

- אבטחת מידע

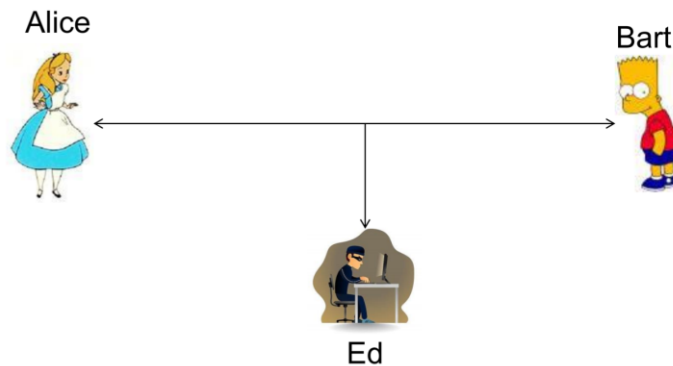
באמצעות אבטחת מידע נרצה להשיג את המטרות הבאות: נרצה לוודא שאין באגים בתוכנה (באג יכול להוות מקור לפריצה), שנתונים שנשלחים יהיו סודיים, למנוע כניסת זרים למערכת, למנוע האזנה לסודות עסקיים, לוודא את נכונות המידע ונרצה לייצר חוזים שהם ברי אכיפה, כלומר במקרה של הזמנה פריט און ליין, למשל במקרה של כתובת מייל ככתובת מבצע ההזמנה.

- דוגמה – דרכון אלקטרוני מוסיף קשיים מהבחינה שבעולם אלקטרוני אין מושג של גורם מקור, להבדיל מהעולם הפיזי בו ניתן להגדיר מסמך מסוים כמסמך המקור, בעולם אלקטרוני כאמור לא ניתן לבצע זאת, ולכן ההמצאה של תחליף אלקטרוני למסמך פיזי היא מאוד קשה. בכדי ליישם דרכון אלקטרוני צריך מאגר תמונות, מאגרי נתונים פנימיים (למדינה) ובינ"ל, כאשר האימות יהיה אלקטרוני ובעת מעבר גבול יתבצע רישום, ולכן בעקבות הסיבוכיות שבדבר, טרם קיימים דרכונים אלקטרוניים.

לא כל איום הוא בר תיקון, אך באמצעות מנגנונים שנבנים, ניתן לזהות איומים ולמזער אותם.

- Kinds of Cryptographic Analysis

- 1- התקפת טקסט מוצפן בלבד (Ciphertext Only Attacks) (*) – סוג התקפה שבה מניחים שלתוקף יש גישה לטקסטים מוצפנים בלבד. לתוקף אין גישה לטקסטים גלויים ולכן תקיפה זו הינה הקשה ביותר עבור התוקף.
- 2- התקפת טקסט גלוי ידוע (Known Plaintext Attacks) (*) – סוג התקפה שבה לתוקף יש גישה לחלק מהטקסטים הגלויים ולטקסטים המוצפנים בהתאמה. כלומר, קבוצת זוגות של טקסטים גלויים ומוצפנים בהתאמה.
- 3- התקפת טקסט גלוי ניתן לבחירה (Chosen Plaintext Attacks) (**) – סוג התקפה שבה מניחים שהתוקף יכול לבחור אקראית טקסטים גלויים, להצפין אותם, ולקבל את הטקסטים המוצפנים התואמים. התקפה מסוג זה חשובה בעיקר בהקשר של הצפנה עם מפתח ציבורי, שבה מפתח ההצפנה הינו ציבורי והתוקף יכול להצפין כל טקסט שהוא בוחר.
- 4- התקפת טקסט מוצפן ניתן לבחירה (Chosen Ciphertext Attack) (**) – סוג התקפה שבה התוקף יכול לבחור טקסט מוצפן כרצונו, לפענח אותו (כאשר הוא לא יודע את המפתח) ולקבל את הטקסט הגלוי בהתאמה



* שיטות פאסיביות, המאזין לקו אינו פעיל

** שיטה אקטיבית, המאזין לקו פעיל

- Beginners Cryptography and History
 - Alphabetic Substitution Ciphers
 - The Caesar Cipher (צופן הקיסר)
 - Transposition (הזזה)
 - One Time Pads
- Computational Security

- Beginners Cryptography and History
 - Alphabetic Substitution Ciphers

- The Caesar Cipher (צופן הקיסר) – צופן שבו ממירים כל אות לאות אחרת, מודל בסיסי של הצפנה של כל סמל בנפרד, מוסיפים מספר קבוע כלשהו לאות ולאחר מכן מבצעים פעולות מודולו בכדי להישאר בטווח הסמלים. נקודת חולשה קריטית בצופן זה, היא, שברגע שמגלים אות אחת ניתן לדעת את כולן.
- דרכים לפריצה: 1- ניחוש כל האפשרויות. 2- ניחוש המפתח על סמך תדירויות האותיות בכתב המוצפן בהשוואה לתדירויות האותיות בשפה הטבעית.

```
abc...xyz
|||  |||
def...abc
```

- Monoalphabetic Ciphers – הרחבה של צופן הקיסר, מיפוי אקראי של כל אות לאות אחרת כלשהי (ע"פ פרמוטציה π שמגדירים).

```

a b c d e f g h i j k l ...
π z d a n c e w i b f g h ...
```

- Vigenère Tableau (השולחן של וינייה) – סוג של מחשבון להצפנה באמצעות פרמוטציה, מעין הרכבה בין פרמוטציה לקיסר. הרעיון היה לקחת מילת סוד (מפתח, כמה שיותר ארוך יותר טוב) ולהפריד את המפתח מהאלגוריתם. באמצעות המפתח מבצעים הצפנה, לכל אות מוסיפים את האות המקבילה ממילת הסוד המצפינה.

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	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

"שולחן וינייה" – אוסף של 26 פרמוטציות בגרید של 26×26 אותיות

- Diffusion and Confusion

- Diffusion (דיפוזיה – פעפוע) – כאשר מידע הממופה אחד לאחד (ביט לביט, סמל לסמל) אין פעפוע, וכאשר יש יותר פעפוע, יותר קשה לתקוף את הצופן, למעשה כאשר מדובר בהצפנה של 1:1 אז אין פעפוע ולכן זה פחות טוב. כאשר סמל אחד משפיע על יותר מסמל אחד, או שמיקום הסמל משפיע על התוצאה, אז יש יותר פעפוע (בדומה לוינייה), ולכן קשה יותר לתקוף את הצופן, שזה דבר רצוי.
 - Confusion (בלבול) – הגדרה מתמטית למורכבות הצפנה, אם הצעדים הם פשוטים יש מעט צעדים ואין בלבול, למשל בהצפנת הקיסר אין בלבול, כאשר מס' הצעדים גדל כך גם הבלבול, ובלבול עוזר להתגונן מפני התקפה.
- צופן עם פעפוע ובלבול יותר קשה לפענח ולעקוב אחר הצעדים

- Transposition (הזזה) – שינוי מיקום האותיות בטקסט. אם המפתח או בגודל 4 אז מזיזים את כל האותיות 4 מקומות מהמיקום המקורי שלהן. לבד כצופן זה לא כזה חזק, אך כאלמנט, הזזה מחזקת את הצופן, ולכן היא נחשבת כאחד המרכיבים של צופן חזק.

- One Time Pads –

- Perfect Substitution Ciphers – בעולם אבטחת המידע יש רק צופן אחד "מושלם" ונוכל להוכיח זאת בצורה פורמאלית. השאלה בעצם היא מהו צופן "מושלם"? ואיך נדע האם צופן נתון הוא אכן כזה? באופן כללי, לכל צופן בעולם הוא בעייתי, מכיוון שגם צופן גדול ביותר 2^{256} אפשר לנחש מתישהו, לכן הצופן "המושלם" יהיה כאשר לא יהיה באמת גורם שיוכל לענות האם התשובה נכונה או לא, צופן כזה שאפילו אם מגלים את התשובה, לא יודעים להגיד אם זו אכן התשובה. ניתן להגיד כי אם הודעה היא בת 10 אותיות ומפתח ההצפנה בעל מס' אותיות זהה, אזי הצופן יהיה מושלם מכיוון שכל אות יכולה להיות אחת מ-10 אותיות במפתח ההצפנה. יחד עם זאת, אסור לבצע שימוש חוזר במפתח מכיוון שצופן חזק רק בזכות העובדה שאין חזרתיות, והמפתח הוא כאורך ההודעה, ולכן כל קלט יכול להיות פלט.

כל מילה (!) שנשלחת צריכה להישלח עם מפתח אחר

לא יישומי למחשבים בגלל המורכבות הלוגיסטית הקיימת. כל פעם צריכים להבין עם איזה מפתח הצפינו מכיוון שהוא משתנה.

• Computational Security

צופן נחשב מאובטח אם לוקח המון (!) זמן לדעת מה התשובה הנכונה. באופן כללי, הגישה באבטחה היא שידוע שצופן נתון הוא לא מושלם, אך מניחים שההתקפה היחידה שניתן לבצע היא Brute Force Attack, ובהסתמך על ההנחה הזו, ניתן לשער כמה זמן יכול לקחת חישוב של כל האפשרויות (לעיתים יכול לקחת מיליוני שנים). * חשוב לזכור – אם יש דרך לעשות התקפה יותר מהירה מ-Brute Force Attack, אז הצופן למעשה פגיע ומכיל קיצורי דרך או מלכודות (Shortcuts and Trapdoors), כלומר עצם העובדה שצופן ניתן לפריצה מלבד Brute Force Attack, זה אומר שהוא מכיל בעיה כלשהי שיכולה להוות מקור לפריצה. ניקח לדוגמה צופן בעל n סיביות, אז מספר האפשרויות למפתח הוא $O(2^n)$, וממוצע הניחוש הוא מחצית.

למעשה ההנחה הטובה לצופן טוב הוא שניתן לפרוץ אותו רק באמצעות brute force בלבד (!) ואך ורק ע"י מחשבים

- **Computational Security**
- **Modern Cryptographic**
 - Stream Ciphers
 - Block Ciphers
 - Kinds of Industrial Strength Crypto
- **Shared Key Encryption**
 - DES

• **Computational Security**

- **מדדי אי יכולת אבחנה** - מדדים מודרניים שמגדירים מהו אלגוריתם הצפנה (cipher) טוב. הוכח שביטחון סמנטי שקול להגדרת ביטחון מערכת הצפנה שנקראת "אי-יכולת הבחנה" (indistinguishability) המסומנת בקיצור IND, שפירושה שהיריב לא יכול להבחין בין טקסטים מוצפנים של המערכת לבין טקסט אקראי אמיתי בהסתברות גבוהה מחצי בשיעור שאינו זניח. אפשר לחלק את ההגדרה של ביטחון סמנטי לכמה דרגות בהתאם ליכולות היריב מהקל לכבד. ההבדל ביניהם הוא שבראשונה היריב מקבל תוצאות הצפנה של טקסטים קריאים אותם.

IND-CPA	IND-CPA1	IND-CPA2
יש לנו 2 הודעות באותו האורך, אחת מהן עוברת הצפנה. אם יש סיכוי יותר מסטטיסטי (מעל 50%) לנחש אילו מבין ההודעות היא זו שהוצפנה, אז נאמר שהצופן הוא לא מספיק טוב.	התקפת גלוי-נבחר המסומנת IND-CPA 1 - קיצור של Indistinguishability Chosen Plaintext Attack. היא סוג של התקפה שבה מניחים שהמתקיף יכול לבחור כל טקסט מקור שהוא מעוניין לקבל עבורו את הטקסט המוצפן המתאים. אך הוא יכול לראות את הטקסט המוצפן רק לאחר שסיים להגיש את כל השאלות. מרגע שהוא רואה את הטקסט המוצפן, הוא אינו יכול להגיש בקשות נוספות.	התקפת גלוי נבחר אדפטיבית המסומנת IND-CPA 2. היא התקפה חזקה יותר שבה המתקיף מסוגל להתאים את בקשותיו בהתאם לתוצאות קודמות, כלומר ביכולתו לבקש טקסט מוצפן גם לאחר שראה טקסט מוצפן שביקש קודם.

