסרים המוצפנים כעמודות במטריצה ריבועית C
3. הופכים את המטריצה B בחשבון לפי  $\bmod 26$ .
4. אנו יודעים כי:  $AB=C$  לפיכך מקבלים:  $A = CB^{-1}$ .

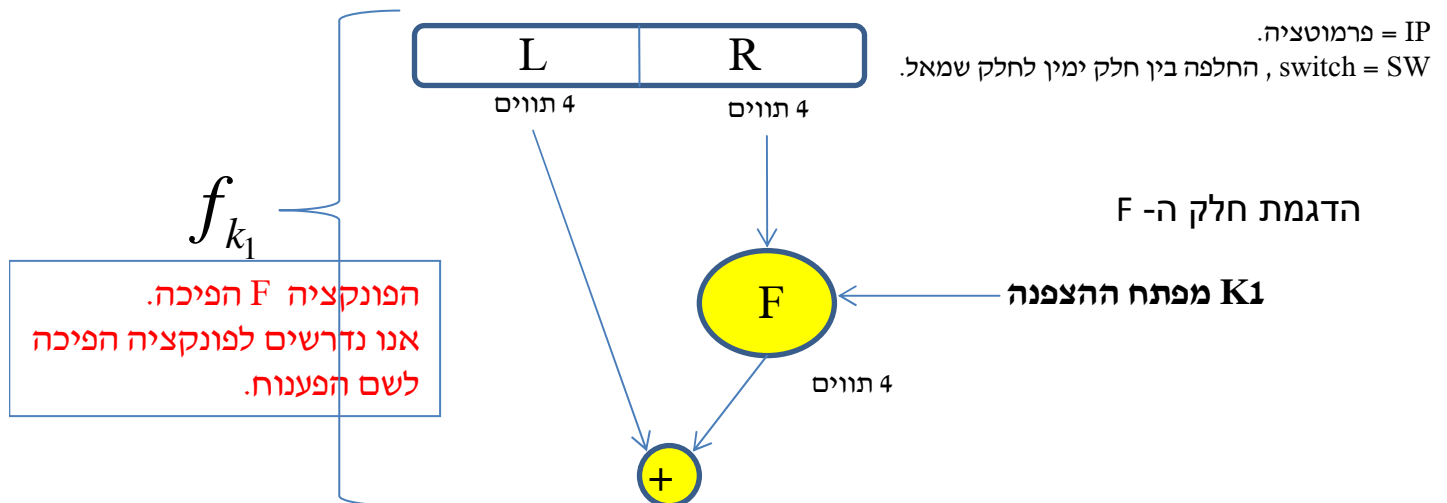
## S-Des – אלגוריתם הצפנה.

צופן DES = מילה של 64 ביט עם מפתח של 56 ביט. על מנת להבין DES נלמד קודם את S DES מדובר בצופן סימטרי

### תרשים כללי:



$$IP^{-1} \circ f_{k_2} \circ SW \circ f_{k_1} \circ IP [8bit]$$



$$f_{k_1}(L, R) = (L \oplus F(R, k_1), R)$$

$$f_k \circ f_k(L, R) = f_k(L \oplus F(R, k), R) = (L \oplus F(R, K)) \oplus F((R, k), R)$$

פעולת ה-XOR אסוציאטיבית ולכן ניתן לשחק עם הסוגריים.

$$f_k \circ f_k(L, R) = f_k(L \oplus F(R, k), R) = (L \oplus \underbrace{F(R, K)}_0) \oplus F((R, k), R) = (L, R)$$

הרכבה של F עם עצמו נותנת את פעולת הזהות:  $f = f_1^{-1}$

## היפוך לפונקציה

הצפנה

$$(IP^{-1} \circ f_{k_2} \circ SW \circ f_{k_1} \circ IP)^{-1}$$

פענוח

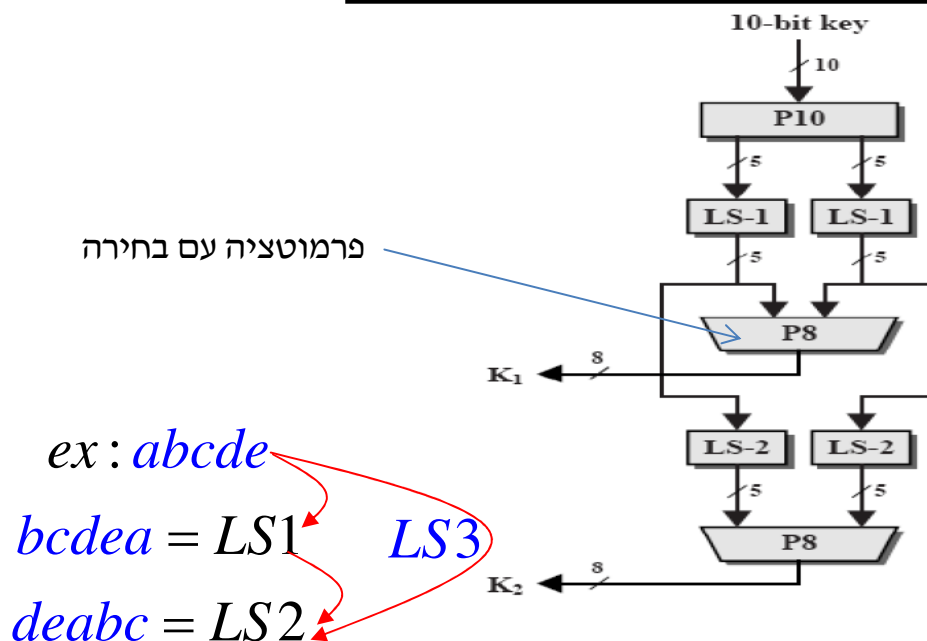
$$IP^{-1} \circ f_{k_1}^{-1} \circ SW^{-1} \circ f_{k_2}^{-1} \circ IP$$

הפעולות בפונקציה ההפוכה עובדות בסדר הפוך עם אותם מפתחות.

## חלק א' – הכנת מפתחות $(k_1, k_2)$

נניח כי נתון סדרה באורך 10 ביט.

## איור 2: S-DES - יצירת תת-מפתחות



## S-DES פירוט יצירת תת-מפתחות (1)

- הפרמוטציה P10 מוגדרת:

$$P10(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) = (k_3, k_5, k_2, k_7, k_4, k_{10}, k_1, k_9, k_8, k_6)$$

- דבר זה נרשם באופן מקוצר כך:

$abcde f g h i j$

P10

$ce b g \dots$

P10									
3	5	2	7	4	10	1	9	8	6

- לדוגמה: אם המפתח המקורי  $K$  הוא

1010000010

- אז:

$$P10(K) = 1000001100$$

## S-DES פירוט יצירת תת-מפתחות (2)

- $$p10: \underbrace{100000}_L \underbrace{01100}_R$$

$$LS1: \underbrace{000011}_{L} \underbrace{1000}_{R}$$

$P8 : 10100100 \Rightarrow k1$

$LS2: \underbrace{00100}_L \underbrace{00011}_R$

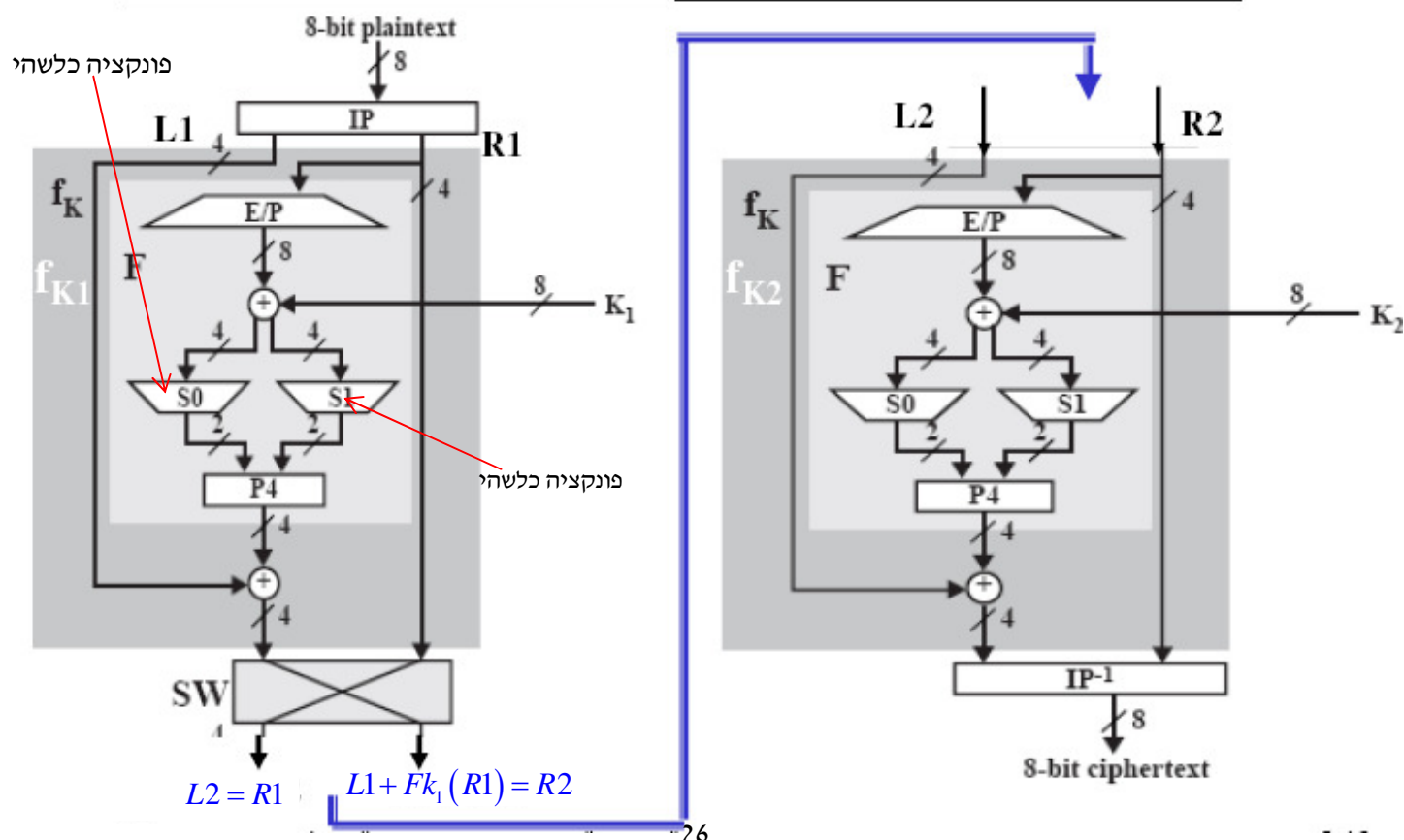
$P8: \underbrace{01000011}_{\text{data}} \Rightarrow k2$

P8							
6	3	7	4	8	5	10	9

- בדוגמה שלנו LS1 יתן : 00001 11000
- P8 יבחר עתה את המפתח:  $K_1=10100100$
- LS2 מופעל על תוצאת LS1 : 00100 00011
- ואז הכלל P8 יבחר את המפתח:  $K_2=01000011$

## פירוט ההצפנה

### איור 3: S-DES : ההצפנה במבט כללי





**הצפנת S-DES: הפרמוטציה IP**

- הפרמוטציה IP שמופעלת תחילה על המסר היא:

IP							
2	6	3	1	4	8	5	7

- הפרמוטציה הסופית  $IP^{-1}$  היא ההיפוך שלה:

$IP^{-1}$							
4	1	3	5	7	2	8	6

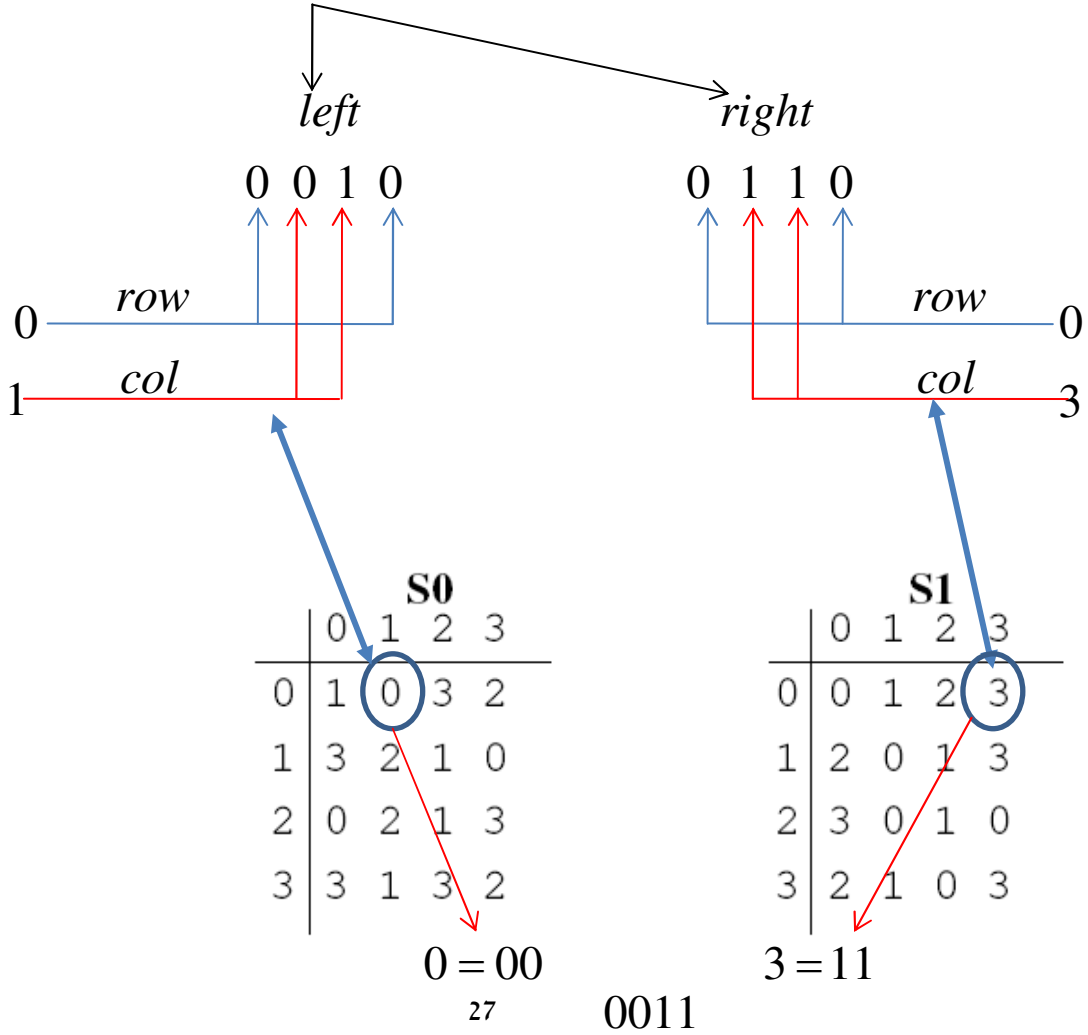
- דוגמה: מוצאו את  $IP(10100110)$ , הפעילו עליו את  $IP^{-1}$  ובדקו שמתקבלים בחזרה את 10100110

*plain text* : 10100110  $IP = 26314857$

$IP$  : 01110001  $E / P = 41232341$   
*L-rest R-work*

$E / P$  : 10000010  $k_1(\text{done before}) = 10100100$   
*from R from R*

$\oplus = (E / P) \oplus K_1 : \underline{00100110}$



## הצפנת S-DES: פונקציית F – שלב 4

- על תוצאה של פונקציית F מפעילים עוד פרמוטציה P4

P4			
2	4	3	1

*plaintext:*

10100110

$IP = 26314857$

$IP:$

01110001  
*L-rest R-work*

$E/P = 41232341$

$E/P:$

10000010  
*from R from R*

$k_1(\text{done before}) = 10100100$

$\oplus = (E/P) \oplus K_1 : 00100110$

$s_0, s_1 \Rightarrow 0011$

$p_4 = 2431$

$p_4 : 0110$

$p_4 \oplus L$

$p_4 \oplus L : 0110 \oplus 0111 = 0001$

0001      0001  
*R-work*

*sw*

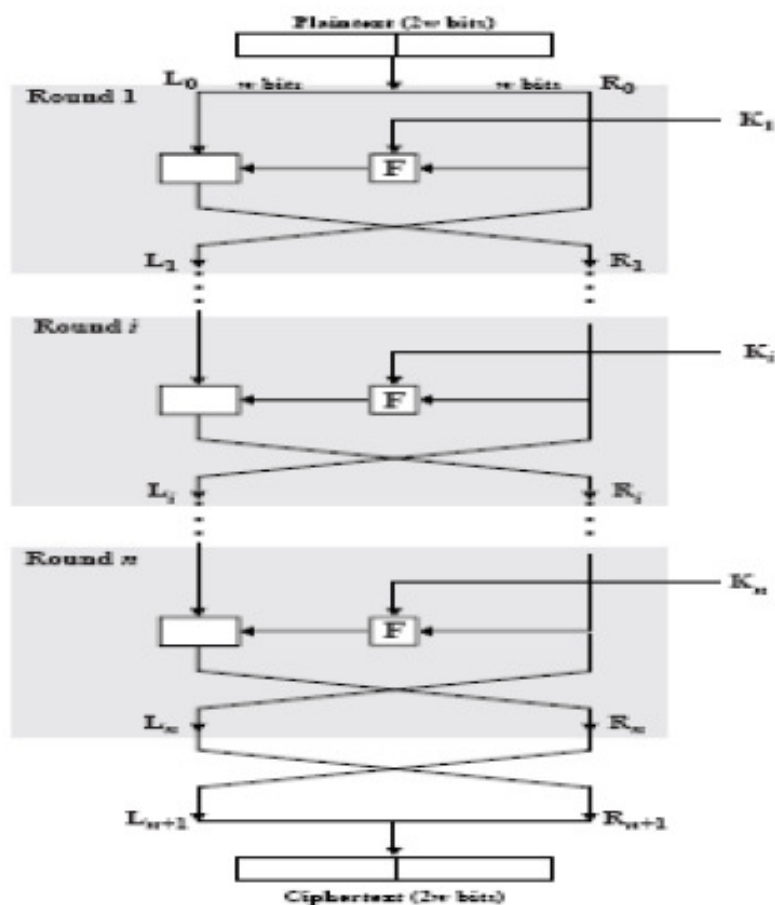
0001  
*R-work*

0001

00010001

להמשיך את כל החלק השני...

## איור 4: עקרון Feistel לצופן בלוק

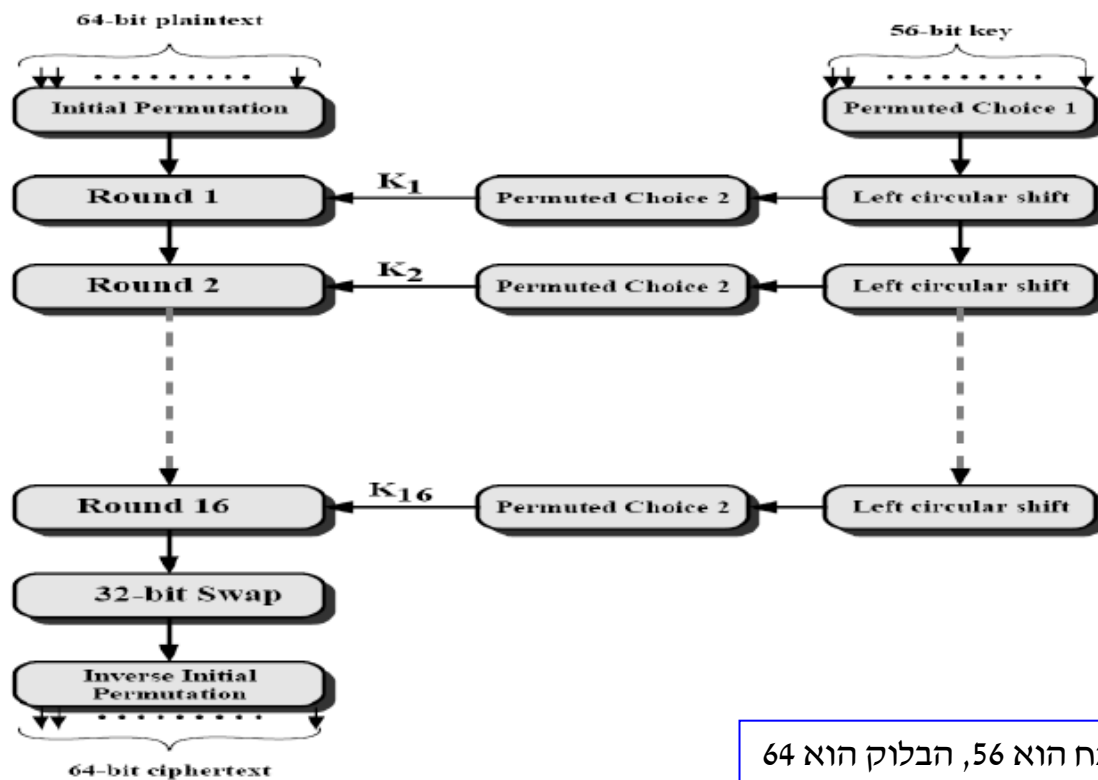


### DES – סיכום כללי:

1. ערבובים גלובליים של כל בלוק.
2. פונקציות ההצבה (S) לא ליניאריות.
3. פונקציות ההצבה מופעלות על קבוצות קטנות של ביטים
4. הערבובים וההצבות מבוצעים לסירוגין ב-16 מחזורים זהים.
5. הערבובים מפזרים בהדרגה את השפעת כל ביט על פני הבלוק = diffusion.
6. פונקציות ההצבה לא ליניאריות ומקשות את הניתוח המתמטי של הצופן = confusion.
7. כדי לגלות את המפתח יש צורך במשוואות ממעלה גבוהה עם משתנים רבים.
8. עבור צופן DES הפרמוטציות מספקות את ה-diffusion וקופסאות ה-S את ה-confusion.

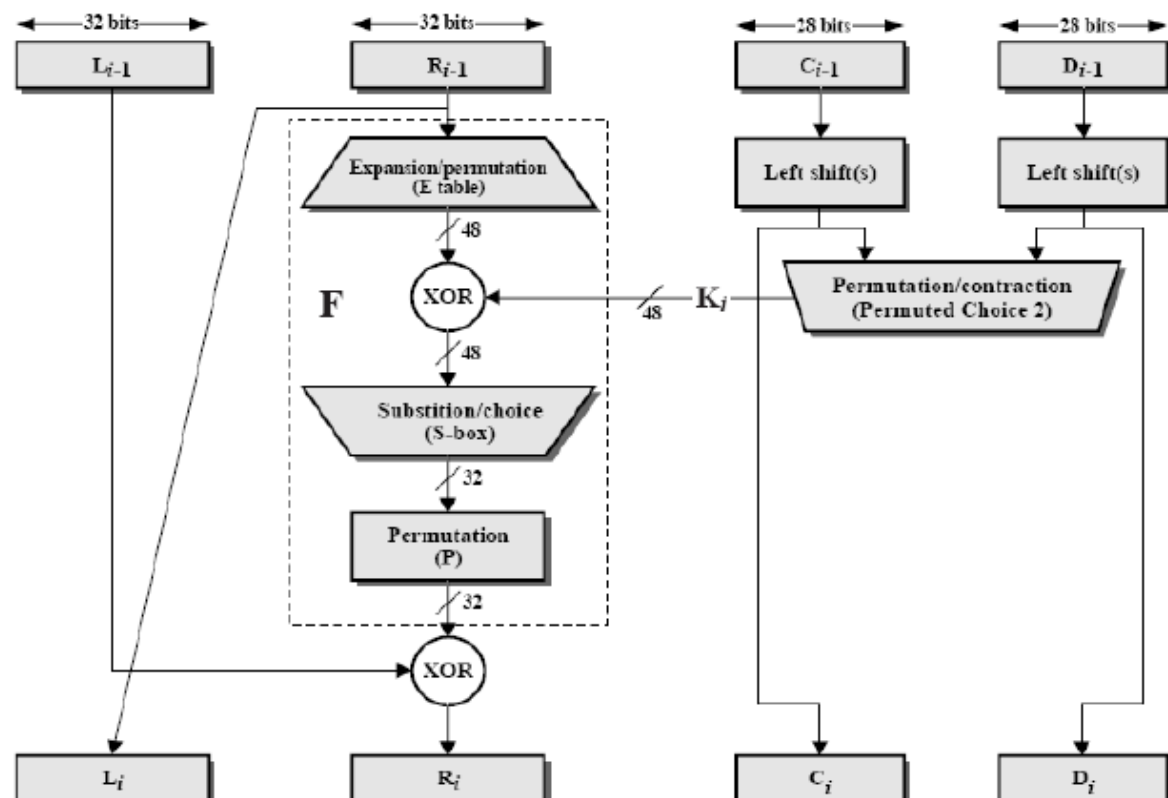
## DES

### איור 6: DES - מבנה כללי

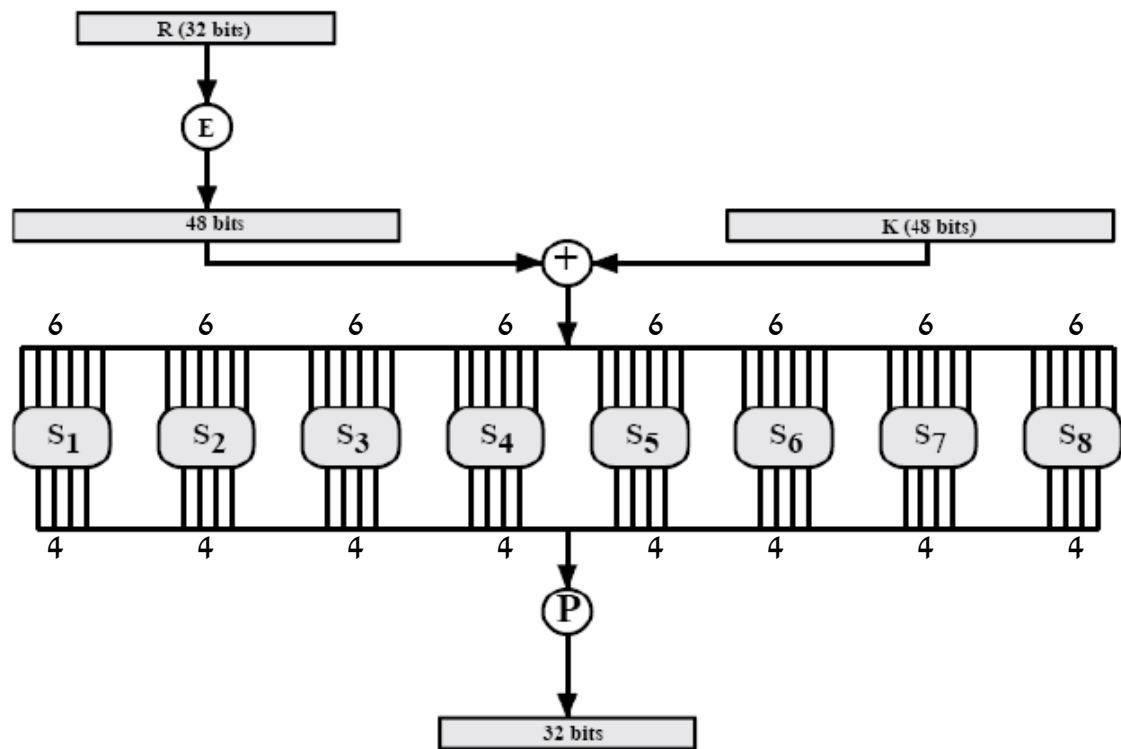


המפתח הוא 56, הבלוק הוא 64

### איור 7: DES - סבב אחד מתוך 16



# איור 8: DES - חישוב $F(R,K)$



101110

$row = 10 = 2$   
 $col = 011 = 7$

101110

$row = 10 = 2$   
 $col = 011 = 7$

0 1 2 3 5 6 7 8 9 10 11 12 13 14 15

0

1

2

3



## DES – הסבר הרעיון

- DES מורכב משני מרכיבים עיקריים:
  - ערבובים (פרמוטאציות) גלובליים של כל הבלוק
  - פונקציות הצבה לא ליניאריות (S-boxes) – שמונה במקביל
  - כל S-box מקבל 6 ביטים קלט ומוציא 4 ביטים על פי טבלה
  - הסיבה לפירוק המילה לשמונה קבוצות קטנות של ביטים:
- הגדרת פונקציה לא ליניארית כללית היתה דורשת טבלה גדולה.
- הטבלה עבור כל S-box דורשת  $2^6 = 4 \times 2^6 = 256$  ביטים. הסבר:
  - יש  $2^6 = 64$  ערכים אפשריים של קלט
  - לכ ערך יש לרשום את ערך הקלט שהוא בן 4 ביטים,
- DES מפעיל 8 S-boxes הדורשים יחד 2048 סיביות
- S-Box עם קלט 48 ביט היה דורש  $32 \times 2^{48} \sim 9 \times 10^{15}$  סיביות
- המון זיכרון וזמן חיפוש.