הדרגה השלישית IND-CCA2 מכילה את 2 הקודמות והיא הכי מחמירה

IND-CCA2 → IND-CCA1 → IND-CPA

מהמדד החמור ביותר (במובן של פריצה) לקל ביותר

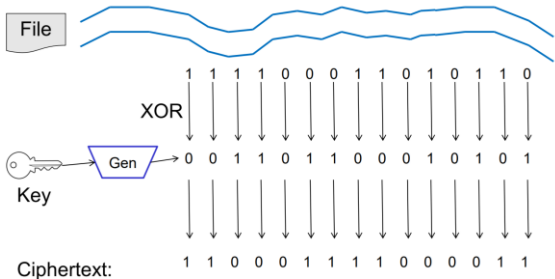
מקורות:

[ביטחון סמנטי](#)

[התקפת גלוי-נבחר](#)

• **Modern Cryptographic**

Block Ciphers	Stream Ciphers	
צופן בלוקים מקבל בלוק של סיביות Plaintext (טקסט גלוי) ומפתח הצפנה סודי ומפיק Ciphertext (בלוק טקסט מוצפן), כאשר תוצאת הטרנספורמציה (התאמת איבר מקבוצה אחת לקבוצה שנייה) נקבעת ע"י מפתח ההצפנה. פעולת הפענוח מתבצעת באופן זהה, אלגוריתם הפענוח מקבל Ciphertext (בלוק סיביות טקסט-מוצפן) והמפתח שאיתו הוצפן ומחזיר את ה-Plaintext (בלוק הסיביות המקורי)	סוג של צופן סימטרי שמצפין זרם באורך משתנה של יחידות מידע (המיוצגות ע"י סיביות, בתים או מילים), תוך שימוש בטרנספורמציה (התאמה של כל איבר בקבוצה אחת לאיבר בקבוצה שנייה) המייצרת מפתח לפי "מצב פנימי" (internal state). לקבלת הטקסט המוצפן זרם המפתח מחובר עם הקלט הקריא בחיבור בינארי (XOR) או בפעולה אחרת ובלבד שתהיה הפיכה.	תיאור כללי
<ul style="list-style-type: none"> ההצפנה מתבצעת ברמה של אוסף של ביטים. קיים פעפוע מכיוון שאם ביט אחד משובש, כל הבלוק משובש. 	<ul style="list-style-type: none"> ההצפנה מתבצעת ברמה של ביט-ביט ולכן מהירה יותר. לא קיים פעפוע (Diffusion) מכיוון שמצפינים ברמה של ביט-ביט ולכן אין השפעה של ביט אחד על יתר הביטים. 	מאפיינים

<p>▪ מתאים לשימוש בזרימת וידאו.</p>		
<p>דוגמא</p> 		
<p>▪ יתרון – יותר פעפוע, לא ניתן לעשות שינויים נקודתיים בתוך הטקסט אלא שינויים בגודל בלוק, כך למעשה מקשים על שינויים זדוניים במידע.</p> <p>▪ חסרון – באופן כללי הצופנים איטיים יותר מכיוון שנדרשת הצפנה של בלוק שלם, ויש הגבלה למינימום גודל הבלוק, ולכן כאשר מבצעים הצפנה של בלוק שאינו מתחלק בגודל המידע צריך לבצע padding.</p>	<p>▪ יתרון – צופני זרימה נמצאים בשימוש בהזרמת וידאו ובהזרמת נתונים ב-Wi-Fi, בשני המקרים לא רוצים לדעת מראש מה גודל ההודעה, יותר קלילים במובן הזה.</p> <p>קל יותר לתקן ולכן הנזק שנגרם כתוצאה מכוונה זדונית יותר קטן, מכיוון שניתן להתאושש בקלות משגיאות נקודתיות, להבדיל משגיאה בסדר גודל של בלוק.</p> <p>▪ חסרון – החיסרון הוא שניתן לבצע שינויים במזיד בהצפנה, אין פעפוע, ניתן לעשות שינויים בכוונה על הstream.</p>	<p>יתרונות/חסרונות</p>
<p>ההבחנה העיקרית בין צופן זרם לצופן בלוקים היא אופן יצירת מפתח ההצפנה. בהגדרת צופן זרם מתכוונים למחולל הפנימי (keystream generator) המייצר זרם-המפתח פסידו-אקראי. המחולל מתבסס על זיכרון פנימי ופועל בצורה של פעימות (clocking), מסיבה זו אומרים שלצופן זרם יש 'זיכרון', זאת בניגוד לצופן בלוקים שכאשר הוא מופעל בצורה ישירה, אינו מכיל זיכרון כלל</p>		

Kinds of Industrial Strength Crypto

Cryptographic Hashes	Public Key Cryptography	Shared Key Cryptography
<p>פונקציית גיבוב קריפטוגרפית שלוקחת קלט גדול ומוציאה פלט בגודל קבוע שנגזר ממנו, נועדה לסכם מידע ברמה גבוהה</p>	<p>מפתח ציבורי להצפנה (שניתן לפרסם לכולם) ומפתח אחר סודי עבור הפענוח.</p> <p>האתגר במפתח משותף הוא שלכאורה תמיד קיים גבול ליכולת הגנה על הנתונים, ולכן מניחים שמישהו יעשה טעות והמפתח ידלוף החוצה, ולכן כל פרק זמן מסוים חייבים להחליף מפתח, כעת השאלה הנשאלת היא כיצד מפיצים מפתח חדש, ומספר המפתחות האפשרי הוא $O(n^2)$ כאשר $n = \text{מס' המשתתפים בשיחה}$</p>	<p>הצפנה סימטרית, עם מפתח משותף, בהם שני הצדדים צריכים להחזיק את אותו המפתח כדי לדבר, ואז אחד מצפין ואחד מפענח</p>
<p>* במפתח ציבורי המפתח האדום הוא המצפין והמפתח הצהוב הוא הפותח, ניתן להניח שלכולם יש את המפתח האדום את לא את המפתח הצהוב.</p>		

כולם עובדים תחת Computational Security וניתן להפעיל עליהם Brute force

SSL למשל בנוי משלושתם, הוא פרוטוקול. עובד עם כולם כדי להגיע למצב שאם גולשים לגוגל הוא עובד, כלומר מגיעים לגוגל והשיחה מאובטחת אפילו שגוגל לא יודעים מי אני

• **Shared Key Encryption**

- **DES (Data Encrypted Standard)** – צופן בלוקים סימטרי שיועד בעיקר לחומרה, הוא מקבל בלוק טקסט קריא

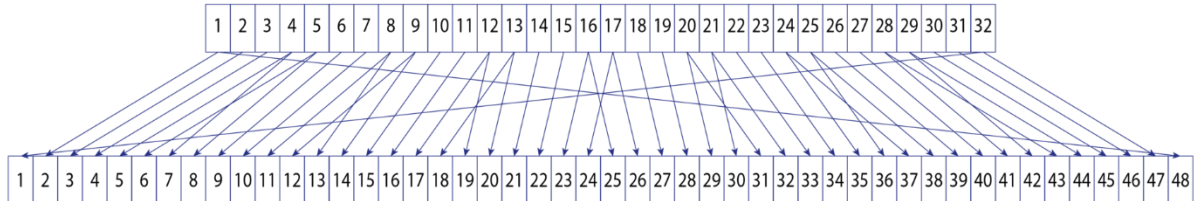
שהוא מחרוזת של 64 סיביות (8 בתים) ומפיק בלוק מוצפן באורך זהה. הפונקציה הפנימית של הצופן מתבצעת ב-16 חזרות בעזרת מפתח סודי באורך 64 סיביות (שמתוכן רק 56 סיביות בשימוש).

אופן הפעולה – הקלט הוא בלוק טקסט קריא (Plaintext) בגודל 64 סיביות ומפתח הצפנה סודי K בגודל 64 (בפועל 56)

סיביות. באמצעות תהליך הכנת המפתח, מרחיבים את המפתח ומחלקים אותו ל-16 מקטעים בני 48 סיביות כל אחד.

כאשר כל מקטע משמש עבור סבב נוסף. מכינים 8 תיבות החלפה S_1, S_2, \dots, S_8 , הן מייצגות אוסף

של פונקציות לא ליניאריות שמקבלות קלט באורך 6 סיביות ומחזירות פלט באורך 4 סיביות בתאם לטבלת הערכים הקבועה המוצגת להלן:



תיבת הרחבה E היא מעין טבלת העתקה/סידור מחדש, בה מעתיקים סיביות מהשורה העליונה לשורה התחתונה לפי החצים, למשל הסיבית הראשונה בקלט מועתקת לשני מקומות, לסיבית השנייה וה-48 בפלט ואילו למיקום הראשון בפלט מעתיקים את הסיבית האחרונה בקלט שאותה מעתיקים גם למיקום אחד לפני אחרון בפלט.

מכינים "תיבת הרחבה" ו"תיבת תמורה", המייצגות פונקציות מיפוי לפי ערכים קבועים. התיבה E (קיצור של Expansion)

היא תיבת תמורה/הרחבה, שתפקידה להכין את הקלט להחלפה בתיבות ההחלפה. היא מרחיבה את הקלט מ-32 סיביות

ל-48 סיביות. 16 מתוך 32 סיביות הקלט מועתקות פעמיים למקומות שונים כמתואר בתרשים. למשל הסיבית במיקום

הרביעי בקלט מועתקת למיקומים החמישי והשביעי בפלט. מכינים תיבת תמורה קבועה הנקראת P-box (קיצור של

Permutation) שמשנה את סדר הסיביות של הפלט בסוף כל סבב תוך שמירה על גודלו.

- **DES S-Box** – טבלת המרה, 6 ביטים בקלט מומרים ל-4 ביטים בפלט, כל המרה מתבצעת לפי הקלט. הטבלה ממומשת

כטבלת חיפוש, ישנם 8 S-Box'ים, כל S-Box היא טבלה המורכבת מ-64 כניסות וכל כניסה מורכבת מ-4 ביטים (לשים

לב לשתי הדוגמאות, בטבלה הראשונה מופיעים מספרים רגילים אבל הם מיוצגים בבינארי כפי שניתן לראות בטבלה השנייה).

סדר	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
S_2	15	7	3	14	2	13	1	10	6	12	11	9	5	3	8	4
S_3	1	14	8	13	6	2	11	15	12	9	7	10	5	0	4	3
S_4	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
S_5	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
S_6	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
S_7	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_8	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
S_9	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
S_{10}	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14
S_{11}	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2
S_{12}	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
S_{13}	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14
S_{14}	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8
S_{15}	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2
S_{16}	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
S_{17}	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8
S_{18}	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0
S_{19}	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5
S_{20}	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5
S_{21}	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3
S_{22}	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11
S_{23}	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8
S_{24}	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6
S_{25}	1	13	0	11	7	4	9	1	10	14	3	5	12	15	8	6
S_{26}	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9
S_{27}	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3
S_{28}	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12
S_{29}	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9
S_{30}	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5
S_{31}	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6

- **DES S-Box Example**

S_5		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

• **Shared Key Encryption**

- DES Summary and Problems
- Triple-DES (3DES)
- AES

• **Block Cipher Models and Attacks**

• **Shared Key Encryption**

- **DES Summary and Problems** – לשיטת DES 2 חסרונות בולטים
 1. אורך הבלוק הוא 64 סיביות (8 בתים). לפי פרדוקס יום ההולדת לאחר הצפנת כארבעה מיליארד בלוקים, ההסתברות ששני בלוקים יוצפנו באמצעות אותו וקטור אתחול גבוהה במיוחד. אפשרות כזו היא בעייתית, ולכן משתמשי DES נאלצים להחליף מפתחות הרבה קודם. בעיה זו נפתרת רק באמצעות הגדלת הבלוק, כפי שנעשה ב-AES.
 2. אורך המפתח הוא 56 סיביות בלבד. המשמעות היא שאם מנסים את כל 72,057,594,037,927,936 האפשרויות, בוודאי ימצא המפתח הנכון. שיטת פריצה זו המכונה "כוח גס" (Brute Force) לא הייתה פרקטית בעבר. כשאלגוריתם DES הוצג לראשונה בשנת 1976 פריצתו הייתה הרבה מעבר ליכולת הטכנולוגית באותה תקופה. אולם כיום ניתן לבנות מחשבים ייעודיים מרובי מעבדים, בעלות של כמה עשרות אלפי דולרים, איתם אפשר לגלות מפתח DES בתוך מספר דקות. בנוסף:
 - אורך המפתח קטן מידי.
 - **Weak Keys** – קיימים 4 מפתחות שידועים שהם לא טובים בכך שהם לא מצפינים (לא מצפינים כלל).
 - **Semi Weak** – זוגות, אם מצפינים בעזרת מפתח אחד אז השני מפענח, נקודה בעייתית כאשר בוחרים מפתחות בצורה אקראית.
- **Triple-DES (3DES)**

צופן 3DES נועד להתגבר על הבעיה השנייה של DES. הצופן פועל עם שלושה מפתחות DES ולכן אורך המפתח הכולל הוא 168 סיביות. אורך זה מונע את האפשרות לפיצוח הצופן על ידי ניסוי כל המפתחות האפשריים (במגבלות כוח המחשוב בימינו).
- **AES (Advanced Encryption Standard)** – בשמו המקורי נקרא ריינדל, צופן בלוקים סימטרי איטרטיבי, עם בלוק ומפתח משתנים בגודלם, ניתן להגדירם ללא תלות אחד בשני; 128, 192 או 256 סיביות (בתקן הצפנה מתקדם גודל הבלוק נקבע ל-128. באופן כללי, AES שונה מ-DES מהבחינות הבאות:
 - 1- AES יכול לקבל מפתחות בגדלים משתנים; 128, 192 או 256 (לעומת DES שיכול לקבל רק גודל אחד).
 - 2- AES עובד במס' משתנה של סבבים לעומת DES שעובד במס' קבוע (16).
 - 3- בשונה מ-DES, AES עובד על כל ההודעה בפעם אחת.