## התיאוריה של סודיות צפנים

- תיאוריה מתמטית של מידת סודיות צופן פותחה ע"י Shannon
  - היא מסתמכת על תורת המידע (Information Theory) שהוא ייסוד
  - תורת המידע מתארת באופן הסתברותי כמות מידע בתוך טקסט.
- שנון הגדיר שני מרכיבים של סודיות צופן:
  - Confusion (בלבול)
  - Diffusion (פיזור, חלחול)
- **Confusion** הוא הסתרת המידע באופן שקשה לפענח אותו
- **Diffusion** הוא פיזור המידע על פני תחום רחב של טקסט
- במובן הבא:
  - כל ביט מקורי משפיע על הרבה ביטים במסר המוצפן
  - וגם כל ביט מוצפן מושפע מהרבה ביטים מהמסר המקורי





## 2. חיזוק השימוש ב-DES.

- למשל, מצפינים עם DES פעמיים עם מפתחות שונים – נקרא **Double DES**.  
ניקח לדוגמא הצפנה כפולה. הקוד בין שתי ההצפנות ייקרא I. ב-DES יהיה:  $|K_1|=|K_2|=56$ ,  
וכן  $|M|=|I|=|C|=64$ .  
נגד כך יש התקפה שנקראת **meet-in-the-middle**. נניח כי יש לנו **known plaintext**. באופן זה:  
תחילה מנסים לעבור על כל המפתח K1 של ההצפנה הראשונה, נבדוק מהו I שמתאים ל-M.  
כעת נעבור על כל האפשרויות של המפתח K2 וננסה לבצע Decrypt של C, נגיע ל-Iים  
פוטנציאליים. עד כה ביצענו  $2^{57}$  פעולות.  
נעבור רק על ה-Iים המתאימים (הזהים בין שני התהליכים). יש סטטיסטית  $2^{48}$  זוגות מפתחות  
המקיימים את שני התנאים האלה, כלומר מצפינים ומפענחים את M ל-I ואת I ל-C, בהתאמה.  
אותם נצטרך לבדוק מול מסר נוסף, ואז נקבל זוג מפתחות נכון בהסתברות קרובה ל-1.  
כך בעזרת סדר-גודל של  $2^{60}-2^{58}$  פעולות פיצחנו את double DES, על-אף שמרחב המפתחות  
הוא  $2^{112}$ .
- **Triple DES** – אותו סיפור, רק עם שלוש הצפנות DES רצופות. נקרא גם **EEE**, שלוש פעמים  
Encrypt. כאן גם עם **meet in the middle**, צריך לפחות מצד אחד לעבור שתי הצפנות, כלומר  
112 ביטים של מפתחות. לכן כאן החיזוק נאמד ב-  $2^{120}$ , ב"עלות" של 168 ביטים של מפתחות.

## שיטות להפעלת צופן בלוק

- **מבוא:**
  - השיטה הפשוטה להפעלת צופן בלוק: הצפן כל בלוק בנפרד
  - שיטה זו נקראת ECB (Electronic Code Book)
  - מתברר שברוב המצבים ECB פתוח לתקיפות (ר' השקפים הבאים)
  - לכן הוגדרו מספר שיטות אחרות להפעיל צופן בלוק
  - שיטות אלה יוצרות זיקה בין הבלוקים
  - כלומר הצפנת בלוק תלויה בבלוקים הקודמים באיזשהו אופן
  - דבר זה מזכיר קצת את שיטת Autokey של Vigenère
- השיטות הנוספות המקובלות הן: OFB, CFB, CBC
  - ארבע שיטות אלה הוגדרו כשיטות הפעלה תקניות לאלגוריתם DES
  - אולם ניתן להשתמש בשיטות אלה עם כל צופן-בלוק
  - נדון בשיטות אלה בשקפים הבאים.
  - אנו מניחים שגודל הבלוק עליו פועל הצופן הוא 64, כמו ב-DES

ECB

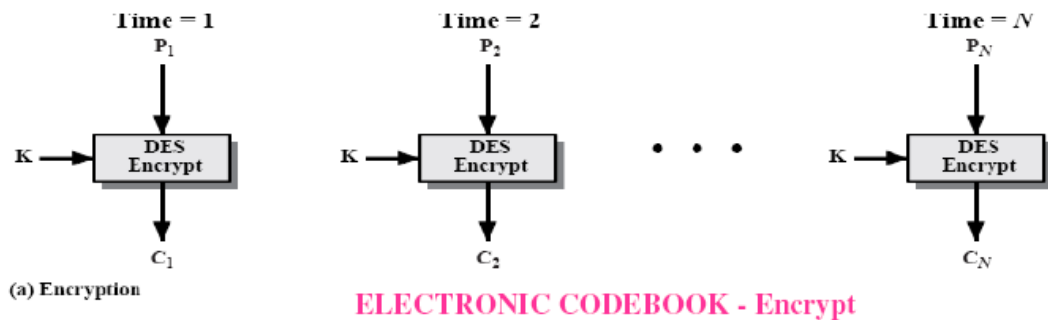
CBC

CFB

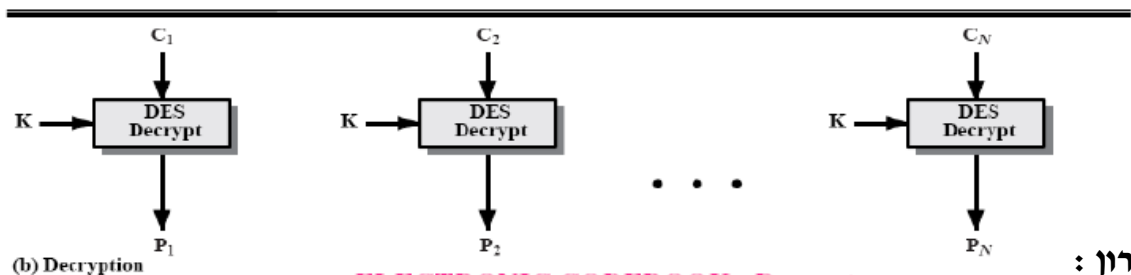
OFB

## electronic code book - ECB

### איור 11: הפעלת DES בשיטת ECB



ELECTRONIC CODEBOOK - Encrypt



ELECTRONIC CODEBOOK - Decrypt

□ עקרון :

□ מצפין כל בלוק של מסר המקור בנפרד.

□ אותו בלוק מקור יוצפן תמיד לאותו צפנבלוק. ( בדומה לספר קוד שבו לכל מילה יש מילת קוד משלה

(קודים בצבא).

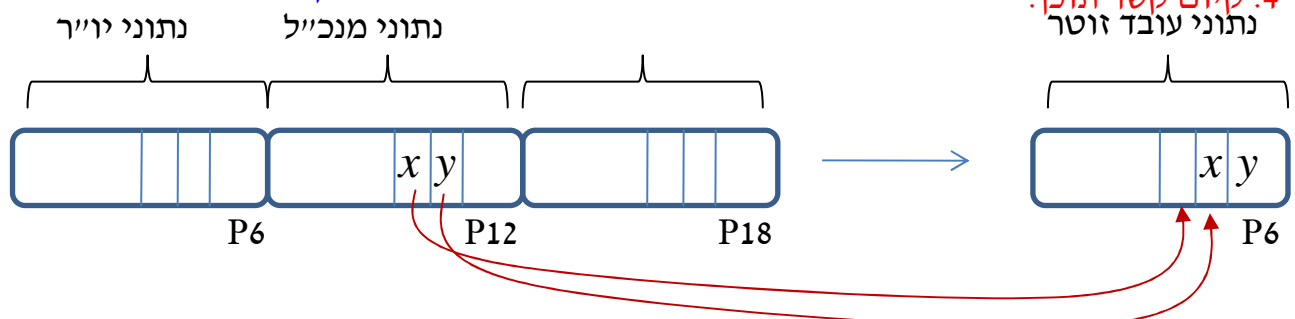
□ ECB עובד טוב לשם הצפנת בלוק בודד.

#### יתרונות ECB

1. מהירות.
2. פשטות.
3. גישה אקראית הצפנה ופענוח.
4. יכולת מקביליות

#### □ חולשת ECB באה לידי ביטוי במסרים ארוכים :

1. בלוקי צופן זהים יהיה גם בבלוקי צפן.
2. מצב זה מקל על תוקפים פוטנציאליים.
3. תוקף יכול להחליף בלוק בבלוק בצופן.
4. קיום קשר תוכן. נתוני עובד זוטרי



נניח כי מדובר במידע על כל עובד ובתאים XY נתוני משכורת העובד.

העתקת משכורת מנכ"ל לעובד..

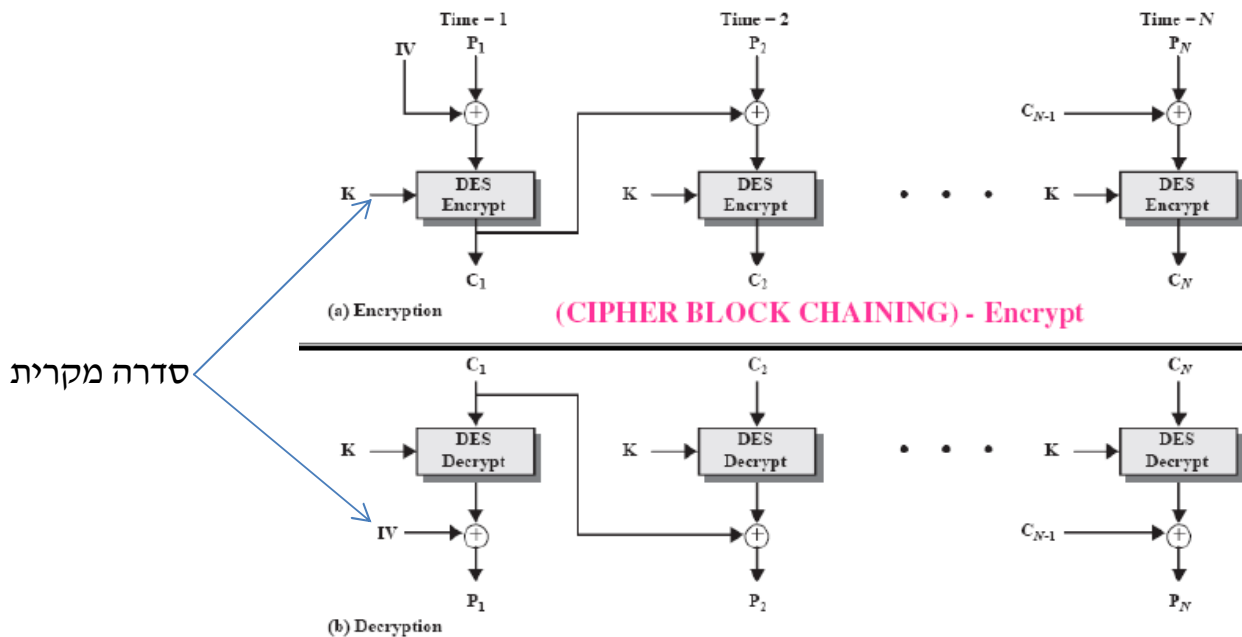
בפענוח – שהוא אותו מפתח עם אותו אלגוריתם.

ע"ס אינפורמציה מסוימת שאנו לא יודעים את תוכנה אלא נתון לגביה...ומתוכו משנים מידע לצידנו

פתח לתקיפת המערכת

## פתרון לחסרון ה-ECB

# איור 12: הפעלת DES בשיטת CBC



סדרה מקרית

## שיטת Cipher Block Chaining (CBC)

יתרונות נוספים:  
יש גישה אקראית בפענוח. ניתן לבצע פענוח מקבילי.  
יש סנכרון עצמי – אם חלק מהבלוקים הלכו לאיבוד ניתן לפענח את שאר החלקי.  
שינוי בלוק כלשהו תשנה את הצפנים של כל הבלוקים הבאים אחריו.

- שיטת CBC (ראה איור 12):
    - ההצפנה:  $C_i = E_K(C_{i-1} + P_i)$ , כאשר  $C_0 = IV$  (Initial Vector) הוא סדרת ביטים מקרית באורך 64 ביט.
      - הוא נשלח ליעד יחד עם הצפנמסר
      - לפני הצפנת  $P_i$  מוסיפים לו XOR את הבלוק המוצפן הקודם
    - הפענוח:  $P_i = C_i + D_K(C_{i-1})$  (כאמור,  $C_0 = IV$  ידוע ביעד)
    - היתרונות:
      - הצפנת כל בלוק תלויה בבלוקים הקודמים. לכן:
        1. אי אפשר לזהות בצפנמסר איזה בלוקים זהים במסר המקור
        2. העתקת צפנבלוק במקום אחר לא תיתן תוצאה ידועה מראש
- זה פותר את הבעיות שהוזכרו בקשר להצפנת ECB

## חסרונות שיטת CBC

חסרונות נוספים:  
אין גישה אקראית בהצפנה.  
הצפנה סדרתית.

- החסרונות:
  1. שיבוש בצפנבלוק אחד משבש את כל מה שבא אחריו
    - ב- ECB זה אינו קורה
  2. התוקף יכול לשנות בלוק  $P_i$  ביעד כרצונו על חשבון שיבוש הבלוק הקודם
    - פירוט בשקף הבא
  3. אם ה- IV נשלח גלוי, ניתן לשנות ביטים של  $P_i$  כרצונו
    - פירוט בשקף הבא
  4. אם IV קבוע, ניתן לדעת אם מסר שגרתי נשאר קבוע בתוכנו ומתי הוא משתנה
- כדי לפתור חלק מבעיות אלה:
  - א- בוחרים IV אקראי
  - ב- שולחים אותו מוצפן בעזרת ECB יחד עם הצפנמסר
- שאלה: איזה מהבעיות פותרים כלים א' ב' ?

## פירוט תקיפת CBC

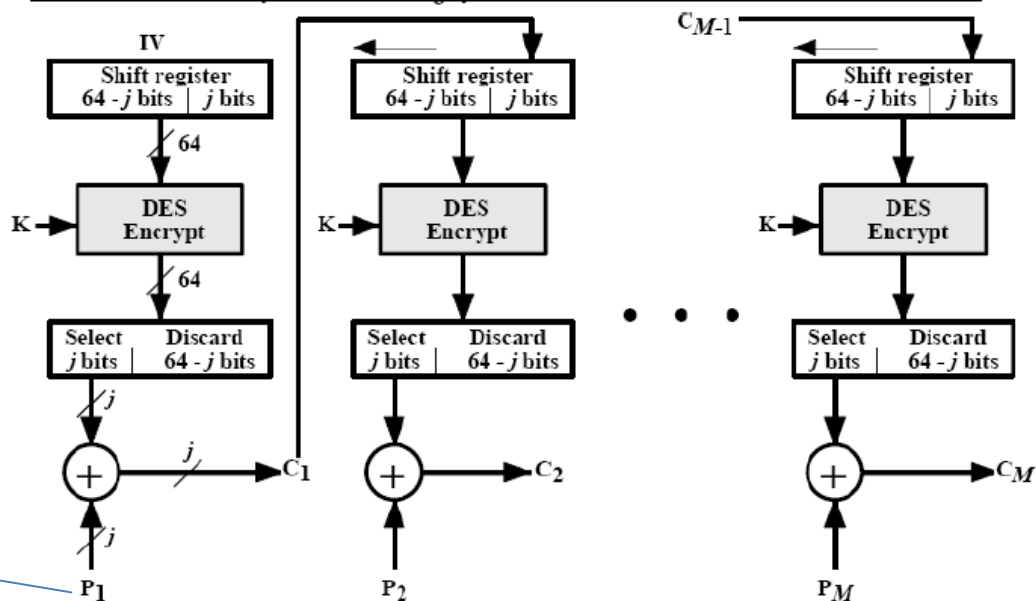
- הסבר לשיטת תקיפה 2
  - מזכיר שפענוח בלוק  $i$  הוא על ידי הנוסחה:
 
$$P_i = C_{i-1} + D_K(C_i)$$
  - נניח שאנו יודעים את בלוק המקור  $P_i$  ורוצים לשנותו לבלוק  $Q_i$ .
  - לשם כך נחשב את הפרש ה-XOR של שני הבלוקים:
 
$$G_i = Q_i + P_i$$
  - עכשיו נשנה את  $C_{i-1}$ :
 
$$C_{i-1}^* = G_i + C_{i-1}$$
  - אז כאשר היעד ירצה לקבל את  $P_i$  הוא יקבל במקומו:
 
$$\begin{aligned} P_i^* &= C_{i-1}^* + D_K(C_i) \\ &= G_i + C_{i-1} + D_K(C_i) \\ &= G_i + P_i = Q_i \end{aligned}$$
  - מחיר התקיפה: במקום  $P_{i-1}$  יתקבל ביעד זבל (אך שאר המסר יהיה נכון)
  - התקיפה 3 זהה אך כאן נשנה את IV ל-  $IV^*$  ואיש לא יזהה אותו כזבל
- שימוש: ניתן לשנות ספרה במשכורת ע"ח שיבוש שדה הערות שלא יתגלה

# יצירת צופן זרימה (Stream Cipher) CFB

## איור 13: שיטת CFB ( $j$ סיביות) - הצפנה

בראשון אין הזזה.

הזזה היא החץ שמאלה  
שלא מופיע בראשון



*abcdefgh*

(  $j$ -bit CIPHER FEEDBACK ) - Encrypt

על אות אחת

הזזה ימינה

*bcdefgh*  $c_1$

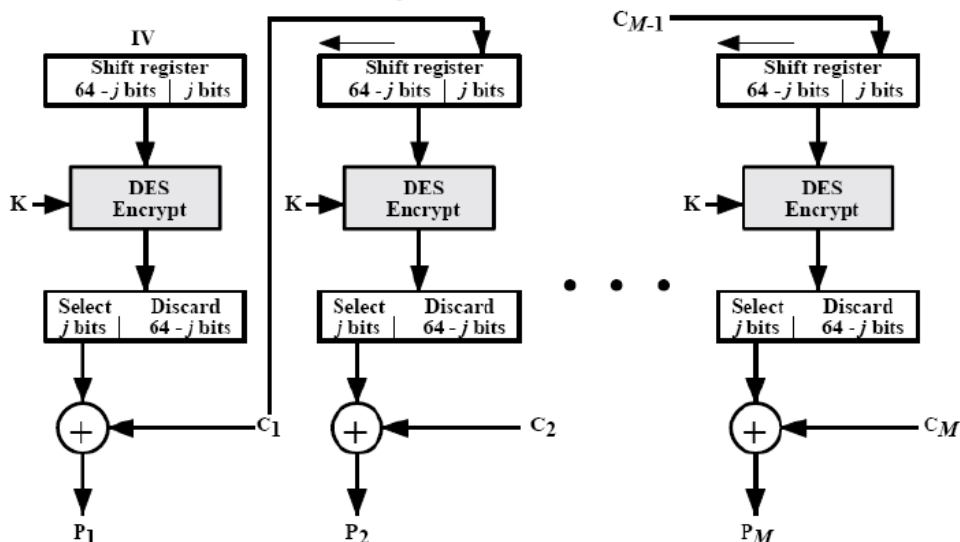
*a'b'c'd'e'f'g'h'*

XOR

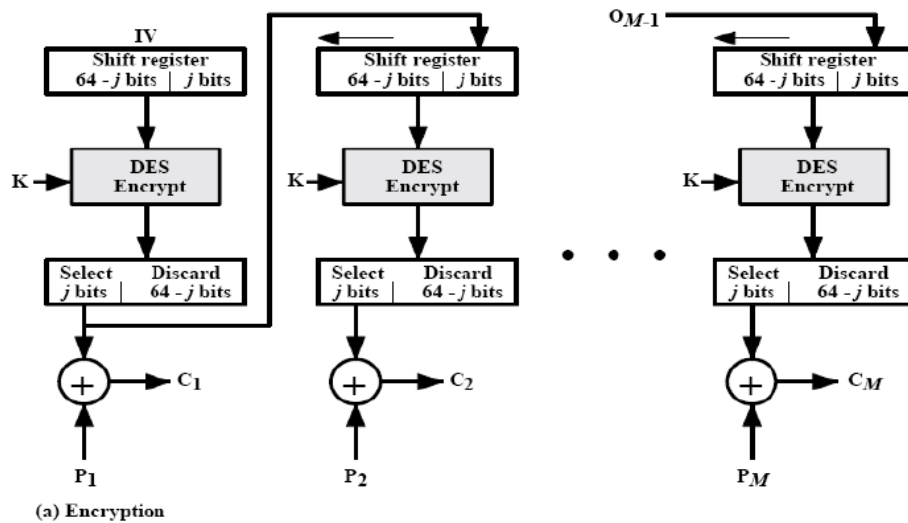
$C_1$

## איור 14: שיטת CFB ( $j$ סיביות) - פענוח

$P_1$



(  $j$ -bit CIPHER FEEDBACK ) - Decrypt

איור 15: שיטת OFB ( $j$  סיביות) - הצפנה(  $j$ -bit OUTPUT FEEDBACK ) - Encrypt

עכשיו גם אם  $C$  בפעולת ה-XOR או בשידור ישתבש הוא לא ישפיע על ה- $C$  הבא אחריו

2. **OFB - Output Feedback**. משתמשים ב- block cipher בכדי לייצר ביטים אקראיים. מתחילים מ-

buffer כלשהו, מצפינים אותו עם block cipher, מקבלים ביטים "אקראיים". מזינים אותם חזרה להצפנה ה- block, ומקבלים עוד. כל קבוצת ביטים שהוצפנה היא ביטים אקראיים, וכן משמשים לייצור עוד ביטים אקראיים.

יתרון – אוטומט סופי, ולכן ניתן לייצר את ה- pad (הביטים האקראיים) מראש.

חסרון – אין סינכרון עצמי.

3. **CFB – Cipher Feedback**. דומה ל- OFB, אך ה- feedback היא של הביטים המוצפנים, ולא של

הביטים שיוצאים מה- CFB. זה לא אוטומט סופי (תלוי ב- message). ה- feedback יכול להיות ביט-ביט, או בלוק בכל פעם (בלוק של ההצפנה הבלוקית) או באמצע.

יתרונות – יש סינכרון עצמי, לפי הביטים האחרונים שהתקבלו.

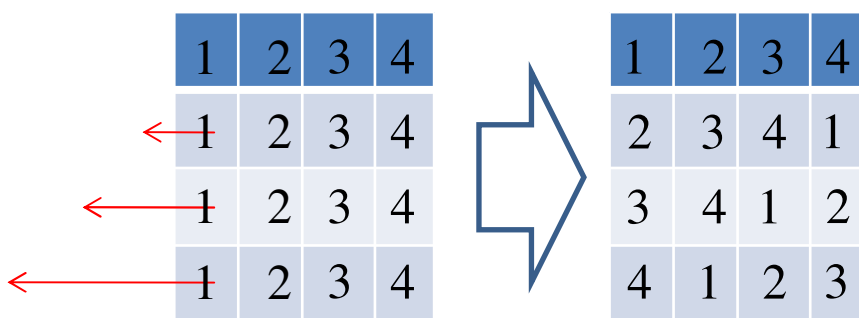
חסרונות – אין יכולת לייצר pad מראש.

לסיכום:

$CBC$	$ECB$	
לא	כן	הצפנה במקביל
בתנאי שהבלוק הקודם קיים	כן	פענוח במקביל
לא	בכל מקום	החלפת בלוקים
רק מהסוף	בכל מקום	קיצוץ בלוקים
רק בלוק אחד	לא	טעות נגררת בפענוח
כן (עם בלוק קודם)	כן	גישה אקראית לקובץ
יש	יש	סינכרון

## צופן AES

- צופן זה משתמש בבלוק של 128 ביט, משמע מספר תוצאות הפענוח / הצפנה הינו  $2^{128}$
  - נתון הרבה יותר גבוה מזה של מ-DES.
  - למעשה מס' האפשרויות הוא: DES בריבוע.
  - אורך המפתח משתנה 128 / 192 / 256
  - DES3 – איטי פי 3. הוא סוג של טלאי על ה-DES
  - הפתרון בדמותו של התקן החדש AES
  - מס' שלבי ההרצה 10 / 12 / 14 תלוי באורך המפתח.
  - כל שלב מורכב מ-4 פעולות (מחזור): תהליך איטראטיבי העובד בתצורה הבאה:
- עבור מפתח של 128



בונה מטריצה 4X4  
מחלק את 128 ל-16 קבוצות.  
בכל משבצת יש 8 ביט [0-255]:

המטריצה נקראת STATE = המצב הנוכחי  
עבור מי שיש את המפתח, הפענוח מהיר.

בשלב ה-0 שמים במטריצה את ה-plain text

מהמטריצה בונים מפתח בגודל 13 11 9

**פעולה 1: S BOX** – מקבל בית ומוציא בית. כל אחד מ-16 הבתים במטריצה נכנס ל-S BOX

את 8 הביטים שמקבלים בכל קובייה מכניסים חזרה לקובייה ממנה נוצר.

**פעולה 2: SHIFTRROWS**: השורה ה-1 נשארת כפי שהיא.

השורה ה-2 זוהי SHIFT RIGHT ציקלי מקום אחד.

השורה ה-3 זוהי SHIFT RIGHT ציקלי 2 מקומות.

השורה ה-4 זוהי SHIFT RIGHT ציקלי 3 מקומות.

כל עמודה כוללת תרומה מכל עמודות מקור...

**פעולה 3: mix columns**

**פעולה 4: XOR**: בין מפתח מסוים של 128 שנגזר מהמפתח המקורי לתוצאה שנתקבלה משלב 3.



## MIX CLOUMS

הכפל נועד לטובת ה-DIFUSION, הערבוב

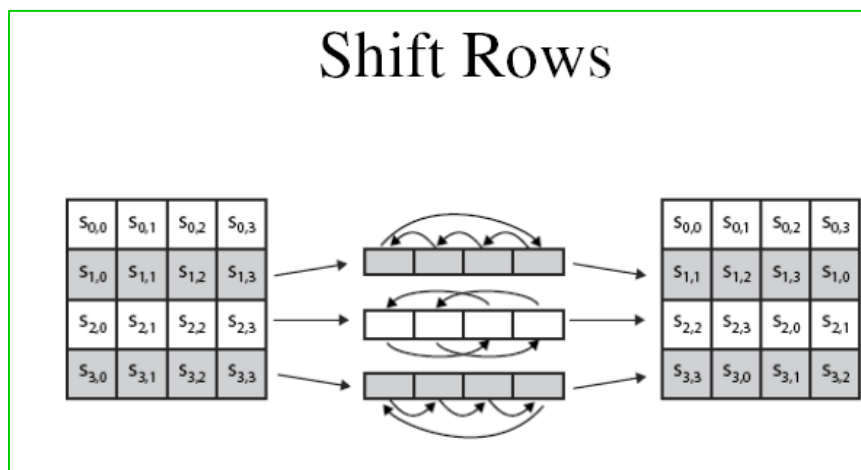
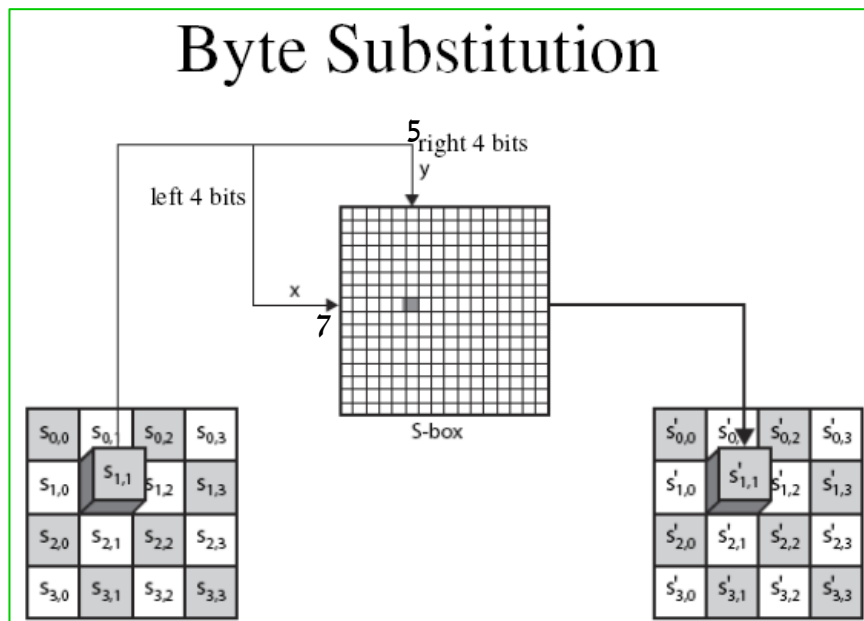
$AB$   
 STATE  
 מטריצה קבועה וידועה  
 מפורסמת בתקן. (4X4)