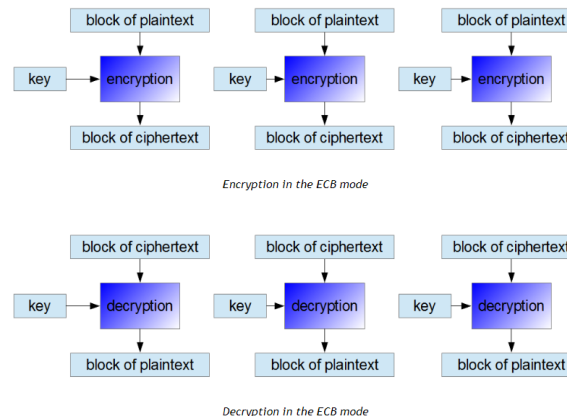
• **Block Cipher Models and Attacks**

- בקריפטוגרפיה "אופן הפעלה" (Mode Of Operation) מתייחס לאלגוריתם שמגדיר כיצד להפעיל צופן בלוקים סימטרי דטרמיניסטי להצפנת Plaintext (טקסט גלוי) באורך שעולה על הבלוק שהצופן מסוגל להצפין בבת אחת. צופן בלוקים כשלעצמו מסוגל רק להבטיח סודיות של בלוק טקסט גלוי, שהוא מחרוזת סיביות באורך קבוע מוגדר מראש. מעצם ההגדרה, בהינתן צופן בלוקים דטרמיניסטי כמו AES ומפתח הצפנה זהה, הצופן ימיר תמיד בלוק קלט זהה לבלוק Ciphertext (טקסט מוצפן) זהה. לכן ניתן לזהות הצפנה חוזרת של בלוקים זהים עם אותו מפתח. זהו מאפיין שעלול להוות חיסרון כי בהצפנה ישירה דולף בדרך אגב מידע ליריב פוטנציאלי שעלול להיות מנוצל להתקפה נגד המערכת.
- **ECB (Electronic CodeBook)** – באופן הפעלה (Mode of Operation) זה מחלקים את המסר לבלוקים בגודל n סיביות, מרפדים את הבלוק האחרון בשיטת ריפוד מוסכמת אם אורך המסר אינו מתחלק בדיוק ב-n ופשוט מצפינים כל בלוק בנפרד. אופן זה הוא תהליך דטרמיניסטי ולכן אינו נחשב בטוח לפי מודל התקפת גלוי-נבחר. יתרה מזו, אופן הפעלה זה אינו מכיל

את התכונה היסודית של ההצפנה שנקראת "אי יכולת הבחנה" שפירושה שיריב בעל עוצמת מחשוב מוגבלת לא יוכל לזהות הבדל בין טקסט-מוצפן שהוא תוצאה של ההצפנה באופן ECB לבין מחרוזת טקסט אקראית באותו אורך. הסיבה לכך היא שאם מצפינים בלוק מסוים פעמיים עם אותו מפתח מתקבלים בלוקים מוצפנים זהים, עובדה שכל מתקיף יכול להבחין בה בקלות. יש לציין שזו אינה רק בעיה תאורטית, אלא בעיה מעשית לכל דבר, ניתן ללמוד הרבה על הצופן רק מהעובדה שבלוקים מסוימים שהתקבלו הם זהים. מסיבה זו אופן ההפעלה ECB אינו נמצא בשימוש כמעט בכלל.

- תכונות ECB

- 1- **ביטחון** – בלוקים של Plaintext (טקסט גלוי) מפיקים תמיד בלוקים של Ciphertext (טקסט מוצפן) זהים, כל עוד מפתח ההצפנה זהה. לכן קל לזהות שהוצפנו בלוקים זהים.
- 2- **כשל מצטבר** – שגיאה בסיבית אחת או יותר של בלוק Ciphertext (טקסט מוצפן) משבשת את פענוח בלוק Plaintext (טקסט גלוי) המתאים בלבד. יתר הבלוקים אינם משתבשים. פענוח של בלוק המכיל סיבית שגויה אחת יהיה אקראי, דהיינו בממוצע 50 אחוז מסיביות הבלוק יהיו שגויות.
- 3- **תלות הדדית** – הבלוקים מוצפנים בנפרד ללא תלות זה בזה. שינוי סדר הבלוקים המוצפנים גורר אחריו לאחר פענוח שינוי בסדר הבלוקים הגלויים בהתאם. סידור מחדש או הסרה זדונית של בלוקים שלמים עלול שלא להתגלות.

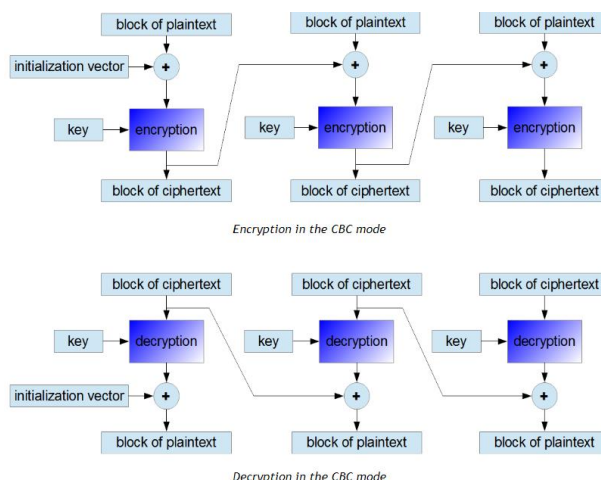


- IV - **CBC (Cipher Block Chaining)** – באופן הפעלה Cipher-Block Chaining (שרשר בלוקים מוצפנים), מוסיף ווקטור אתחול IV באורך n סיביות להצפנה. במצב זה כל בלוק Plaintext (טקסט קריא) מחובר לפני שהוא מוצפן בחיבור XOR עם תוצאת הצפנת הבלוק הקודם. וקטור האתחול הוא ערך אקראי חד-פעמי. הוא אינו סודי ואינו נחשב למפתח ההצפנה. כדי שהמקבל יצליח לפענח את הטקסט המוצפן הוא צריך לקבל את וקטור האתחול במצב גלוי, יחד עם הטקסט המוצפן.

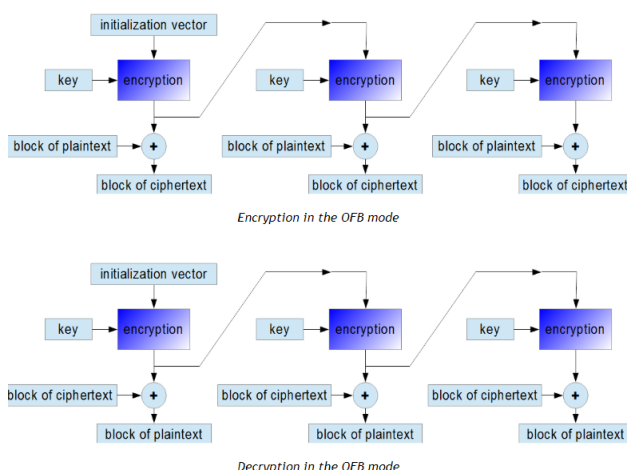
- תכונות CBC

- 1- **ביטחון** – תהליך זה הסתברותי בהנחה שווקטור האתחול אקראי, כי תוצאת הצפנת כל בלוק גלוי תהיה גם בטקסט המוצפן קודם והבלוק הראשון תלוי בווקטור האתחול. לכן במצב ששני בלוקים זהים הוצפנו עם אותו מפתח יתקבלו בלוקים מוצפנים שונים.
- 2- **התנפחות** – הטקסט המוצפן מתרחב בבלוק נוסף שהוא וקטור האתחול שצריך להישלח גם הוא לצד המקבל. הביטחון המוכח האמור בסעיף הקודם נכון רק אם וקטור האתחול נבחר באקראי. וקטור האתחול יכול להיות מספר סידורי. ואז השולח אינו חייב לשלוח את וקטור האתחול כי שני הצדדים יכולים להתחיל מ-1 ולקדם את ערכו אחרי כל הצפנה. אך יש לזכור שבמקרה זה אופן CBC לא יהיה בטוח לפי מודל התקפת גלוי נבחר (Chosen Plaintext Attacks).
- 3- **שרשור** – בהינתן בלוק Plaintext (טקסט גלוי), מפתח ההצפנה וגם וקטור האתחול זהים, יופק תמיד בלוק מוצפן זהה. לכן חובה לבחור וקטור אתחול חדש בכל הצפנה..
- 4- **תלות הדדית** – מנגנון השרשור של CBC גורם לתלות של כל בלוק צופן בבלוק שלפניו. סידור מחדש של בלוקים עלול להשפיע ישירות על תוצאות הפענוח של אותם בלוקים ועל כן יתגלה.
- 5- **כשל מצטבר** – סיבית אחת או יותר שגויים בבלוק כלשהו, עשויים לגרום לתוצאות פענוח שגויות בבלוק הנוכחי ובבלוק הבא אחריו. יתרה מזו פענוח הבלוק הנוכחי יהיה אקראי (50% שגיאה) ובבלוק הבא אחריו תופיע השגיאה בדיוק באותן סיביות בהן ארעה בבלוק הקודם. לכן במחיר של איבוד בלוק אחד (הבלוק הראשון) היריב מסוגל לבצע שינויים זדוניים בסיביות מסוימות של הבלוק הבא אחריו מבלי שיתגלו.
- 6- **התאוששות עצמית** – שיטת CBC בעלת מנגנון סנכרון עצמי, במובן שבמקרה שגיאה אפילו כגון אם בלוק של או יותר מתוך הצופן שגויים או נעדרים לגמרי, רק הבלוק הבא לאחריהם לא יפוענח כראוי, כל היתר לא יפגעו. אולם יש לשים לב שבמקרה של כשל בקריאה או חוסר של סיביות במספר שאינו בגודל הבלוק עשוי לגרום לפריצת גבולות הבלוק ולכן ההתאוששות תהיה בלתי אפשרית בבלוקים הבאים.

- 7- גישה אקראית – אופן הפעלה CBC אינו מתאים במקרה שדרושה גישה ישירה, כיוון ששיגאה בבלוק אחד בזמן הצפנה תגרור שיגאה בכל הבלוקים הבאים אחריו.
- 8- מקביליות – היות שכל בלוק תלוי בקודמו CBC אינו מתאים ליישום מקבילי. מאותה סיבה כאשר יש צורך לשנות רק חלק מהבלוקים צריך להציפין מחדש את כולם.