$AB_1 \quad AB_2 \quad AB_1 \quad AB_2$   
 כפל בשדה גלואה 2 בחזקת 8

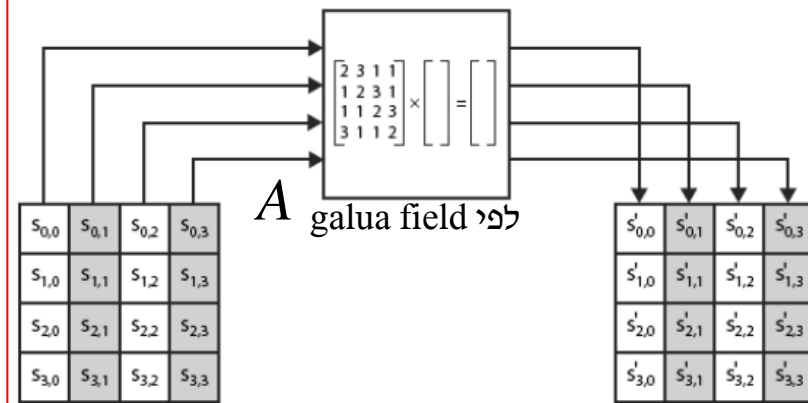
איברים  $\square = GF(p) = p$

$$GF(2^8) = a^7 x^7 + a^6 x^6 + \dots a^1 x^1$$

פעולת היל = כפל מטריצות



## Mix Columns



## Add Round Key

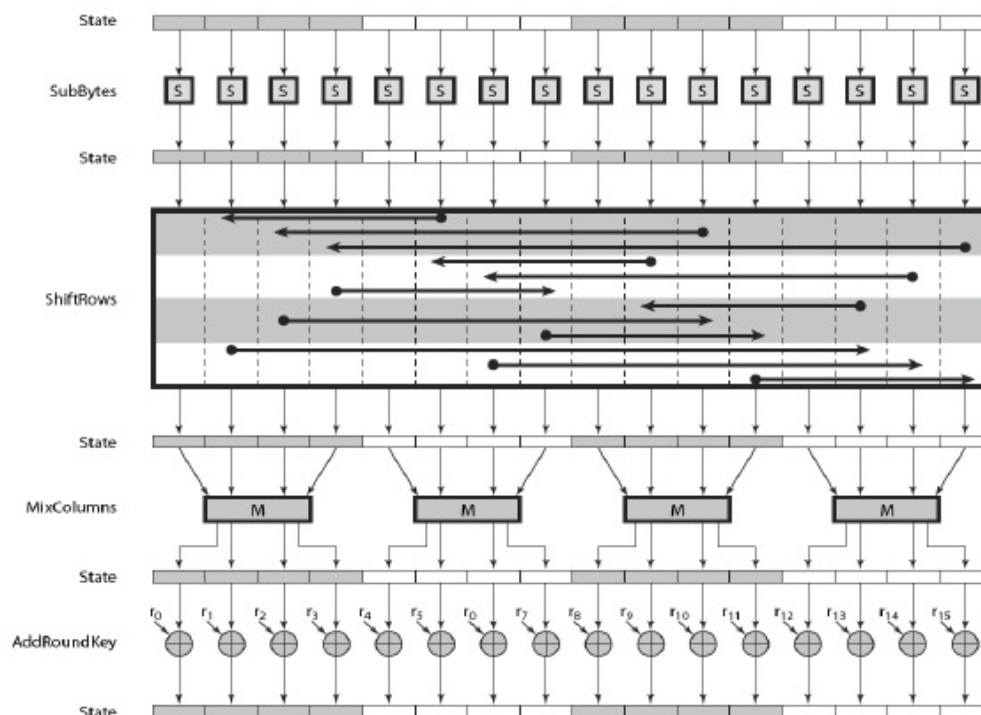


מתוך המפתח

המקורי בונים מפתח

$i =$  מחזור

## AES Round



## Confusion + diffusion

### Diffusion - פיזור.

פיזור המידע ע"פ תחום רחב של טקסט: כל ביט מקורי משפיע על הרבה ביטים במסר המוצפן ולכן גם כל ביט מוצפן מושפע מהרבה ביטים במסר המקורי. אותיות ה- plaintext מתפזרות על פני ה-chipper text ולהיפך. כל עוד ב- CT תלוי בערכי הרבה מאותיות ה- PT. בצפנים הקלאסיים הבסיסים אין כלל diffusion (במיוחד המונו-אלפביתים והואזנר). "ספקיות" ה-diffusion הן הפרמוטציות. בצופן playfair יש מעט יותר diffusion (בשל העבודה בזוגות). בצופן HILL יש diffusion רב (היחיד מהצפנים הקלאסיים). בין הצפנים המודרניים, עבור DES ו-AES = יש diffusion רב.

### confusion = בלבול-

הסתרת מידע באופן שקשה לפענח אותו. מכניס להצפנה אלמנט קושי בהבנת האלגוריתם. "ספקיות" ה-confusion הן קופסאות ה-S. בניסיון הפענוח, מקשה להבין את התהליך שבוצע ולשחזרו. confusion גדול – צופן חד פעמי. one time pad

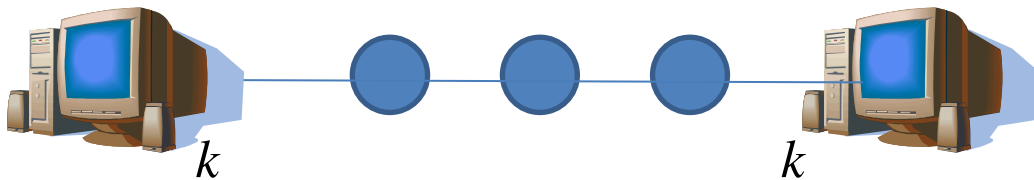
עבור צופן DES הפרמוטציות מספקות את ה-diffusion וקופסאות ה-S את ה-confusion

## כיצד משתמשים בהצפנה:

איפה שמים את ההצפנה?

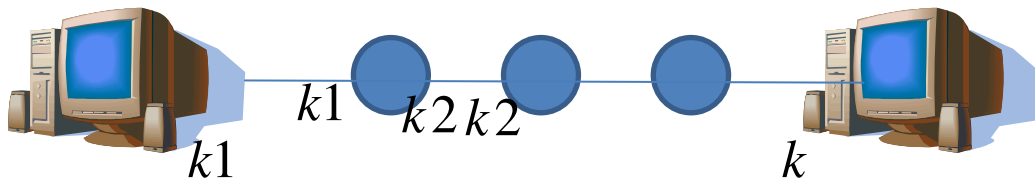
1. בקצוות ETE – end to end
2. ברשת עצמה LBL – link by link

### הצפנת קצוות ETE



- ☐ יודע כמה ומתי יש תנועה
- ☐ בנתבים לא יודעים כלל מה קורה. המידע שם מוצפן.
- ☐ אנו מצפינים רק את ה-DATA (יש פחות מידע להצפין)
- ☐ הכותרת H3 חייבת להיות גלויה לכל אורך הדרך.
- ☐ ההצפנה הינה בשכבות 4 ו-5.
- ☐ קל להאזין משכבה 3 = ניתוח תנועה.
- ☐ כל ה-host במערכת צריכים מפתח – דבר הגורר בעיית הפצת מפתחות.

### LINK BY LINK



- ☐ אם נצפין על כל לינק – נוכל להצפין גם את כותרת 3.
- ☐ הרווחנו את הצפנת שכבה 3
- ☐ חסרון – לוקח הרבה יותר זמן.
- ☐ חסרון עיקרי - כל ה-data חשוף בנתבים.

### שילוב יוצר תמונה יותר טובה LBL+EPE

## הצפנת LINK BY LINK



(a) Application-Level Encryption (on links and at routers and gateways)



On links and at routers



In gateways

(b) TCP-Level Encryption



On links

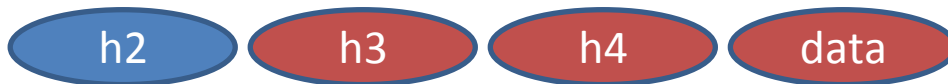


In routers and gateways

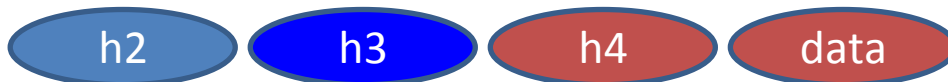
(c) Link-Level Encryption

Shading indicates encryption.

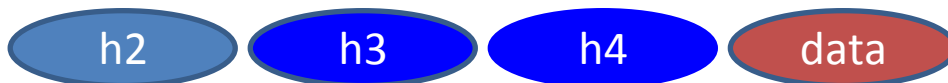
TCP-H = TCP header  
 IP-H = IP header  
 Net-H = Network-level header (e.g., X.25 packet header, LLC header)  
 Link-H = Data link control protocol header  
 Link-T = Data link control protocol trailer



הצפנה בשכבה 3



הצפנה בשכבה 4



הצפנה רבה לעומת פתיחה וסגירה = בהוא חייב לפתוח הכל ....  
 הכי מאובטח בערוץ הכי פרוץ בנתב.

**ניתוח תנועות וזרימת מידע:**

1. זיהוי בין מי למי.
2. השיטה להלחם בזה: העברת דיסאינפורמציה.
3. השיטה הסטנדרטית – traffic padding – תנועה קבועה או תנועה אקראית של מידע כדי להסתיר את התנועות האמיתיות.
4. ההצפנה מבוצעת ב-link, אז גם כותרת הכתובות מוצפנת. (כמובן שבכל link חייב להיות המפתח המתאים).

ETE	3.5 שיטות: (א)
LBL	(ב)
שילוב של 2 השיטות.	(ג)

**כיצד מצפנים מפתחות?**

כמובן שבהצפנה סימטרית, חייבים שבכל צד יהיו מפתחות זהים.

**בעיה: כיצד מאבטחים את הפצת המפתח???**

1. פיזית.
2. גוף מרכזי בוחר את המפתח ומפיץ לכל צד.
3. אם לשני הצדדים כבר יש מפתח, שימוש במפתח הקודם כדי להעביר את המפתח החדש.  
(בעייתי כי אם המפתח הקודם נפרץ אז גם הבאים אחריו כפרוצים...)
4. גוף אמצע שלכל אחד מהצדדים יש קשר מוצפן איתו: A ל-C + C ל-B.

**היררכית מפתחות:**

2 רמות מפתח:

**Session / temporary key (מפתח עבודה):**

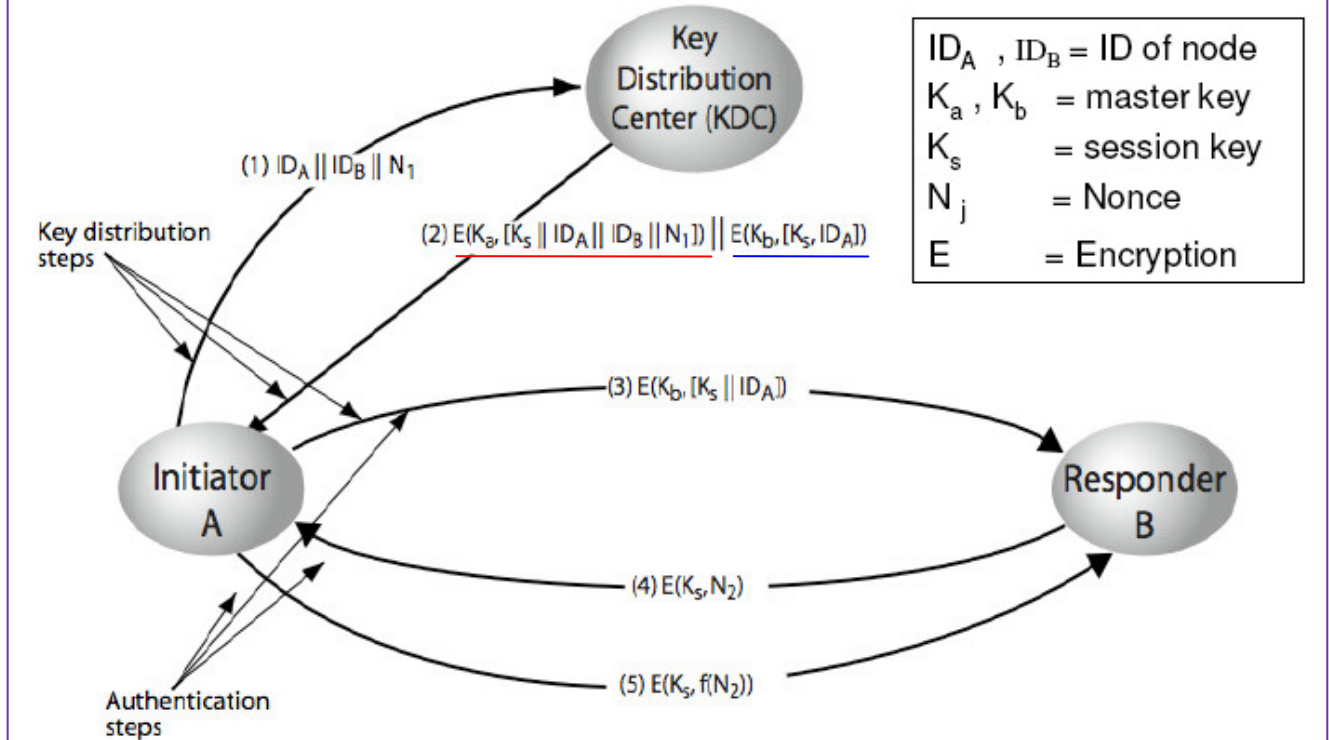
- מפתח זמני המיועד לזמן קצר (התקשרות בודדת).
- משמש להצפנת המידע ובסיום נפטרים ממנו.
- לרוב מיוצר ומופץ ע"י מרכז הפצה.
- מפתח שמיועד להצפנת מפתחות.

**Master key:**

לכל משתמש יש מפתח משותף ייחודי עם המרכז שמשמש להצפנת המפתחות (כלומר ל-A יש מפתח משותף שונה מ-B).