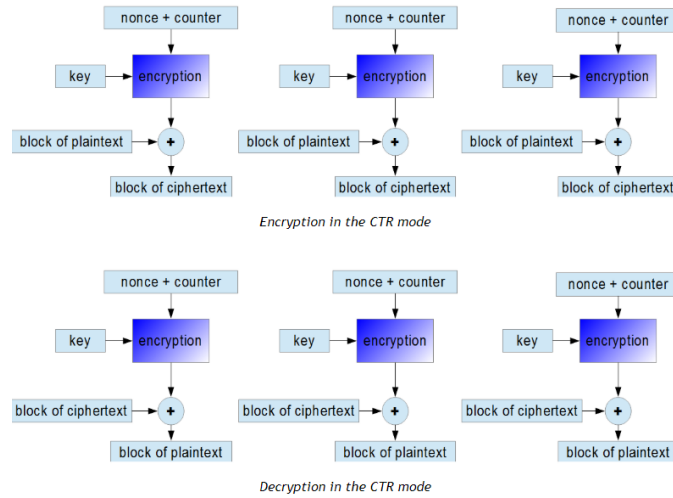


- OFB (Output Feedback) – אופן ההפעלה Output Feedback (משוּב פלט) מדמה צופן זרם והוא שמיש במקרים בהם יש צורך להציפין זרם מידע באורך לא ידוע וכן יש צורך להימנע מכשל מצטבר. OFB מאפשר הצפנת בלוקים בכל גודל רצוי.
- תכונות OFB
- 1- ביטחון – OFB מוגדר כבטוח לפי מודל התקפת גלוי נבחר (Chosen Plaintext Attacks) אם ווקטור האתחול אקראי.
 - 2- מקביליות – אופן זה סדרתי באופיו ולכן אינו מתאים ליישום מקבילי.
 - 3- שרשרון – בדומה לאופן CBC, וקטור האתחול אחראי לכך שבלוקי מסר (Plaintext) זהים יוצפנו לבלוקי צופן (Ciphertext) שונים.
 - 4- תלות הדדית – זרם המפתח עצמאי ואינו תלוי בטקסט גלוי (Plaintext)
 - 5- כשל מצטבר – סיבית אחת או יותר בבלוק צופן נתון, ישפיעו אך ורק על אותן סיביות בבלוק הגלוי המתאים. כאשר סיביות אלו בבלוק המפוענח יהפכו למשלמים של סיביות המקוריות באותם המקומות.
 - 6- התאוששות עצמית – אופן OFB מסוגל להתאושש ממצב של סיביות שגויות אך לא ממצב של חוסר סיביות.



- CTR (Counter Mode) – אופן מונה (Counter Mode) בדומה לאופן OFB, משנה את התנהגות צופן הבלוקים לצופן זרם ע"י ייצור זרם מפתח מתוך המונה. בניגוד ל-OFB במצב זה מצפינים את המונה פעם אחר פעם, כאשר בכל שלב המונה משתנה ותוצאת כל הצפנה מהווה חלק מזרם המפתח איתו מצפינים את הבלוק הבא של המידע בחיבור XOR.
- יתרונות CTR
- 1- מקביליות – אופן מונה ניתן ליישום באופן מקבילי, למשל אפשר לחלק את ההצפנה של הבלוקים הקריאים בין מספר מעבדים כאשר כל מעבד מתחיל מהיסט אחר של המונה.
 - 2- ביטחון – הוכח כבטוח לפי מודל התקפת גלוי נבחר בתנאי שווקטור האתחול נבחר באקראי בכל פעם שמצפינים מסר חדש.

- 3- אקראיות – תומך בגישה אקראית, כאשר מעוניינים לפענח או להצפין בלוק מסוים, אין צורך להתחיל מהתחלה.
- חסרונות CTR – החיסרון של CTR הוא ששינוי של סיבית אחת של הצופן משפיע רק על הסיבית המקבילה בטקסט המקור (Plaintext). בתנאים מסוימים הדבר מאפשר התקפת הבחנה שבה התוקף מבצע Bit Flipping כדי להשיג מידע על הטקסט המוצפן, לכן המתכנת המתוארת אינה מוכחת בטוחה לפי מודל התקפת טקסט מוצפן ניתן לבחירה (Chosen Ciphertext Attack) המחמיר ביותר.



בקצרה –

CTR	OFB	CBC	ECB
<ul style="list-style-type: none"> ■ מצפינים את מספרי הבלוקים (מספרים טבעיים שתמיד עולים) בזה אחר זה ומקבלים ביטים אקראיים שתלויים במספרי הבלוקים ובמפתח, כלומר מייצרים מפתח זרימה בעזרת מספרי הבלוקים בלבד. ■ כל עוד המפתח הוא סודי, המצב טוב. ■ גודל Counter הוא כגודל הבלוק, כלומר 128 סיביות. ■ אין כאן צורך לחשב דברים מהבלוק הקודם, כלומר כל הבלוקים עצמאיים, וכך ניתן להעלות את התפוקה ולעבוד על כולם במקביל. ■ CTR הוא יותר טוב מ-OFB ולא פחות טוב מ-CBC. ממיר Block Cipher ל-Stream Cipher 	<ul style="list-style-type: none"> ■ המרת צופן לצופן זרימה, כאן מצפינים את IV ומייצרים ממנו מפתח זרימה (key stream) שמוצפן שוב ושוב, ממנו מתקבל רצף אקראי של ביטים, איתנו מצפינים את הקיבוץ בעזרת XOR. ■ OFB הוא לא פחות חזק מ-CBC. ■ הקובץ לא עובר הצפנה, אלא רק ה-IV שהוא הגורם המרכזי כאן, המאפשר את מפתח הזרימה. ■ אין כאן שום תלות בין הבלוקים, כי אין אשר בין ההודעה למפתח הזרימה. ■ אין כאן שום פעפוע, וסיבית פגומה אחת לא תקלקל עוד סיביות, לטוב או לרע. ■ במקרה של אובדן סיביות – הכל מאבד סנכרון והכל אבוד. ממיר Block Cipher ל-Stream Cipher 	<ul style="list-style-type: none"> ■ הוסיפו תלות בין הבלוקים, כך שהסדר שלהם חשוב ומשפיע על הפלט. ■ הודעות זהות יוצפנו באופן שונה תחת אותו מפתח (בתנאי שה-IV ווקטור האתחול שונה), לצורך כך הוסיפו את גורם האקראיות לפני הצפנה. אותו מפתח עם ווקטורי הצפנה שונים ייתן תוצאות שונות, עד כדי שלא ניתן יהיה לזהות שמדובר באותה הודעה. ■ הודעות זהות ייתנו תוצאה זהה רק אם ה-IV זהה (והמפתח). ■ IV (ווקטור האתחול) חייב להיות אקראי, הוא משפיע ומכניס אקראיות רק לבלוק הראשון, ואחריו כל בלוק משפיע על הבלוק שבא אחריו, כלומר באופן עקיף כל בלוק (וגם IV) משפיע על כל הבאים אחריו. ■ IV לא מוסיף סיביות על המפתח, אלא מכניס רק אקראיות להודעה, כדי להגדיל את אי יכולת ההבחנה. ■ CBC מייצר שרשור בין הבלוקים. 	<ul style="list-style-type: none"> ■ כדי להצפין קובץ שהוא מעל לגודל הבלוק, יש לחלק את הקובץ לחלקים. ■ הקובץ מחולק לבלוקים, עפ"י גודל הבלוק, ומצפינים את הבלוקים בנפרד, אחד אחרי השני. זו השיטה הפשוטה, אך היא בעייתית. ■ מדובר בתצורת הפעלה (block cipher mode), ללא קשר לצופן עצמו, כלומר כל צופן בלוקים יעבוד כאן. ■ הבעיה כאן היא ש-2 בלוקים בעלי אותו תוכן ייתנו תוצאה זהה, כי החלקים עצמאיים והוא משמר את התבניות ביניהם, וזה לא טוב. ■ בנוסף, אין כל קשר בין הבלוקים, ולכן אין לדעת אם הם הגיעו בסדר הנכון. ■ בלוק פגום לא משפיע כאן על הבלוקים האחרים. עניין זה יכול להיות יתרון, אך יכול גם להיות חיסרון, כי אולי במקרה כזה נעדיף לדעת שכל ההודעה פגומה. ■ לא משנה כמה חזק הצופן, הוא משמר את הבלוקים, ולכן זוהי טעות לעבוד עם ECB.

		<ul style="list-style-type: none"> ▪ כדאי לבחור IV חדש עבור כל הודעה. ▪ לרוב מעבירים את ווקטור IV באופן גלוי מהשולח למקבל, ללא הצפנה, רק המפתח חייב להיות סודי. ▪ כל בלוק תלוי בבלוקים הקודמים לו. בפועל עובדים רק מול הבלוק הקודם, המתמצת את ההשפעה של כל מה שקדם לו. IV מאפשר שרשור במחיר נמוך. ▪ אם סיבית כלשהי בבלוק מוצפן כלשהו היא פגומה – אחרי פענוח, בצופן טוב, כ- 50% מהסיביות בבלוק זה יהיו פגומות, כלומר הוא יתקבל משובש לגמרי. ▪ התפשטות הטעות (Error Propagation) – בגלל השימוש ב-XOR אם יש בבלוק מסוים ביט פגום, הבלוק שייגזר מבלוק זה אבוד לגמרי. ▪ אובדן ה-IV ישפיע על הבלוק הראשון בהודעה, לאחר מכן יהיה בסדר. 	
--	--	---	--

- Cryptographic Hashes

- Cryptographic Hashes

פונקציית גיבוב קריפטוגרפית היא פונקציית גיבוב חד-כיוונית הממירה קלט באורך כלשהו לפלט קצר, באורך קבוע, הנקרא "קוד גיבוב" או "ערך גיבוב". ערך הגיבוב משרת כייצוג קומפקטי של הקלט או כאמצעי זיהוי ייחודי שלו, מעין "טביעת אצבע דיגיטלית", יש נוהגים לכנותו "תמצית-מסר" (Message Digest) ועיקר השימוש בו הוא להוכחת שלמות ואימות הקלט. אם נעשה שינוי כלשהו, אפילו מינורי, בתוכן הקלט לפונקציה, בסבירות גבוהה מאוד הפלט יהיה שונה לגמרי ולכן ניתן יהיה להבחין בזאת בקלות. בניגוד לפונקציית גיבוב רגילה, פונקציית גיבוב קריפטוגרפית חייב להיות חד-כיוונית במובן שבהינתן הפלט יהיה קשה מבחינה חישובית למצוא את קלט המקור שלו ולכן יהיה קשה ביותר לגרום לגורם זדוני לזייף קלט באופן כזה שיתקבל ממנו ערך גיבוב זהה.

- תכונות בסיסיות –

1- קושי בהיפוך (Preimage Resistance) – בהינתן פלט מסוים, מציאת הקלט המקורי שלו, קשה מבחינה חישובית. כלומר היריב מקבל Digest וצריך לתת קלט מתאים, אם הוא מצליח לעשות את זה מהר – הוא מנצח ואז למעשה פונקציית הגיבוב לא טובה. הסיבוכיות הרצויה במקרה הזה היא $O(2^n)$, אם באופן עקבי זה לוקח ליריב פחות זמן אזי הפונקציה לא טובה.

2- קושי במציאת מקור נוסף (2nd-Preimage Resistance) – בהינתן קלט כלשהו יהיה קשה חישובית למצוא קלט אחר

המוביל לאותו פלט. במילים אחרות כאן היריב מקבל Digest וגם File1 כקלט, כאשר ערבול של File1 נותן את ה-Digest, זה מחייב את היריב למצוא קובץ קלט אחר ושונה הממופה לאותו Digest, סיבוכיות $O(2^n)$.

3- קושי במציאת התנגשויות (Collision Resistance) – מציאת שני קלטים שונים כלשהם המפיקים פלט זהה קשה מבחינה חישובית, במילים אחרות האתגר כאן הוא למצוא 2 קלטים שונים (File1, File2) הממופים ע"י פונקציית ערבול לאותו

Digest, כאשר הסיבוכיות כאן היא שונה מהקודמים $O(2^{\frac{n}{2}})$

פונקציה שעומדת בתנאים 2+3 נקראת Collision Resistance Hash Function (מניעת התנגשויות)

פונקציה שעומדת בתנאים 1+2 נקראת One Way Hash Function (יצירת התנגשויות)

באופן כללי, נרצה שפונקציית הגיבוב שלנו תעמוד בכל 3 התנאים, על מנת שתהיה טובה.

SHA-2 ו-SHA-3 עומדים בשלושתם.

- שימושים –

1- הבטחת שלמות – השימוש העיקרי בפונקציית גיבוב הוא הבטחת שלמות של פיסת מידע דיגיטלי כמו קובץ, הודעת דואר, מפתח הצפנה או מסר כלשהו.

2- חתימה דיגיטלית – פונקציית גיבוב בד"כ מהירה מאוד לכן מסיבות של יעילות כאשר רוצים לחתום חתימה דיגיטלית על מסמך גדול, עדיף תחילה להכין מהמסמך תמצית ולחתום עליה במקום לחתום על המסמך כולו.

3- הגנה על סיסמאות – שיטה ותיקה לניהול סיסמאות נעשית באמצעות פונקציית גיבוב. סיסמאות לעולם אינן נשמרות במצב גלוי מחשש לפריצה למערכת וגניבתן.

- סוגי פונקציות –

1- פונקציית גיבוב ללא מפתח – נקראת גם (MDC) Modification Detection Code. תפקידה להוות ייצוג תמציתי של המסר כך שיהיה אפשר לחשוף כל שינוי קל ביותר בסבירות גבוהה. דוגמאות שימוש: הגנה על סיסמאות וחתימה דיגיטלית. בסיסמאות למשל הן אינן נשמרות במצב גלוי אלא רק ערך הגיבוב שלהן נשמר ובחתימה דיגיטלית מסיבות של יעילות, החתימה מתבצעת על תמצית מהמסר (Message Digest) במקום המסר כולו.

2- פונקציית גיבוב עם מפתח – נקראת גם (MAC) Message Authentication Code ומשמשת בעיקר לאימות מסרים. באופן כללי MAC שונה מ-MDC בכך שהוא מכיל גם **TIO** שמקשה על תקיפה של ההודעה, תוקף שלא יודע את הסוד לא יצליח לחשב את ה-Digest שיתאים להודעה.

- HMAC – שילוב בטוח של המפתח הסודי עם המסר וערכים קבועים נוספים, ע"י XOR.

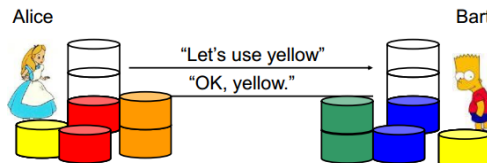
- Key Establishment - Diffie-Hellman
- Intro to Public/Private Key Encryption
- RSA Algorithm
- High Level View
- Modular Mathematics
- Proof of Correctness

• Key Establishment – Diffie-Hellman

פרוטוקול דיפי-הלמן הוא פרוטוקול שיתוף מפתח הראשון, שמאפשר לשני משתתפים שלא נפגשו מעולם ואינם חולקים ביניהם סוד משותף כלשהו מראש, להעביר אחד לשני מעל גבי ערוץ פתוח (שאינו מאובטח) סוד כלשהו כך שאיש מלבדם אינו יודע מהו. הסוד יכול להיות בהקשר של פרוטוקול תקשורת הצפנה. פרוטוקול דיפי-הלמן מתמודד עם בעיית הפצת המפתחות בשיטה אסימטרית ע"י פונקציה חד-כיוונית.

- יתרונות – יתרונות הפרוטוקול הם שהוא פותר את המשתתפים בתקשורת מוצפנת לשתף מפתחות הצפנה סודיים מראש, תחת זאת המצפין יכול להכין מפתח שיחה ארעי, להעבירו באמצעות הפרוטוקול לצד השני ואז התקשורת ביניהם יכולה להיות מוצפנת באמצעות מפתח זה עם צופן סימטרי לפי בחירה כמו AES ובגמר השימוש המפתח מושמד. אם כל המפתחות בפרוטוקול הינם חד-פעמיים ולא חוזרים להשתמש בהם שוב, הרי שהפרוטוקול מספק סודיות מושלמת קדימה במובן שמרגע שהמפתח הושמד אין שום דרך לשחזר אותו, אפילו אם בעתיד תתגלה דרך יעילה לפצח את הפרוטוקול.
- ביטחון הפרוטוקול – הפרוטוקול מספק הגנה על סודיות המפתח המשותף כנגד יריבים פסיביים המסוגלים לצותת לערוץ התקשורת בלבד. הפרוטוקול לא מספק הגנה מפני יריב אקטיבי המסוגל ליירט, לחסום או "להזריק" מסרים כרצונו, למעשה הפרוטוקול אינו מספק מה שקרוי "אימות זהויות המשתתפים". הם יכולים להיות סמוכים ובטוחים כי מלבדם אין איש יודע מהו המפתח ששיתפו, אך אינם מקבלים כל ערובה לגבי זהות המשתתף האחר. מסיבה זו הפרוטוקול בגרסה הבסיסית פגיע במיוחד להתקפת Man In The Middle (מתקפת האדם באמצע).
- דוגמה לאופן הפעולה (אנלוגיית צבעים) –

- 1- אליס וברט מסכימים על צבע ציבורי (**צהוב** במקרה הזה) שגלוי לכולם, ושומרים ליטר ממנו.
- 2- כל אחד בתורו, בוחר צבע רנדומלי (אצל ברט צבע **כחול** ואצל אליס צבע **אדום**) ושומר 2 ליטר ממנו.
- 3- כל אחד בתורו, מערבב ליטר אחד של הצבע הציבורי עם ליטר אחד של הצבע הסודי מהשלב הקודם, כתוצאה מהערבוב בארט מקבל צבע **ירוק** מערבוב של **צהוב** ו**כחול**, ואילו אליס מקבלת צבע **כתום** מערבוב של **אדום** עם **צהוב**. לאחר הפעולה, אליס וברט נשארים עם ליטר מהצבע שנבחר בסעיף הקודם, אליס נשארת עם ליטר צבע **אדום** וברט נשאר עם ליטר צבע **כחול**.



- 4- אליס וברט מחליפים ביניהם את מיקס הצבעים על גבי ערוץ ציבורי.
- 5- כאשר אליס וברט מקבלים כל אחד את הצבע של השני, הם שופכים לתוכו את הליטר של הצבע הסודי שנשאר להם לאחר הערבוב, אצל ברט מתקבל צבע **חום** כתוצאה מערבוב של **כתום** עם **כחול**, ואילו אצל אליס מתקבל צבע **חום** כתוצאה מערבוב של צבע **אדום** ו**ירוק**.
- 6- לסיכום, הסוד המשותף לאליס וברט מורכב מהצבעים ציבורי + צבע של אליס (אדום) + צבע של ברט (כחול).
- דוגמה מעשית לאופן הפעולה –
- 1- אליס ובוב מחליטים ש- $p=71$ ו- $g=7$.
- 2- אליס בוחרת מפתח סודי $A=5$ ומחשבת את המפתח הציבורי בעקבותיו באופן הבא: $g^A=7^5=51 \pmod{71}$, את התוצאה היא מעביר לברט.
- 3- ברט בוחר מפתח סודי $B=12$ ומחשב את המפתח הציבורי בעקבותיו באופן הבא: $g^B=7^{12}=4 \pmod{71}$, את התוצאה הוא מעביר לאליס.
- 4- אליס מחשבת את המפתח הסודי באופן הבא: $S=(g^B)^A=4^5=30 \pmod{71}$.
- 5- ברט מחשב את המפתח הסודי באופן הבא: $S=(g^A)^B=51^{12}=30 \pmod{71}$.

• Intro to Public/Private Key Encryption

הצפנת מפתח ציבורי נקראת גם הצפנה אסימטרית, שבו מפתח ההצפנה שונה ממפתח הפענוח. כלומר, כל משתמש מכין לעצמו זוג מפתחות: מפתח ציבורי (Public Key) שהוא מפתח ההצפנה נגיש לכל ומפתח פרטי (Private Key) מתאים, הנשמר בסוד ומשמש לפענוח. ההתאמה היא חח"ע (לכל מפתח ציבורי קיים אך ורק מפתח פרטי יחיד המתאים לו, ולהפך). כדי להצפין מסר בשיטה זו על המצפין להשיג לידי עותק אותנטי של המפתח הציבורי של המקבל, שבעזרתו הוא מצפין ושולח לו את המסר. רק המקבל מסוגל לשחזר את הטקסט המוצפן בעזרת המפתח הפרטי המתאים שברשותו.

- יתרונות

- 1- Principal צריך מפתח ציבורי אחד ומפתח פרטי אחד.
- 2- מס' המפתחות האפשרי עבור צמד זוג מפתחות הוא $O(n)$.

- חסרונות

- 1- אלגוריתמים שמממשים עיקרון זה הינם קשים ליישום.
- 2- דורש משאבים גבוהים.

• RSA Algorithm

אלגוריתם ההצפנת מפתח ציבורי דטרמיניסטי. ב-RSA כבכל מערכת מפתח ציבורי, מפתח ההצפנה אינו סודי והוא שונה ממפתח הפענוח שנשמר בסוד, ולכן האלגוריתם נחשב אסימטרי. ב-RSA השולח משתמש במפתח ההצפנה הציבורי של הנמען כדי להצפין עבורו מסר כך שרק הנמען מסוגל לפענחו באמצעות המפתח הפרטי המתאים שברשותו.

- דוגמא לאופן הפעולה –

- 1- נבחר p ו- q אקראיים, למשל $p=71$ ו- $q=71$, שניהם חייבים להיות מספרים ראשוניים.
- 2- מחשבים את $n: n=pq$, **n חייב להיות מעל 2048 סיביות**. n הוא ציבורי, כולם מקבלים אותו.
- 3- כעת נבחר את e , שהוא צריך להיות מספר ראשוני רלטיבי ל- n . כלומר, מספרי ה- totient הם מספרים ראשוניים רלטיביים ל- n (מספרים בין 0 ל- n שהמחלק המשותף המקסימלי ביניהם הוא 1).
למשל אם ניקח את $n=9$ אז totient יהיו 1,2,4,5,7 מספרים אלו נקראים ראשוניים רלטיביים מאחר והמחלק המשותף המקסימלי ביניהם לבין 9 הוא 1. $Gcd(9,4) = 1$ לעומת $gcd(9,6) = 3$, לכן 4 יהיה ראשוני רלטיבי ל-9 ו-6 לא יהיה ראשוני רלטיבי.
- בחרים מס' נוסף e ראשוני רלטיבי (כלומר מס' ראשוני שהוא לא מחלק ב-3220), אפשר לבחור 79, 97 ועוד מספרים ראשוניים אחרים בטווח. לצורך הדוגמה נבחר ב-79.
- 4- בוחרים d כך ש: $d=79^{-1} \pmod{3220}$
- 5- המפתח הסודי הוא (1019, 3337).

על מנת להצפין את ההודעה $m=688232687966683$ נחלק לחלקים הקטנים מ-3337 (688, 232, 687, 966, 683).

הצפנה: $E((79, 3337), 688) = 688^{79} \pmod{3337} = 1570$

פענוח: $D((1019, 3337), 1570) = 1570^{1019} \pmod{3337} = 688$

***שני מספרי הם ראשוניים רלטיביים (relatively prime) אם המחלק המשותף שלהם הוא המס' 1 בלבד.

- Dolev-Yao
- Shared Key Authentication

• Dolev-Yao

מודל מושגי המגדיר מול איזה תוקף רוצים להתגונן. למעשה המודל מתאר את המצב הבא: יש משתתפים טובים שמעבירים הודעות על תווך תקשורת מסוים בזמן שעל התווך נמצא תוקף שיכול להאזין לכל מה שעובר. התוקף הוא לא ווירוס באחד הצדדים אלא נמצא ממש על התווך ולא יכול להפריע לפעולתם של המשתתפים.

נהוג לראות את הרשת עצמה כיריב/אויב מכיוון שמניחים שנמצא בה תוקף שיכול להפריע להודעות, לערוך אותן, להמציא הודעות, לשלוח הודעות חוזרות, לחסום הודעות וכו'. לכן, לפי מודל זה אנחנו מקשים על התוקף בהקשר של הצפנה/פענוח ורוצים להגיע למצב שהתוקף לא מסוגל לפענח הודעות אם אין לו מפתח ללא תלות בצופן, התוקף מצידו לא יכול לנחש מפתחות ולא יכול לבצע Brute Force. במילים אחרות, הרעיון הוא שתוקף מסוגל לשלוט על התקשורת ולא להתעסק באופן ההצפנה (במידה ומידע מוצפן אז הוא מוצפן ללא ספק).