## תרחיש אופייני להפצת מפתח

# Typical Key Distribution Scenario



KS – המפתח של ה-session שיוצר בין A ל-B

KA – מפתח בין הסנטר ל-A

זמן חיות של מפתח מוגבל ל-session בודד או גודל מסוים כדי לא לאפשר יותר מידי חומר שיעבור עם אותו מפתח.

❑ A רוצה לדבר באופן מוצפן עם B.

❑ A יוצר קשר עם מרכז ההפצה (1):

מוסיף לו את  $N_1$  כדי שיהיה ברור שמדובר בשיחה נוכחית בין A ל-B ולא בהתקשרות שהייתה בעבר.

N מהווה סימן זיהוי שמדובר בהתקשרות הנוכחית.

❑ מרכז ההפצה מחזיר תשובה ל-A (2): בתשובה מופיע:

(1) המפתח הסימטרי של ה-session הנוכחי, העובדה שהמפתח יהיה בין A ל-B ו- $N_1$  לאישור כי מדובר

בשיחה שזה עתה נתבקש אישור לגביה. (סימון אדום)

(2) מסר מוצפן לפי B (שבשלב הבא ימסר ל-B) בדבר רצונו של A לנהל שיחה עם B ואותו מפתח סימטרי

שהועבר ל-A מועבר גם ל-B.

❑ A מעביר ל-B את ההודעה שקיבל עבורו ממרכז ההפצה 3 הכוללת את המפתח לשיחה Ks. ההודעה מועברת מוצפנת לפי B.

❑ B שולח ל-A הודעה מוצפנת (4) לפי הסימטרי שקיבל ממנו בשלב (3) הכוללת  $N_2$ .

❑ A מקבל את ההודעה ושולח בתגובה (5) הודעה מוצפנת במפתח החדש הכוללת פונקציה של  $N_2$ .

❑ **הסבר על שלב 2 והפונקציה  $f(N_2)$**

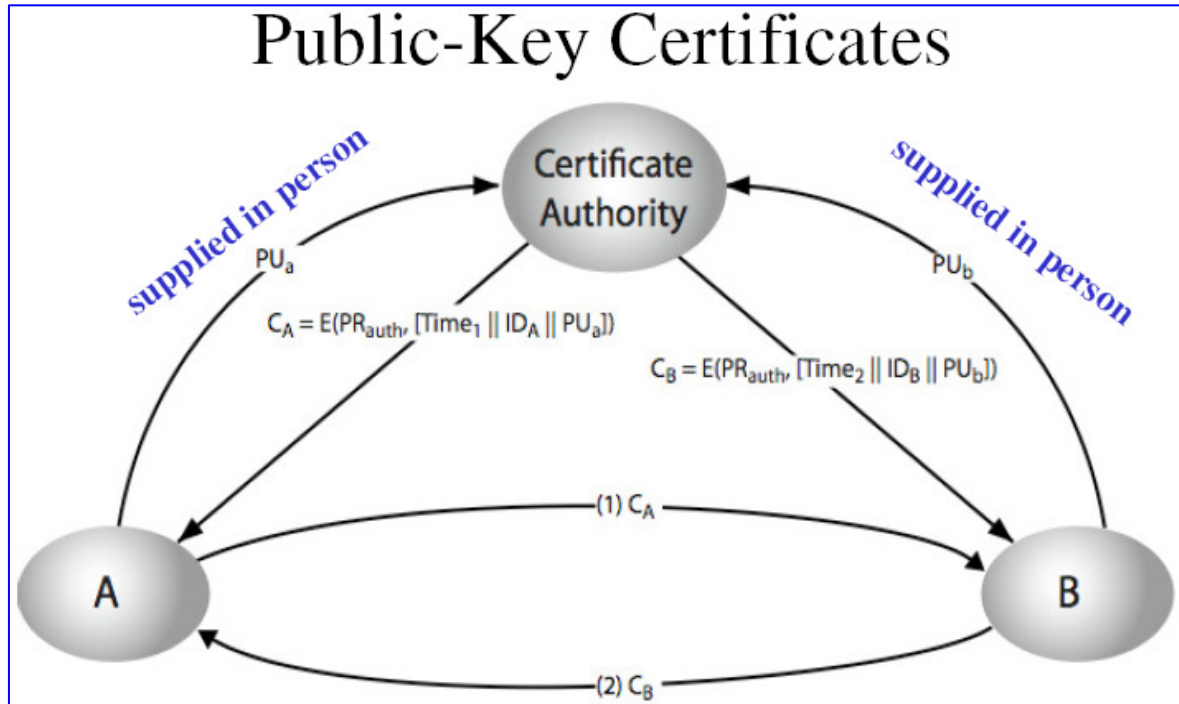
אם B שולח את  $N_2$  ו-A היה מחזיר לו את  $N_2$  מבלי לבצע שום פעולה עליו היינו חשופים לתקיפה כי כל מי שיירט את הודעה 4, יכול למעשה להחזיר אותה כפי שקיבלה ו-B יחשוב שמדובר ב-A.

דוגמא נוספת להפצת מפתח.

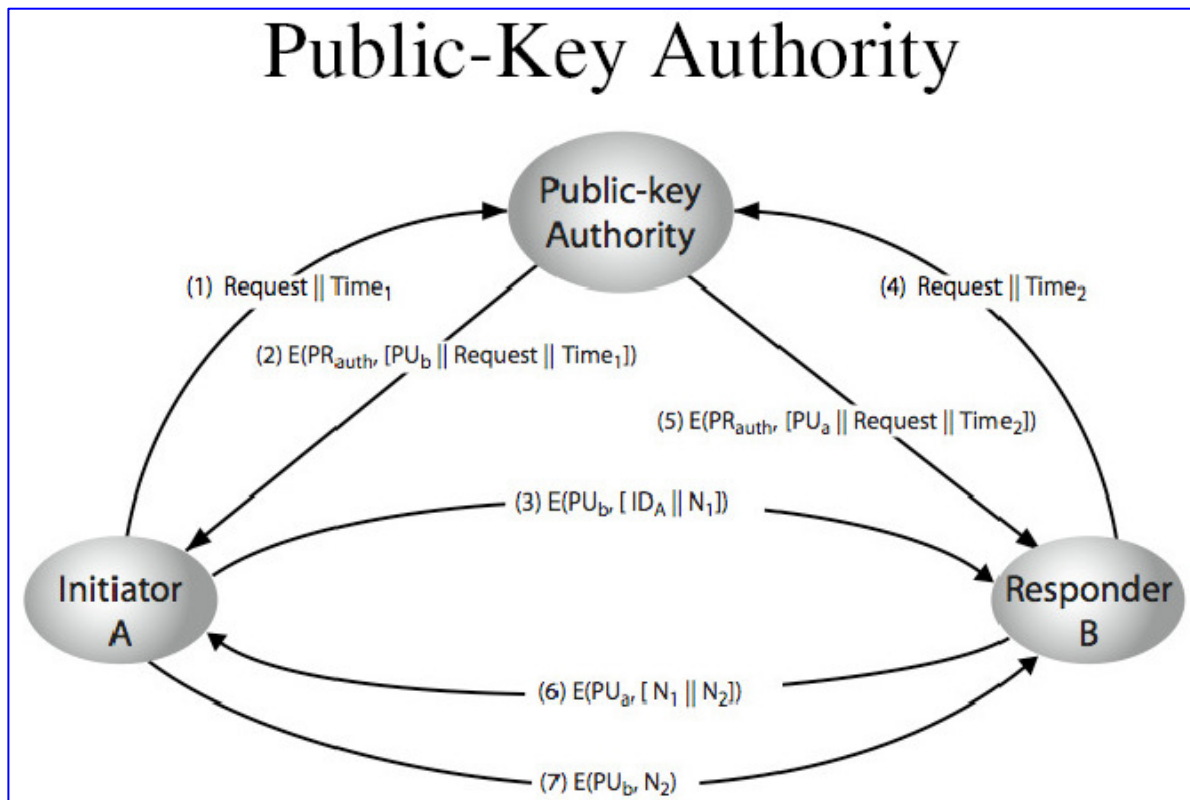
בדוגמא זו מדובר על שיטת הצפנה בדומה לצבא בו מחליפים מפתחות מול מרכז ההפצה אחת לתקופה

מסוימת ואז ניתן ליצור התקשרות בין A ל-B

Time מחליף את N



דוגמא נוספת להפצת מפתח פרטי ופומבי:





## הצפנה במפתח פומבי

עבור מס' ראשוני  $147 = 3 * 7^2 = 3^1 * 5^0 * 7^2 * 11^0 * 13^0 \dots$   
משפט פרמה הקטן:

אם  $P$  ראשוני ואם  $a$  זר  $a^{p-1} \equiv 1 \pmod{p}$

– if  $\gcd(a, p) = 1$  then  $a^{p-1} \equiv 1 \pmod{p}$

– for any  $a$ ,  $a^p \equiv a \pmod{p}$

$$p = 5$$

$$a = 2$$

$$2^4 = 16 \pmod{5} = 1$$

$$2^5 = 32 \pmod{5} = 2$$

פונקציה אוילר  $\phi(n)$

ב-  $\square_p$  לכל  $a \neq 0$ ,  $a^{p-1} = 1$

$N$  מס' כלשהו  $\{0 \dots n-1\} = Z_n$

$\phi(n)$  כמות המספרים ב-  $Z_N$  שזרים ל- $n$

$$12: \{1, 5, 7, 11\} \Rightarrow \phi(12) = 4$$

$$15: \{1, 2, 4, 7, 8, 11, 13, 15\} \Rightarrow \phi(15) = 8$$

$$10: \{\dots\} \Rightarrow \phi(10) = 4$$

$$\phi(p) = p - 1$$

$$\phi(pq) = (p-1)(q-1) \quad \text{P Q ראשוניים שונים}$$

$$\phi(15) = \phi(3 * 5)$$

$$\phi(15) = (3-1)(5-1) = 8$$

מתאים

אם המספר אינו מתחלק בשני מספרים ראשוניים, החישוב נעשה ידנית:

$$\phi(16) = \{1, 3, 5, 7, 11, 13, 15\} = 8$$

- Theorem (Euler):  
Let  $n$  be any natural number  $> 1$ . Then:  
a) if  $\gcd(a, n) = 1$  then  $a^{\Phi(n)} \equiv 1 \pmod{n}$   
b) for any  $a, k$ ,  $a^{k\Phi(n)+1} \equiv a \pmod{n}$
- Example for part (a)  
 $a=3; n=10; \Phi(10) = (2-1) * (5-1) = 4;$   
hence  $a^{\Phi(n)} = 3^4 = 81 \equiv 1 \pmod{10}$   
 $a=2; n=11; \Phi(11) = 11-1=10;$   
hence  $a^{\Phi(n)} = 2^{10} = 1024 \equiv 1 \pmod{11}$
- Example for part (b)  
 $a=2; k=3, n=10; \Phi(10)=4;$   
hence  $a^{k\Phi(n)+1} = 2^{3*4+1} = 2^{13} = 8192 \equiv 2 \pmod{10}$

### משפט אוילר / מסקנה סופית:

$$a^{k\Phi(n)+1} \equiv a \pmod{n}$$

$c = a^e \pmod{n} \quad \leftarrow \text{הצפנה של } a$

$$a^{ed} \equiv (a^e)^d \equiv c^d \pmod{n} \quad \leftarrow \text{פענוח של } c$$

$$a^{\Phi(n)} \equiv 1 \pmod{n} \quad \text{אם } a \text{ זר ל-} n, \text{ אז} \quad (\text{א})$$

$$\text{מכסה את כל המקרים (לא רק } a \text{ זר ל-} n): \quad (\text{ב})$$

$$\underbrace{(a^e)^d}_{\text{RSA}} = a^{ed} = a^{k\Phi(n)+1} \equiv a \pmod{n} \quad \text{לכל } a \in \mathbb{Z}_n \text{ ולכל } K \text{ טבעי:}$$

$$ed = k\Phi(n) + 1$$

$$ed \equiv 1 \pmod{\Phi(n)}$$

דוגמא:

$$n = 35$$

$$n = 7 * 5 \Rightarrow \Phi(35) = (7-1)(5-1) = 6 * 4 = 24$$

בחר  $e$  גדול מ-7:  $7 < e$  הכי קטן

$$e = 11 \quad e \text{ חייב להיות זר ל-} \Phi(n)$$

משפט Fermat אומר שעבור כל מספר ראשוני  $p$  ועבור כל מספר טבעי  $0 < a < p$  מתקיים:

$$a^{p-1} \pmod{p} = 1$$

משפט Fermat הוא מקרה פרטי של Euler

## תכונות מפתח פומבי / פרטי:

□  $E_{PU_A}$  היא פונקציה המצפינה בעזרת המפתח  $PU_A$

□ הפונקציה  $E_{PU_A}$  היא "פונקציה חד כיוונית", כלומר מידיעתך הפונקציה (כולל המפתח

$PU_A$  לא ניתן למצוא את הפונקציה ההפוכה (שהיא  $D_{PR_A}$ ), אלא ע"י כמות גדולה

מאד של עבודת חישוב לא מעשית.

□ את המסר  $E_{PU_A}(M)$  יכול לפענח רק  $A$ . הפענוח מבוצע ע"י הפונקציה  $D_{PR_A}$

השירות שמבטיחות ההצפנה והפענוח האלה הוא העברה חסויה של מסר מכל משתמש אל  $A$ .

□ את המסר  $E_{PR_A}(M)$  יכול לפענח כל אחד. הפענוח הוא ע"י הפונקציה  $D_{PU_A}$

השירות שמבטיחות ההצפנה והפענוח האלה הוא: אימות זהות שולח, אימות שלמות תוכן

ההודעה וחתימה ספרתית.

## אלגוריתם RSA

נניח שה- $P$  בגודל  $M$ ,  $0 \leq M \leq n$ ,  $M$  בינארי.

על מנת להצפין: לוקחים את המפתח הפומבי של היעד ומחשבים את  $c$  באופן הבא:

$$c = M^e \bmod n \quad PU\{e, n\}$$

על מנת לפענח: לוקחים את המפתח הפרטי של היעד ומחשב את

$$M = c^d \bmod n \quad PR\{d, n\}$$

# RSA

## ה-RSA מבוסס על קושי של פירוק מספר לשני מספרים ראשוניים.

user generates his/her public/private key pair as follows:

1. selects two large primes at random:  $p, q$
2. computes the system modulus  $n=p \cdot q$
3. computes  $\Phi(n) = (p-1)(q-1)$
4. selects a random public index  $e$  satisfying:

$$(*) \quad 1 < e < \Phi(n), \quad \gcd(e, \Phi(n)) = 1$$

4. finds the private index  $d$  by solving the equation:

$$(**) \quad e \cdot d = 1 \bmod \Phi(n), \quad 0 < d < n$$

Qn1 : Write (\*\*) in the form  $d = \dots$

Qn2 : Why is there always a solution to (\*\*) ?