המודל מניח שיש הודעות כאשר כל הודעה בנויה משדות, כל שדה יכול להיות מורכב מטיפוסים פשוטים. התוקף מסוגל להקשיב ולאגור המון ידע, כעת התפקיד שלו להפריע לאחרים, והשאלה הנשאלת היא כיצד התוקף יכול לבצע זאת.

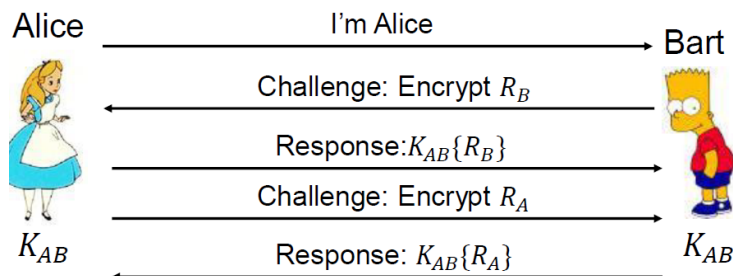
- כעת נראה איך נוכל להגיע למצב שבארט יודע שאליס דיברה איתו במודל של Dolev-Yao

איך בארט יודע מי זו אליס בכלל? נניח איכשהו שיש להם מפתח משותף וכעת רוצים להשתמש בו לאימות, איך אליס תאמת מול בארט שזו אכן היא?

מה כן כדאי לעשות? קוראים לזה Challenge/Response, הרעיון היא פילוסופי קצת: במקום שאליס תגיד לברט מה הסוד או שבארט יגיד לאליס מה הסוד, אנחנו נחייב את שניהם להשתמש בסוד, ועצם השימוש משכנע את בארט שזו אכן אליס, וההפך. זה נקרא הוכחה אינטראקטיבית. העיקר הוא זה שהמוכיח לא מגלה את הסוד, אלא רק משתמש בסוד ומשכנע את הצד השני בעזרת אתגר ותשובה. הנקודה כאן היא שהבודק משוכנע אבל הבודק לא מסוגל להוכיח לאף בן אדם אחר שהבן אדם לא משקר.

• Shared Key Authentication

- נניח שברט ואליס חולקים מפתח סודי K_{AB} (נניח המפתח הגיע אליהם דרך גורם שלישי), כעת אליס וברט רוצים לתקשר מעל רשת תקשורת אך קודם לכך אליס רוצה לוודא שבברט זה אכן הוא, ולהפך.
- הפתרונות –
- 1- Weak Authentication – אליס שולחת לברט את המפתח K_{AB} שמשמש בתור סיסמא, הסיסמא נחשפת בפני משקיפים פאסיביים. עובד רק בכיוון אחד ואליס לא יודעת שהיא מדברת עם ברט.
 - 2- Strong Authentication – הפרוטוקול לא חושף את הסוד, ברט מבקש הוכחה מאליס שהיא יודעת את הסוד, אליס מבקשת הוכחה גם כן מברט.



נושאים:

• **Protocol Types**

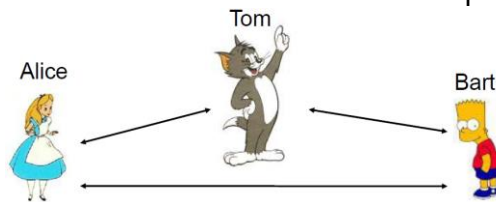
- Arbitrated Protocols
- Adjudicated Protocols
- Self-Enforcing Protocols

• **Protocol Types**

- **Arbitrated Protocols** – בין שיחה של שני משתתפים (ברט ואליס) קיים גורם שלישי נוסף מתווך (תום) שעליו משתתפי השיחה (ברט ואליס) סומכים בכדי לנהל תקשורת ביניהם. המתווך (תום) אינו תופס עמדה והוא למעשה גורם שמבטיח הוגנות והוא יכול להיות בן אדם, מכונה או תוכנה. דוגמאות לגורם שלישי מתווך: מתווך נכסים, עו"ד, Gmail ועוד.

- **חסרונות**

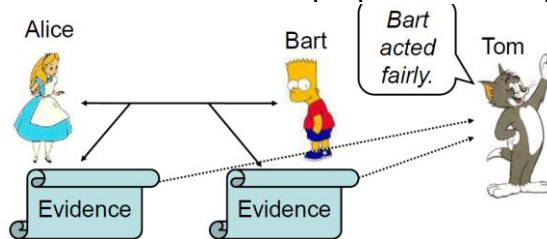
- 1- שני הצדדים עלולים שלא למצוא גורם שלישי שעליו שניהם הסכימו.
- 2- עלויות החזקת זמינות המתווך הן גבוהות, הן עבור המשתמשים והן עבור הרשת עצמה.
- 3- מתווך גורם לעיכובי זמנים בפעולות תקשורת מכיוון שבתור גורם שלישי, הוא צריך לקבל, לעבד ולהעביר כל פעולת תקשורת.
- 4- אם מתווך נמצא בשימוש רב (משמש כגורם שלישי במס' שיחות) הוא עלול להיות צוואר-בקבוק עקב כך שהרבה משתמשים מנסים להיעזר בו כמתווך.
- 5- הפרטיות נפגעת עצם העובדה שהמתווך חשוף למידע שמועבר בין שני צדדים.
- 6- המתווך עלול להוות יעד למתקפות.



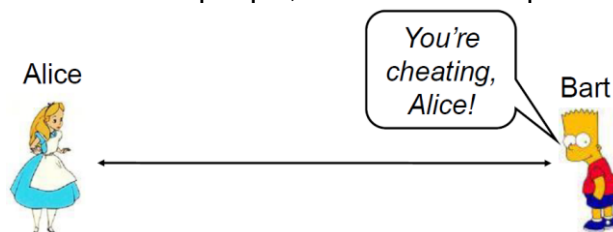
- **Adjudicated Protocols** – בין שיחה של שני משתתפים (ברט ואליס) לא קיים מתווך, אך הפרוטוקול נותן לכל אחד ממשתתפי השיחה מספיק ראיות למקרה שגורם שלישי (תום) יצטרך להתערב ולהחליט במקרה של קונפליקט בין הצדדים. לדוגמה כאשר שני אנשים הסכימו על חוזה מסוים ואחד מהם הפר סעיף שקיים בחוזה, החוזה יכול לשמש בתור ראיה. כמו כן, במקביל לכך שגורם שלישי יכול להכריע מי רימה במהלך השיחה.

- **נקודות חשובות**

- 1- עלויות נמוכות יותר בהשוואה ל-Arbitrated Protocols.
- 2- גילוי הכישלון מתרחש רק לאחר שהכישלון אכן התרחש, ולא לפני.



- **Self-Enforcing Protocols** – פרוטוקול מתוחכם בו אף אחד ממשתתפי השיחה לא צריך לסמוך על משתתף השיחה השני. כאשר משתתף מנסה לרמות המשתתף השני מיד יודע זאת, אין צורך בגורם שלישי שיבטיח הגינות



נושאים:

- Digital Signatures
- Key Establishment
- KDC, KTC

• Digital Signatures

חתימה דיגיטלית היא שיטה מתמטית המאפשרת להוכיח אותנטיות של מסמך או מסר דיגיטלי, חתימה דיגיטלית תקפה אמורה לשכנע את המקבל שהמסר או המסמך שקיבל נוצרו במקור מהשולח המוכר לו (אימות), השולח אינו יכול להכחיש שהוא זה שהכין את המסמך או המסר (אי-הכחשה) וכך שהמסמך או המסר לא שונו במהלך המשלוח אם בגלל תקלה בתקשורת, באמצעי האחסון או בזדון ע"י צד שלישי.

רק Principal יכול ליצור חתימה אך אחרים יכולים בקלות לזהות אותה

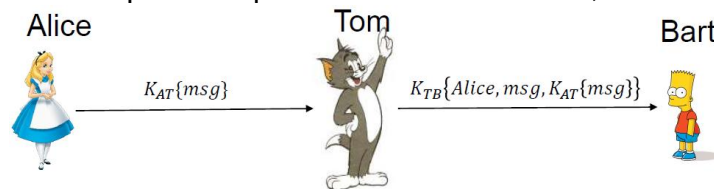
- שלושה סוגי חתימות דיגיטליות –

- 1- חתימה אלקטרונית – חתימה שהיא מידע או סימן אלקטרוני כלשהו המוצמד למסר אלקטרוני, סימן כזה יכול להיות תוכנה, סריקה של חתימה ידנית, סימן כלשהו המופק באמצעות מכשיר כמו עט דיגיטלי או משטח רישום דיגיטלי.
 - 2- חתימה אלקטרונית מאובטחת – חתימה שנתונה לשליטתו הבלעדית של החותם, המאפשר את זיהויו לכאורה וכן מאפשר לאתר כל שינוי אם אירע בתוכן המסר או המסמך החתום.
 - 3- חתימה אלקטרונית מאושרת – חתימה מאובטחת שמתלווה אליה תעודה אלקטרונית שהונפקה ע"י גורם מאשר ותפקידה לאמת את תקפות החתימה באופן כזה שהחותם אינו יכול להתכחש אליה.
- דרישות לחתימה דיגיטלית –
- 1- לא ניתנת לזיוף.
 - 2- ניתן לוודא שהקובץ והחתימה דומים.
 - 3- אם קיבלת חתימה על קובץ תדע שהקובץ לא עבר שינוי אחרי החתימה.
 - 4- לא ניתן להשתמש בחתימה על קובץ X ולהעביר לקובץ Y. כל חתימה היא חד פעמית. כמובן שבעולם האלקטרוני אין מושג של קובץ מקור, אבל המושג כאן הוא שאם חתמת על קובץ X לא ניתן להעביר את החתימה לקובץ אחר ששונה מ-X המקורי, במידה ומבצעים שינוי בקובץ חותמים שוב.
 - 5- במידה וכל הארבעה מתקיימים, נקבל את החמישי והאחרון: אי יכולת הכחשה, כאשר כל הארבעה קיימים לא ניתן להכחיש חתימה.

- חתימה דיגיטלית עם מפתח משותף (Digital Signature with Shared Keys) –

הרעיון הוא כזה: אליס מייצרת קובץ שנקרא "msg", היא רוצה לחתום על הקובץ אך אין לה את האפשרות לבצע זאת, לכן היא מעבירה את הקובץ למתווך (תום) שאליס סומכת עליו, ומבקשת ממנו שיחתום. כאשר ברט ירצה לקבל את הקובץ החתום, הוא יפנה לתום, שבתורו ישיב לו את השדות הבאים: שם השולח (אליס), תוכן ההודעה והצפנה. כך למעשה, יוכל תום לדעת שאליס היא זו שחתמה ע"י ווידוי מול תום.

- יתרונות: תום מבצע את העבודה, אליס וברט לא צריכים להחזיק סוד משותף כי הגורם השלישי עוזר.



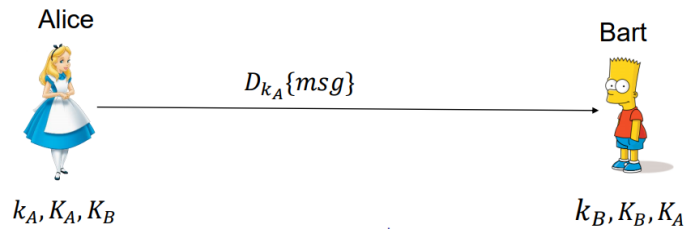
ניתן לייעל חותמת דיגיטלית ע"י הוספת שדות נוספים (שעה, יום, מס' רץ וכו')

- חתימה דיגיטלית עם מפתח ציבורי (Digital Signature with Public Keys) –

הרעיון הוא כזה: כל מי שמאזין יכול לשחזר את ההודעה. קיימת בעיה טכנית שלא נוכל לחתום על קבצים גדולים. נוכל להוסיף חותמות זמן כדי לבדוק שההודעה רלוונטית ולא נשלחה לפני 5 שנים.

- יתרונות: אין צורך בגורם שלישי, אלגוריתם פשוט יותר.

- חסרונות: יקר יותר, אין סודיות.



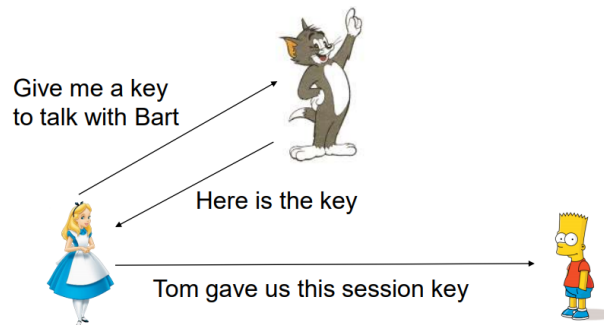
ניתן לייעל חותמת דיגיטלית ע"י הוספת חותמת זמן, שכבה נוספת של הצפנה ולשלב עם פונקציית גיבוב.

• Key Establishment

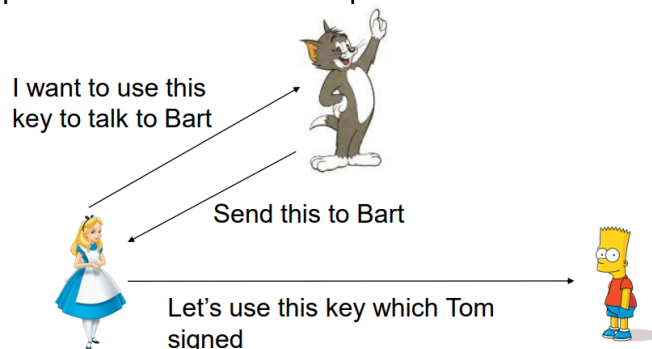
- כל שימוש במפתח מייצר בלאי, ככל שהמפתח בשימוש במחשב הסיכוי שייגנב/ייחשף עולה (להבדיל ממפתח שלא בשימוש), לכן מעדיף שימוש במפתחות חד פעמיים.
- Session Key – מפתח משותף (shared key) להצפנה של תקשורת למשך זמן קצר, יש צורך לאמת אותו תחילה, מוגבל להתקשרות (session) ספציפית.
- Symmetric Keys – מפתח אחד שידוע לשני המשתתפים, באמצעותו מצפינים ומפענחים.
- החיסרון העיקרי של הצפנת המפתח הסימטרי הוא כי כל הצדדים המעורבים יש להחליף את המפתח המשמש להצפין את הנתונים לפני שהם יכולים לפענח את ההודעות.
- Asymmetric Keys – שני מפתחות, סודי וציבורי, אחד עבור הצפנה והשני עבור פענוח.
- Point-to-Point with Symmetric – לכל אחד מהצדדים יש מפתח שידוע לשניהם באמצעותו מצפינים ומפענחים, ללא גורם מתווך, שיחה ישירה.
- Point-to-Point with PKE – כל אחד מהצדדים מחזיק מפתח סודי משלו ומפתח ציבורי ללא גורם מתווך, שיחה ישירה.

• KDC/KTC

- KDC (Key Distribution Center) – לפני כל התקשרות המתקשרים פונים תחילה לצד השלישי בסודיות באמצעות מפתחות ההצפנה שהם משתפים עמו כל אחד בנפרד כדי לקבל ממנו מפתח שיחה זמני מוצפן המתאים להתקשרות אחת.

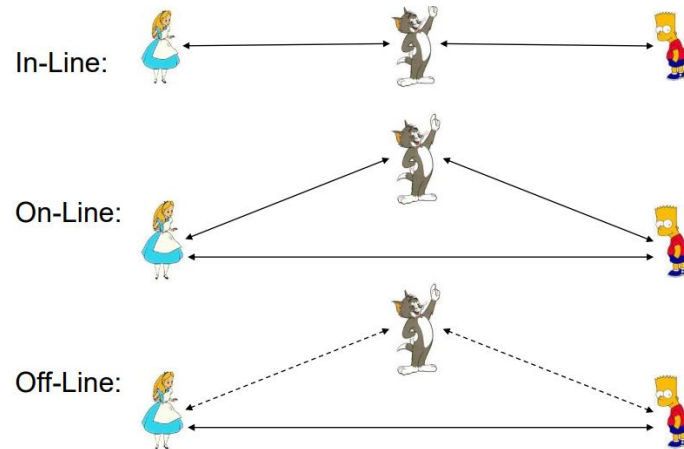


- KTC (Key Translation Center) – בדומה ל-KDC, אך כאן אליס בוחרת את המפתח לשימוש בשיחה. לפני כל התקשרות פונים תחילה לצד השלישי בסודיות, מעבירים לו מפתח והוא בתורו מחזיר את המפתח לאליס חתום ומוצפן בכדי שתוכל להעביר אותו לברט. ברט בתורו פותח את המפתח ומבקש מאליס להוכיח שהמפתח אכן שלה.



KDC ו-KTC נקראים TTP (Trusted Third Parties)

- TTP (Trusted Third Parties) – ישות שלישית שיוצרת תקשורת בין צדדים אשר סומכים עלייה, הישות הזו מנטרת את כל התקשורת בין שני הצדדים. לישות שלישית יכולים להיות שלושה תפקידים:
 - 1 - In-Line – תום באמצע, הכל עובר דרכו.
 - דוגמאות – WhatsApp, SMS.
 - 2 - On-Line – עובדים עם תום בתחילת השיחה ואח"כ השיחה מנוהלת בין אליס לברט ישירות.
 - דוגמאות – KTC, KDC.
 - 3 - Off-Line – אליס וברט מדברים עם תום מתישהו (בהתחלה) ואחרי שדיברו איתו, הם עובדים בלעדיו לפרק זמן ארוך.
 - דוגמאות – רשות אמון (Certificate Authority, CA).



נושאים:

- PKI (Public Key Infrastructure)
- X.509 Certificates
- Certificate Transparency

• PKI (Public Key Infrastructure)

- תשתית מפתחות ציבוריים (בקיזור PKI) היא תשתית של חומרה, תוכנה, אנשים, מדיניות ותהליכים הנדרשים כדי להנפיק, לנהל, להעביר, להשתמש, לאחסן ולגנוז תעודות דיגיטליות.
- PKI הוא הסכם המקשר מפתחות ציבוריים עם הגופים (למשל אדם או ארגון) שאליהם הם שייכים באמצעות רשויות אמון (בקיזור CA, Certificate Authority) אשר בסמכותן להנפיק תעודות דיגיטליות. הקשר מיוסד בתהליך של רישום וניפוק. הניפוק מתרחש אל רשות האמון (באמצעות תוכנה מחשב או פיקוח אנושי), בהתבסס על רמה מסוימת של אמון בין הצדדים (למשל ע"י הצגת מסמכים משפטיים המאשרים זהות של חברה). הגוף האחראי על בדיקת האמון בין הצדדים הוא מרכז רישום (בקיזור RA, Registration Authority). על ה-RA לוודא שהמפתח הציבורי אכן שייך לאותו גוף יחיד אשר נרשם, באופן שלא יאפשר לאותו גוף להתכחש לכך בעתיד. כדי לדעת אם תעודה כלשהי תקפה פונים לרשות תיקוף (VA, Validation Authority).
- תשתית PKI מורכבת מהגורמים הבאים:
- 1- רשות אמון (Certificate Authority, CA) – אחראית על ניפוק ואימות תעודות דיגיטליות. התפקיד המרכזי של רשות האמון הוא לחתום דיגיטלית ולהפיץ את המפתח הציבורי המקושר אל גוף נתון. פעולה זו מתבצעת ע"י המפתח הפרטי של רשות האמון, כך שהאמון במפתח הציבורי מבוסס על האמון במפתח הפרטי של רשות האמון. הקשר בין המשתמש למפתח שלו מיוסד בהסתמך על רמה מסוימת של אמון, באמצעות תוכנת מחשב או פיקוח אנושי. כל אחד יכול להיות CA, אבל לא בטוח שייסמכו עליו, כיום יש כ-200 ארגונים בעולם ואין היררכיה של CA מהטוב לפחות טוב. דפדפנים לרוב מגיעים מראש עם גורמי CA מאושרים, כל כמה ימים/שבועות יש עדכון של רשימת המאושרים בכל דפדפן מכיוון שאין צורך שה-CA תמיד יהיה online, מספיק שהוא הנפיק תעודה מכיוון שהבדיקה שלה בלאו הכי מתבצעת offline.
 - 2- מרכז רישום (Registration Authority, RA) – המאמת את הזהות של גופים המבקשים מידע מרשות האמון.
 - 3- רשות תיקוף (Validation Authority, VA) – ישות המספקת שירות המשמש לאימות תוקפו של אישור דיגיטלי לפי המנגנונים (למשל) של X.509.
 - 4- ספרייה מרכזית – מקום מאובטח שבו יאוחסנו מפתחות צופן.

• X.509 Certificates

- תעודה דיגיטלית הבנויה בצורה יחסית XML'ית, נפוצה ביותר בשימוש תשתית מפתח ציבורי (PKI), כדי לוודא שייכות מפתח ציבורי לישות מסוימת; מחשב, שירות או משתמש. התעודה נמצאת בשימוש נרחב באבטחת רשת האינטרנט כמו פרוטוקול SSL, SSH ועוד.
- תעודת X.509 מכילה פרטים אודות זהות נושא התעודה המחזיק במפתח הציבורי שהוא מצהיר כבעליו, זהות הרשות המאשרת שהנפיקה את התעודה וחתימתה. את תעודת המפתח הציבורי המשתמש שולח באופן גלוי לצד השני כחלק מפרוטוקול התקשורת. X.509 מפרט בין היתר שיטות לתיעוד, שיוך, אישור, פסילה ואימות מפתחות הצפנה אסימטריים.
- שדות התעודה
- 1- מס' גרסה - מספר הגרסה של התעודה משליך על אופי המידע הכלול בה ומבנהו.
 - 2- מס' סריאלי - המנפיקה חייבת לשייך מספר מזהה לתעודה כדי להבדילה מאחרות.
 - 3- אלגוריתם - מזהה אלגוריתם החתימה ששימש את המנפיקה כדי לחתום על התעודה.
 - 4- שם המנפיקה - שמה של הרשות המאשרת שהנפיקה את התעודה.
 - 5- תוקף - תוחלת החיים של התעודה, תאריך הנפקה ותפוגה.
 - 6- נושא - שמו של בעל התעודה, הכולל גם מאפיינים כמו מס' טלפון, דוא"ל, חברה תפקיד וכו'.
 - 7- מפתח ציבורי - סוגי אלגוריתם אסימטרי של נושא התעודה, מפתח ציבורי של נושא התעודה.
 - 8- הרחבות פונקציונאליות - כל הרחבה מזוהה ע"י עצם ייחודי הכולל אוסף ערכים וכן ערך המייצג את חשיבותה.
 - 9- אלגוריתם חתימה - פרטי אלגוריתם החתימה הדיגיטלית ששימש את המנפיקה כדי לחתום ולאשר את התעודה.
 - 10- חתימה - החתימה הדיגיטלית של המנפיקה.