5. publishes his/her public key:  $PU = \{e, n\}$
6. keeps secret the private key:  $PR = \{d, n\}$

אין בעיה להגיע ל-N (מכפלה של 2 ראשונים, אבל לשחזר אותו כמעט בלתי אפשרי לכן גם אם N ידוע, זה לא מספיק לפענוח. מה שמונע מפורץ למצוא את d.

$$d = e^{-1} \bmod \Phi(n)$$

## דוגמא / RSA

1. נבחר אקראית 2 מספרים ראשוניים:  $p = 17, q = 11$
2. מכפלה מייצרת עבורנו את המספר  $n = pq = 17 * 11 = 187$
3. חישוב  $\Phi(n) = \Phi(187) = (p-1)(q-1) = 16 * 10 = 160$
4. נבחר אקראית מפתח פומבי (e) כך ש:

$$e=7 \left\{ \begin{array}{l} (א) \quad \gcd(e, 160) = 1 \quad e \text{ זר ל-} 160 \\ (ב) \quad 1 < e < 160 \quad 1 < e < \Phi(n) \quad \text{וגם} \end{array} \right.$$

5. נמצאת את המפתח הפרטי (d) כך שיקיים:

$$7 * d = 1 \bmod 160 \quad \left\{ \begin{array}{l} (א) \quad e * d = 1 \bmod \Phi(n) \\ (ב) \quad 0 < d < n \end{array} \right.$$

$$7 * 23 = 161 = 1 \quad \downarrow \quad d = 23$$

## מסקנות:

1. מפתח ציבורי:  $PU = \{7, 187\}$
2. מפתח פרטי:  $PR = \{23, 187\}$  (סודי)

עֵתָה , מְשִׁישׁ בִּידִינוּ מִפְתַּח פְּרִטִי, מִפְתַּח פּוֹמְבִי וּמִסָּר לַהֲצַפְנָה  $M$ , נִצְפִּין אוֹתוֹ:  $M = 88$

נוֹדָא שׁ:  $0 \leq M \leq n$  (בִּהְתָּאִם לַהֲתַנִּיית הָאֲלִגּוּרִיתִם)

$$0 \leq 88 \leq 187$$

$$c = a^e \bmod n \quad c = 88^7 \bmod 187 = 11$$

עֵתָה נִנְסָה לַפְעֵנַח אֶת הַצּוּפִן:

$$PR\{d, n\} \quad M = c^d \bmod n \quad M = 11^{23} \bmod 187 = 88$$

$$n = pq = 13 * 5 = 65$$

תְּרַגִּיל נוֹסֵף:

$$\Phi(n) = 48 \quad \text{choose } e = 11$$

$$p = 13, \quad q = 5$$

נִתּוֹן:

$$\gcd(e, \Phi(n)) = \gcd(11, 48)$$

$$48 : 11 = 4(4)$$

$$4 = 48 - 4 * 11$$

$$\gcd(11, 4)$$

$$11 : 4 = 2(3)$$

$$3 = 11 - 2 * 4$$

$$3 = 11 - 2(48 - 4 * 11)$$

$$3 = 11 - 2 * 48 + 8 * 11$$

$$3 = 9 * 11 - 2 * 48$$

$$\gcd(4, 3)$$

$$4 : 3 = 1(1)$$

$$1 = 4 - 3 * 1$$

$$1 = 48 - 4 * 11 - (9 * 11 - 2 * 48)$$

$$1 = 3 * 48 - 13 * 11$$

$$\gcd(11, 48) = 1$$

$$d = e^{-1} \bmod \Phi(n)$$

$$88^{11} = ??$$

$$88^2 \bmod 187 = a$$

$$88^4 = (88^2)^2 = b^2 \bmod 187 = b$$

$$88^4 = (88^2)^2 = b^2 \bmod 187 = c$$

$$88^{10} = 88^8 88 = ca \bmod 187 = d$$

$$88^{11} = 88^{10} 88 \bmod 187 = 88d \bmod 187$$

$$PU = \{11, 65\}$$

אלגוריתם מילר רבין (בודק האם מספר ראשוני) בודק באופן הבא:

## Miller-Rabin Algorithm (1)

- first define algorithm TEST ( $n, a$ ) , for  $1 < a < n-1$
- TEST( $n, a$ ) :
  - 1 Find integers  $k, q, k > 0, q$  odd, so that  $(n-1) = 2^k q$
  2. **if**  $a^q \bmod n = 1$  **then** return (" $n$  maybe prime");
  3. **for**  $j = 0$  **to**  $k-1$  **do**
  4. **if**  $(a^{2^j q} \bmod n = n-1)$   
     **then** return("  $n$  may be prime ")
  5. return ("  $n$  is composite")

מספר לא ראשוני עובר את הבדיקה הזאת ב-25% מהמקרים

הסבר:

## Miller-Rabin Algorithm (2)

- Miller\_Rabin( $n$ ):
  - choose  $d$  independent random numbers  
 $1 < a_j < n-1$  ,  $j = 1, 2, \dots, d$
  - perform Test( $n, a_j$ ) for  $j = 1, 2, \dots, d$
  - if one of the results is "  $n$  is composite"  
     then  $n$  is composite
  - if all results are "  $n$  may be prime"  
     then  $n$  is prime with high probability

# Number of Tests d in Miller-Rabin

Theorem: Given integers  $n$  and  $1 < a < n-1$

1. if  $n$  is prime then  $\text{TEST}(n, a) = \text{"n maybe prime"}$
2. if  $n$  is composite, then for random  $1 < a < n-1$

$\text{Probability}\{ \text{TEST}(n, a) = \text{"maybe prime"} \} < 1/4$

Conclusion: Take a large value of  $d$  (e.g.  $>10$ )

- if  $\text{TEST}(n, a) = \text{"n is composite"}$  then, according to (1),  $n$  must be composite
- if  $\text{TEST}(n, a) = \text{"n maybe prime"}$  for  $d$  independent  $a$ 's then  $n$  is almost surely prime:
- in fact, if we assume  $n$  is composite, then by (2) the probability of  $d$  "maybe prime" results is  $< 4^{-d} < 10^{-6}$

## Examples of Test( $n, a$ )

- Example 1:  $n=13$  (PRIME),  $n-1 = 12 = 2^2 \times 3$ ;  $q=3$ ,  $k=2$ 
  - choose  $a=2$  :
 

Step 2:  $a^q = 2^3 = 8 \not\equiv 1 \pmod{13}$ , so continue to step 3

Step 4: ( $j=0$ )  $a^{2^0 q} = 2^3 = 8 \not\equiv 12 \pmod{13}$ ,  
 ( $j=1$ )  $a^{2^1 q} = 2^6 = 64 \equiv 12 = n-1 \pmod{13}$ , so:  
 $\text{Test}(13, 2) = \text{"13 may be prime"}$
  - choose  $a=3$  :  $3^3 = 27 \equiv 1 \pmod{13}$ , so:  
 $\text{Test}(13, 3) = \text{"13 may be prime"}$
- Example 2:  $n=15$  (Composite),  $n-1 = 14 = 2^1 \times 7$ ;  $q=7$ ,  $k=1$ 
  - take  $a=2$  : Step 2:  $a^q = 2^7 \equiv 8 \not\equiv 1 \pmod{15}$ , so continue
  - Step 4: ( $j=0$ )  $a^{2^0 q} = 2^7 \equiv 8 \not\equiv 14 \pmod{15}$ , so:  
 $\text{Test}(15, 2) = 15 \text{ is composite}$

## הסבר לנכונות האלגוריתם:

$$n = 29(\text{odd}) \pmod{n}$$

$$n-1 = 28(\text{even}) \Rightarrow 28:2 = 14$$

$$14:2 = 7$$

$$7$$

$$28 = 2 * 2 * 7 = 2^2 * 7$$

$$(1) a^{n-1} \equiv 1, \quad 1 < a < n \text{ for each } a \text{ (pherm)} \text{ in } (\mathbb{Z}_n \text{ \& } Z = \text{prime})$$

$$(2) \mathbb{Z}_n \Rightarrow (\text{field}) \Rightarrow \text{no zero dividers.} \begin{pmatrix} \text{if } (b * c = 0) \\ b = 0 \text{ or } c = 0 \end{pmatrix}$$

$$\text{choose } a \quad 1 < a < n$$

$$a^{2^k q} = a^{n-1} \equiv 1$$

$$a^{2^2 * 7} = 1$$

$$\text{rule: } n-1 = -1 \pmod{n}$$

$$(a^{2^1 * 7})^2 = a^{2^2 * 7} = 1$$

$$(a^{2^1 * 7})^2 = 1$$

$$(a^{2^1 * 7})^2 - 1 = 0 \quad \text{look like } x^2 - 1 = 0$$

$$(x-1)(x+1) = 0$$

$$\Downarrow$$

$$x = \pm 1$$

$$a^{2^1 * 7} \equiv 1 \quad \text{or}$$

$$a^{2^1 * 7} \equiv -1$$

$$\Downarrow$$

$$\text{line4 positive}$$

$$(a^7)^2 \equiv 1$$

$$(a^7)^2 - 1 \equiv 0 \Rightarrow a^7 = \pm 1$$

$$\Rightarrow a^7 = 1 \quad \text{or}$$

$$a^7 = -1$$

$$\text{line2 positive}$$

$$\text{line4 positive}$$

$$ax = 0$$

$$1. a = 0$$

$$2. a \neq 0$$

$$\Downarrow$$

$$\text{yesh } a^{-1} \Rightarrow a^{-1}(ax) = 0$$

$$\Downarrow$$

$$x = (a^{-1}a)x = 0$$

$$z6 = 2 * 3$$

$$2 \neq 0$$

$$3 \neq 0 \text{ but } 6 = 0$$

$$2 \text{ כן מחלקי } 0 \leq \text{לא שדה}$$



check :

$$a^{2^0 q} = a^q$$

$$a = 2, a^7 = 2^7 = 128 \equiv 12$$

$$j=0: \quad 2^{2^0 7} \equiv -1 \quad = 12 \neq 1 \text{ NO}$$

$$j=1 \quad 2^{2^1 7} \equiv -1 \quad = 12^2 = 144 \equiv -1 \pmod{29} \quad \text{YES} = \text{maybe prime}$$

$\mathbb{F}_7$  is field  $GF(7)$

$$2, 2^2, 2^3 = 2, 4, 1, 2, 4, 1 \text{ no good}$$

$$3, 3^2, 3^3, 3^4, 3^5 = 3, 2, 6 \equiv -1, 4, 5, 1 = 3^6 \text{ (PHERMA) good \& enough}$$

$$\text{insted : } 3 \quad (3*2) \bmod 7 \quad (3*3) \bmod 7 \quad (3*4) \bmod 7 \\ (3*5) \bmod 7 \dots$$

$\Downarrow$

3 is primitive root of 7

## הצפנה במפתח פומבי

למה יש שיטה נוספת (לא סימטרית כמו שלמדנו עד עתה):

1. צד ב' יכול לפרק מידע של צד א' כי לשניהם יש מפתח.
2. הצד השולח יכול להתכחש ששלח.
3. הצד המקבל יכול להכחיש שקיבל.

**בהצפנה המסורתית / ההצפנה הסימטרית אין לנו מנגנון של אימות / מנגנון של חתימה.**

□ מפתחות : ציבורי ופרטי

הצפנה במפתח פומבי אינה מחליפה ממש את השיטה הישנה הסימטרית.

## **חסרונה הגדול = איטיות.**

נדרשים מספרים מאד גדולים. הצפנה כבדה.

שימושים: הפצת מפתחות.

חתימה אלקטרונית.

סודיות = הצפנה בציבורי + פענוח בפרטי.

שימוש בחתימה = מצפין בפרטי + פענוח בפומבי.

□ הצפנה במפתח פומבי מצריכה מפתח פומבי ומפתח פרטי.

□ במידה ונרצה סודיות : ההצפנה היא במפתח הציבורי והקידוד יהיה במפתח הפרטי.

□ אם נרצה סודיות חתימה : נצפין במפתח פרטי ונקודד במפתח הציבורי.

□ השיטה א- סימטרית כיוון שצד אחד יכול להצפין או לוודא חתימות אך לא לקודד הודעות או

ליצור חתימות.

□ **הערה:** לכל משתמש יש מפתח פומבי שידוע לכל המשתמשים , ומפתח פרטי שידוע רק לו.

השיטה:

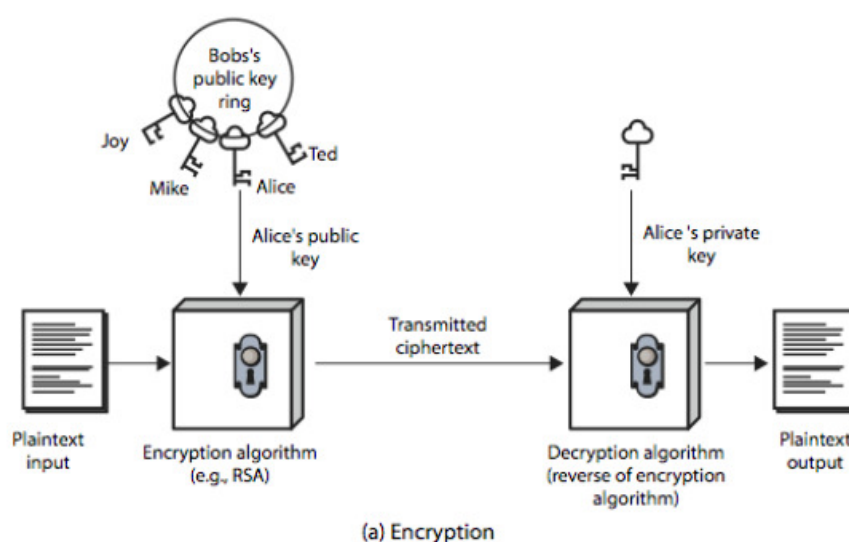
- (א) צד A מצפין הודעה עם  $PR_A$  ואח"כ מצפין שוב עם  $PU_B$
- (ב) צד B מקבל את הצופן ומפענח עם  $PR_B$  ואח"כ מפענח שוב עם  $PU_A$



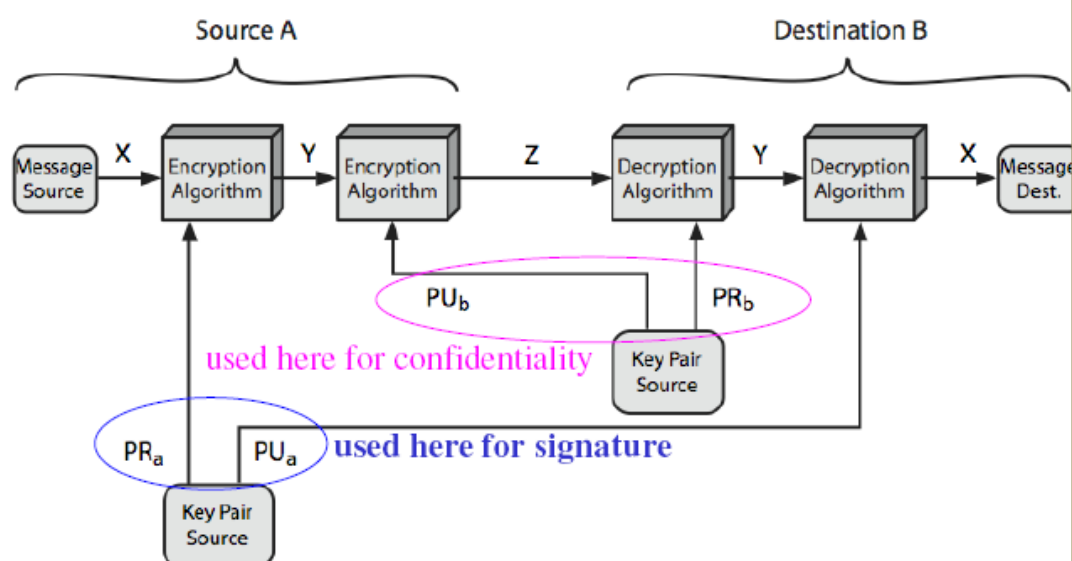
הצפנה עם מפתח פרטי כלשהו ואח"כ פענוח עם מפתח ציבורי של אותו צד מבטלת אחד את השני (כנ"ל גם ההיפך)

**מסקנה:**

## Public-Key Cryptography



## Public-Key Cryptosystems



### דרישות השיטה / מפרט:

1. לא מעשי מבחינת סיבוכיות למצוא את הפרטי גם אם ידועים המפתח הציבורי והאלגוריתם (שיטת ההצפנה).

2. קל להצפין ולפענח כשנתונים המפתחות המתאימים.

3. דו כיוונית = פרטי מפענח פומבי + פומבי מפענח פרטי

### שימושים:

1. הצפנה ופענוח

2. חתימה דיגיטלית.

3. החלפת מפתחות.

# Why RSA decryption works

- because of Euler's Theorem (part (b)):
  - $a^{1+k\Phi(n)} \bmod n = a$  , for any  $0 \leq a < n$
- in RSA we have:
  - $n=p \cdot q$
  - $\Phi(n)=(p-1)(q-1)$
  - we chose  $d$  to be inverse of  $e$  according to  $\bmod \Phi(n)$
  - hence  $e \cdot d=1+k \cdot \Phi(n)$  for some  $k$
- hence :
 
$$C^d \equiv M^{e \cdot d} \equiv M^{1+k \cdot \Phi(n)} \equiv M \pmod{n}$$
 by Euler's theorem, as stated above

$$a^{k\Phi(n)+1} \equiv a \bmod(n) \quad k \text{ is int}$$

$$a < n$$

מחפשים 2 מספרים המקיימים

$$ed = 1$$

$\Downarrow$

$$d = e^{-1}$$

מספר הפיך אם הוא זר

$$c = a^e \bmod(n) \quad \leftarrow a \quad \text{הצפנה:}$$

$$a^{ed} \equiv (a^e)^d \equiv c^d \bmod(n) \quad \leftarrow c \quad \text{פענוח:}$$

$$ed \equiv 1 \pmod{\Phi(n)} \Leftrightarrow ed = k\Phi(n) + 1$$

## תזכורת:

$$b = c \pmod{a} \Rightarrow b \pmod{a} = c \pmod{a}$$

$$b = k_1 a + r$$

$$c = k_2 a + r$$

$$c - b = (k_2 - k_1) a$$

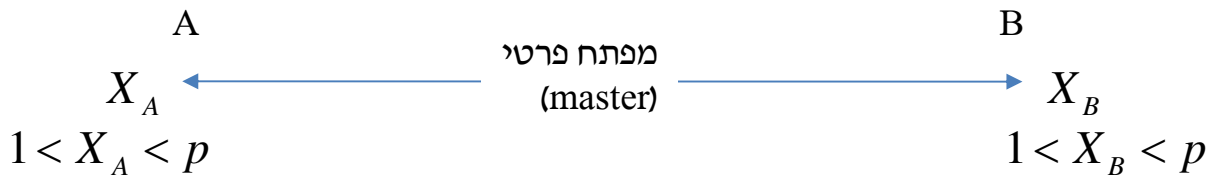
## דוגמא מספרית:

$$26 \equiv 1 \bmod 5$$

$$26 \equiv k5 + 1 = 5 * 5 + 1$$

$$31 \equiv 1 \pmod{5}$$

## DIFFIE HELLMAN



בוחרים P מספר ראשוני:

a שורש פרימיטיבי של P.

(שניהם ידועים למערכת).

שורש פרימיטיבי של P = יכול להגיע לכל המספרים ב-  $\mathbb{Z}_p$  פרט ל-0

$$Y_A = a^{X_A} \pmod{p} \longrightarrow$$

$$\longleftarrow Y_B = a^{X_B} \pmod{p}$$

סימטרי

$$K_s = (Y_B)^{X_A} = (a^{X_B})^{X_A} = a^{X_B X_A} = a^{X_A X_B} = (a^{X_A})^{X_B} = (Y_A)^{X_B}$$

**זוהי שיטה להסכים על מפתח מבלי לחשוף אותו.**

דוגמא:  $a = 3$      $p = 353$

$X_A = 97$      $X_B = 233$  : עתה, נבחר אקראית:

$Y_A = 3^{97} \pmod{353} = 40$  : נחשב:

$Y_B = 3^{233} \pmod{353} = 248$

**המספר הסודי המשותף לשניהם**

סימטרי

$$K_{AB} = Y_B^{X_A} \pmod{353} = 248^{97} = 160$$

$$K_{BA} = Y_A^{X_B} \pmod{353} = 40^{233} = 160$$

המשך חישוב:

**במצב בו 2 משתמשים מחליפים מפתחות diffie hellman בתקשורת, תוקף פעיל יכול לפעול כך:**

□ התוקף תופס את ההודעה A הכוללת את  $Y_A$  ולשלוח במקומה מספר אחר  $Y_E$  שיוצר על ידי התוקף ממספק  $X_E$  הנמצא ברשות התוקף בלבד.

□ התוקף עורך את ההודעה כך שייראה כאילו נשלחה ל-B ע"י A.

□ B ייצר מפתח סימטרי שגם התוקף E יכול ליצר וכך ישלח מידע סודי המיועד ל-A ומוצפן באופן

שייקרא ע"י E

## אימות וחתימה:

חתימה = אימות הצפנה.

אם צד A מצפין בעזרת המפתח הפרטי שלו,

צד B יפענח את ההודעה בעזרת המפתח הפומבי של A.

בצורה זו אנו מוודאים כי צד A חתם על ההודעה המוצפנת.

בעזרת ההצפנה: מטרת פונקציית hash ליצור מאפיין קטן ו"איכותי" לטקסט מסוים.

1. מוודאים שצד A חתם את ההודעה.  
2. אף אחד אחר לא חתם את ההודעה.

$$PU_A PR_A(m) = m$$

$$E(PR_B, M) = c \rightarrow D(PU_B, c) = m$$

$H = \text{hash function}$

$D = \text{digest}$

$$H(m) = D$$

$$M \parallel E(PR_B, H(M)) \longrightarrow D(PU_B, c) = m$$

$H'(m)$  - חישוב ישיר של צד B לפי הקידוד של A.  
 $H(m)$  - חישוב של צד A לפני שהוא שלח את M. אם הם שווים, אז בטוח שצד A קודד את M

והוא השולח.

ישימות לפונקציית hash:

1. בדיקת תקינות קובץ – שמירה של hash של המידע והשוואה שלו ל-hash של מידע קיים.
2. אימות זהות.

$$M, E_{ks}(H(m))$$



נותן אימות.  
לא נותן חתימה.

$$M, E_{PR_A}(H(m))$$



נותן אימות.  
נותן חתימה.

כאשר התוקף יודע את פונקציית הערבול, הדבר לא עוזר לא. זאת כיוון שמדובר בפונקציה שאינה ח.ח.ע.

חתימה = חד משמעית. רק מקור אחד יכול להוציא אותה.

אימות = ניתן לעקוף אותו ולהזדהות כמישהו אחר אם פורצים את המפתח.

דוגמאות לפונקציות HASH:

message digest, crypto, fingerprint, Checksum

המשתמשים A ו-B הגדירו מפתחות פרטיים ופומביים בשיטת הצפנה RSA. נסתכל בהצפנות הבאות:

$$E_{PU_A}(E_{PR_B}(M)) \quad (1)$$

רק B יכול להצפין, רק A יכול לפענח, השיטה מבטיחה סודיות.  
מטרה טיפוסית: העברה חסויה מ-B אל A עם חתימה דיגיטאלית.

$$E_{PR_A}(E_{PU_B}(M)) \quad (2)$$

רק A יכול להצפין, רק B יכול לפענח. השיטה מבטיחה סודיות.  
מטרה טיפוסית: העברה חסויה מ-A אל B עם חתימה דיגיטאלית.

$$E_{PU_A}(E_{PU_B}(M)) \quad (3)$$

כל אחד יכול להצפין, פענוח אפשרי רק על ידי שתוף פעולה של A עם B. השיטה מבטיחה סודיות.  
מטרה טיפוסית: הכנסת מידע למשמורת משותפת של שני נאמנים (למשל, שני עורכי דין) שרק בשיתוף פעולה מותר להם לראות את המידע (למשל מספר חשבון בנק חסוי). (Escrow)

$$E_{PR_A}(M) \quad (4)$$

רק A יכול להצפין, כל אחד יכול לפענח. השיטה אינה מבטיחה סודיות.  
מטרה טיפוסית: חתימה ללא הבטחת סודיות.

- (א) הסבירו מדוע נחוץ מרכז הפצת מפתחות כדי להפיץ מפתחות פומביים של משתמשי שיטת הצפנה במפתח פומבי.
- (ב) הסבירו כיצד שני משתמשים הרוצים ליצור קשר תקשורת מוגן ע"י מפתח פומבי רוכשים את המפתחות הנדרשים לשם כך.  
איזה הודעות נשלחות ומה כלול בכל אחת מהן?  
איזה תנאי קדם צריכים להתקיים כדי שהשיטה תעבוד?
- (ג) הסבירו את שיטת ה-Certificate להחלפה בטוחה של מפתחות. במה היא עדיפה על השיטה שתוארה בסעיף (ב)?

- (א) בעקרון מפתח פומבי מותר להפיץ לכולם ולכן אפשר היה שכל משתמש ישים את המפתח הפומבי בבסיס נתונים כללי.
- אלא שאז אי אפשר לדעת האם המפתח של A הוא אמיתי (כלומר הוכנס לשם ע"י A) או שהוא מושגל על ידי גורם עוין המעוניין לקלוט ולקרוא דואר ומידע אחר המיועד אל A.
- שימוש במרכז הפצת מפתחות (KDC) נועד להבטיח שהמפתח שאמור להיות של משתמש A הוא אכן שלו.
- (ב) (ג) ראה שקף 7 בפרק 10 והסברים בכיתה סביב שקף זה.



## התקפות:

❑ כאשר מעריכים את חוזקה של שיטת הצפנה מסוימת, יש להביא בחשבון מספר סוגי תקיפה:

❑ **Chipertext Only Attack** – לתוקף יש את המסר המוצפן בלבד.

❑ **Plaintext and Chipertext Attack** –

התוקף מחזיק במסר המקורי יחד עם ההצפנה של המסר וברצונו לגלות את המפתח.

(דוגמא: הודעה מדינית מגיעה לשגרירות באופן מוצפן ומפורסמת בכלי התקשורת לאחר מכן).

❑ **Chosen Plaintext Attack** –

התוקף משיג את הנוסח המוצפן של הטקסט (הצד התוקף משיג לזמן קצר את מכשיר ההצפנה)



❑ תקיפה אקטיבית = התוקף יכול גם לקרוא את הטקסט המוצפן וגם לשנותו.

❑ תקיפה פאסיבית = התוקף רק קורא את הטקסט המוצפן, מבלי להפריע או לשנותו.

❑ **BRUTE FORCE** – כוח גס הוא סוג של אלגוריתם שאין בו תחכום, כזה הפועל בדרך הפשוטה

ביותר להשגת המטרה, תוך שהוא צורך לעתים כמויות גדולות יחסית של משאבי מחשב (זמן או זיכרון). דוגמה לאלגוריתם מסוג זה הוא פיצוחן של סיסמאות כניסה למחשב באמצעות ניסיון להיכנס עם כל הסיסמאות האפשריות, עד להצלחה. טכניקה זו אינה יעילה, בדרך-כלל, מכיוון שמספר הצירופים האפשריים הוא עצום.

❑ **ניתוח חכם. אלגוריתם מתמטי.**

תקיפת REPLAY היא ש-M שומר הודעות ישנות ושולח אותן מחדש בזמן מתאים.  
תקיפה אפשרית אחת היא אם M גילה את המפתח הפרטי של A ובעקבות זאת A החליף את המפתח שלו.  
במצב זה יכול M לחטוף את ההודעה המבשרת ל-B את המפתח החדש של A ולשלוח במקומה את ההודעה הישנה הנושאת את המפתח הישן.  
באופן זה M יוכל לקרוא את המידע הסודי המיועד אל A שתי שיטות למניעת REPLAY:  
1. להכניס חתימות שעון בהודעה המוצפנת יחד עם המפתח ששולחים  
2. B ישלח אל A NONCE בלתי צפוי (בגלוי) ואז A יכלול בהודעה המוצפנת פונקציה ידועה של ה-NONCE.