- Certificate Chains – הלב של PKI, תעודות דיגיטליות חותמות על תעודות דיגיטליות אחרות בכדי לייצר שרשרת אמון, התהליך מפסיק כאשר תעודה רואה תעודה אחרת מוכרת.
- Certificate Revocation – השימוש הכי נפוץ בתעודות דיגיטליות הוא בסביבת האינטרנט. לכן, מה עושים כאשר תעודה נגנבה? כאשר עובד פוטר מחברה ויודע את כל המפתחות ויכול להפיץ אותם? לצורך כך יש רשימה של תעודות פסולות. כל הזמן קורה שתעודות נפסלות, אפשר להוציא אחת חדשה תוך מס' דקות. בנוסף אפשר להכניס שדה "תעודה פסולה" כחלק משדות התעודה.
- Certificate Revocation List – רשימה של תעודות פסולות, צריכה להיות מאושרת ע"י CA, כוללת חותמת זמן ומתעדכנת בתדירות גבוהה. הרשימה הזו עלולה להיות ארוכה מאוד, ולכן אפשר לבצע את הפעולות הבאות במקרה כזה:
 - 1- להראות רק תעודות פסולות חדשות (Delta CRL).
 - 2- חלוקה של הרשימה לפי סיבת הפסילה (לעיתים פסילה יכולה להתרחש כתוצאה מגניבה, מעבר של אתר למקום אחר וכו').
 - 3- חלוקה של הרשימה למקבצים.
- בעיות שנובעות מפסילת תעודות – רוב התעודות נפסלות כתוצאה מסיבות שלא קשורות לנושא אבטחה, ולכן רשימות ה-CRL הן לרוב ענקיות, ולכן עולה השאלה האם להוריד כל פעם את הרשימה כולה?
 - OCSP (Online Certificate Status Protocol) – בדיקה של תעודה און ליין, כאשר מתחילים עם שרת כלשהו, פותחים שיחה נוספת עם OSCP Responder שבודק ועונה מיד האם התעודה ששלח השרת בתוקף. הבדיקה היא מידית מול מאשר התעודה. הבעייתיות היא שהבדיקה מגדילה את תעבורת האינטרנט, פגיעה בפרטיות מכיוון שניתן לדעת לגבי פניות שביצע גורם מסוים לגורמים אחרים.
 - OCSP Stapling – בכדי לקבל יותר פרטיות, כאשר מתחילים שיחה חדשה עם שרת והוא שולח תעודה, הוא שולח עם התעודה תעודה נוספת, שמאשרת שהתעודה הראשונה מקורית ועדיין בתוקף.
 - Certificate Pinning – שיטה נוספת להתגברות על תעודות פסולות. בשיטה זו דואגים מזיוף של תעודה ספציפית, ולכן נוסף באופן ידני בדפדפן את מס' הזיהוי של תעודה ספציפית שאותה אני מצפה לקבל, למשל בעת גלישה לאתר האינטרנט של המכללה. דוגמא נוספת, למשל חברה שמודאגת מזיופים מספקת לדפדפים את התעודה שלה בכדי שבעת גישה לשירותיה של החברה, הגולשים למעשה יידעו לאיזו תעודת לצפות.
- בעיות שנובעות משימוש בתעודות – באופן כללי, יש המון חברות שמנפיקות תעודות ולכן יש המון תעודות, הבעיה היא שאין היררכיה בין המנפיקים (מנפיק טוב/לא טוב), ולכן דפדפן צריך להניח שהתעודות שבאות מכל ה-CA שהוא מכיר נכונות.
- רמות שונות של אמון תעודות (לפי Firefox ו-Chrome) –
 - 1- Domain Validated – בדיקה אוטומטית של הדומיין.
 - 2- Premium – בדיקה עם בן אדם, מוסיף שה-CA בדק עם בן אדם מהארגון המאושר.
 - 3- Extended Validation – בנוסף לשניים הקודמים, ברמה זו ה-CA חוקר ובודק בעצמו. התהליך למעשה כולל בדיקה עמוקה יותר ויש מעקב קפדני אחר הגוף המאושר.

• Certificate Transparency

- הופך את כל אישורי TLS הציבוריים של ישות סופית למידע הציבור הרחב. רשויות האישורים (CA) אחראיות באופן פומבי על כל התעודות שהן מנפיקות. שקיפות האישורים לא תכניס עוד גורמים מאושרים מצד שלישי.
- איך זה עובד?
- לוג של אישורים שניתן רק לכתוב אליו ולא למחוק ממנו.
- הסרבר של הלוג מוודא כי:
- בודק את שרשור האישורים (בודק ספאם וכו').
 - בתדירות קבועה מוסיף אישורים ללוג.
 - חותם על הלוג.
 - מפרסם את הלוג לכל העולם.
- הערות:
- הלוג לא מעיד על טיב האישורים, הוא רק מציג אותם.
 - הלוג הוא ציבורי, כולם יכולים לעיין בו.
 - הלוג אינו מהימן (לא ניתן לבטוח בו) – מאחר והלוג חתום, והעובדה שכולם יכולים לעיין בו ורואים את אותה רשימת אישורים, זאת אומרת שניתן לאמת באופן קריפטוגרפי.

אלגוריתם אבטחה לרשתות נתונים אלחוטיות (Wi-Fi). מטרת האלגוריתם לספק פרוטוקול אבטחה לרשתות אלחוטיות באותה רמת אבטחה שסופקה לרשתות נתונים קוויות. תפישת הפעולה מתבססת על הצפנת הנתונים לצורך העברתם בצורה מאובטחת. הפרוטוקול עובד בצורה של Stream Chipper. הרעיון הוא שמי שנכנס לרשת האלחוטית צריך לדעת את המפתח כדי לדבר עם ה-AP (Access Point).

- WEP Authentication – כל קליינט חייב להזדהות וליצור שייכות עם AP (נקודת גישה) לפני שהוא משדר מידע. השייכות החיבור בין הקליינט ל-AP, הפרוטוקול תומך בשתי שיטות זיהוי:

1- אימות במערכת פתוחה – שזהו חלק מדרישות הפרוטוקול וברירת המחדל של רוב ה-AP. מערכת פתוחה מאפשרת לכל הקליינטים לתקשר עם ה-AP, כל עוד הם מחוברים לאותה רשת אלחוטית (SSID זהה). עובדה זו מקנה אפשרות התחברות לרשת זו לכל קליינט בטווח השידור של הרשת.

- תהליך ההתחברות –

1. קליינט מבקש אימות.

2. AP עונה OK.

3. קליינט מתחבר לרשת.

2- אימות מפתח משותף – שולט על הגישה לרשת האלחוטית באמצעות מפתח משותף וע"י כך מקשה את ההתחברות לרשת של קליינטים לא רצויים (קליינטים ללא המפתח). ההצפנה חייבת להיות מאופשרת במקרה של מפתח משותף, מכיוון שהמפתח משמש להצפנה משמש גם לאימות.

- תהליך ההתחברות –

1. קליינט מבקש אימות.

2. AP שולח אתגר.

3. קליינט מצפין עם מפתח משותף ומחזיר ל-AP.

4. AP בודק את התשובה ומאשר בהתאם.

- WEP Encryption Details – WEP זהו מנגנון הצפנת זרם סימטרית (Symmetric Stream Chipper) אשר לוקח את גוף מסגרת מידע (Data Frame Body) ומעביר אותו דרך אלגוריתם הצפנה. גוף מסגרת המידע אז מוחלף בגוף מסגרת המידע המוצפן ומשודר לאוויר. התחנה הקולטת משתמשת באותו אלגוריתם על גבי המידע המוצפן כדי לפענח אותו לצורתו המקורית. חשוב לציין שמנגנון WEP מצפין אך ורק את גוף המסגרת (החלק המכיל מידע) לא את כותרתו (Headers) ובכך משאיר את כתובת המען והנמען חשופים לכל.

בצורתו המקורית, WEP משתמש במפתחות הצפנה באורכים של 40 bit או 104 bit.

מנגנון ה-WEP משתמש באלגוריתם ההצפנה RC4.

גודל ווקטור האתחול במנגנון WEP הינו 24 bit ולכן מס' האפשרויות הינו $2^{24} = 16,777,216$ אפשרויות.

• SSL• SSL

SSL) Secure Sockets Layer (ובגרסתו המעודכנת יותר TLS (Transport Layer Security) אבטחת שכבת התעבורה, הם פרוטוקולי האבטחה הפופולריים והחשובים ביותר של רשת האינטרנט. כמעט כל אתרי האינטרנט המוגנים באמצעים קריפטוגרפיים מסתמכים על פרוטוקולים המהווים חלק מהחבילה SSL/TLS. מסחר אלקטרוני, בנקאות מקוונת, דואר אלקטרוני, VOIP, מחשוב ענן ועוד. SSL/TLS נתמך על ידי מרבית הדפדפנים, בראשם גוגל כרום, מיקרוסופט EDGE, אינטרנט אקספלורר, ספארי, פיירפוקס ואופרה.

SSL/TLS הוא פרוטוקול ורסטילי שמטרתו אבטחת שיחת שרת/לקוח בשיטות קריפטוגרפיות חזקות והוא אמור למנוע ציתות, זיוף, או חבלה (שינוי זדוני) של המידע העובר בין השרת והלקוח. מאפשר חיבור אנונימי, אימות שרת (חד-צדדי) או אימות דו-צדדי, תוך שמירה על דיסקרטיות ושלמות המסרים. שלוש נקודות עיקריות שהפרוטוקול אמור לתת להן מענה הן:

- פרטיות – המושגת באמצעות הצפנה סימטרית.
- אימות – המושג באמצעות תעודת מפתח ציבורי.
- אותנטיות – המושגת באמצעות קוד אימות מסרים.

לפי מודל ה-OSI פרוטוקול SSL שוכן בשכבת השיחה, בין שכבת התעבורה לשכבת היישום.

- SSL Record Protocol – בשכבה זו מבוצעת חלוקת המידע העובר בקו לרשומות בגודל לכל היותר 2^{14} בתים, כאשר לכל רשומה צמוד תג אימות HMAC. שכבה זו תומכת בדחיסת נתונים, אם כי מסיבות של זכויות יוצרים לא מצוינת טכנולוגיה ספציפית למעט אלגוריתם NULL מנדטורי שאינו עושה דבר.

- אופן התהליך –

1. חלק את הפקטות לחבילות בגודל $2^{14} = 16,384$ או פחות.
2. בצע דחיסה (לא מחייב).
3. הודעת קוד אימות
4. הצפן
5. שרשר Headers:

- a. Change Cipher Spec, Alert, Handshake, Application Data – Content Type
- b. Major Version
- c. Minor Version
- d. Compressed Length

- SSL Handshake Layer – שמתחילה תהליך התקשרות, קובעת מספר פרמטרים משותפים כמו מספר גרסה ואלגוריתמים נתמכים (cipher suits). במהלך לחיצת היד המשתתפים משלימים תהליך אימות זהויות ומייצרים את מפתח השיחה. שכבה זו מניחה כברירת מחדל מצב "צופן NULL" שמשמעו ללא הצפנה ואימות כלל.

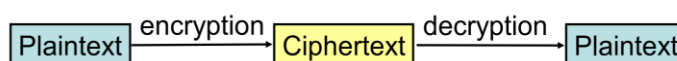
- SSL Alert Protocol – מכיל 2 בתים של מידע.

1. בייט ראשון – מדד חומרה, אם מאותחל ל-"1" אזי זו אזהרה, אם מאותחל ל-"2" אזי זוהי תקלה שלא ניתן לחזור ממנה.
2. בייט שני – קודים של אזהרות.

- SSL Alert Protocol

1. מאפשר שיחות וחיבור מאובטח ברשת האינטרנט.
2. עובד גם אם יש רק לצד אחד Certificate.
3. משמש בעיקר ל-Certificates ול-PKI.
4. ערוץ תקשורת מאובטח.

- 5- זיהוי/אימות (Authentication) – בודקים שמשאב/מחשב/מערכת הוא אכן מי שהוא טוען שהוא, כאשר הדרך הקלאסית נעשית באמצעות שם משתמש וסיסמא הנבדקים מול מאגר משתמשים כלשהו, בתום התהליך ניתן לדעת מי/מה התחבר. *** חשוב לזכור: בעת ביצוע התחברות, למעשה מעבירים למחשב מידע שכתוצאה מכך מקבל את היכולות של הגורם המתחבר ומזדהה במקומו, כסוכן של האדם הגורם שביצע את ההתחברות.
- 6- שלמות תנועות ואחריות (Transaction Integrity and Accountability) – הבטחת שלמות של דברים בין לקוח לעסק כלשהו. מנגנונים מאפשרים בדיקת שלימת בקלות בכדי להגיע למצב של "Non Repudiation" (אי הכחשת העובדה), אחרת כל גורם יוכל להכחיש כל עובדה, למשל הכחשת משלוח שהגיע למזמין רכישה.
- 7- באגים (Fault Tolerance) – באופן כללי באג = פעולה שמעצב המערכת לא תכנן להכניס. ככלל, נעדיף שלא יהיו כלל באגים, יחד עם זאת נעדיף באגים "רועשים" מאשר באגים "שקטים", ונרצה להתאושש מהם בקלות/נוחות.
- 8- סודיות (Message Secrecy) – המידע שעובר מצד א' לצד ב' לא קריא/מובן לגורמים חיצוניים. למעשה, הכי קל להשיג סוגיות וזאת באמצעות טכנולוגית הצפנה.
- 9- חשאיות (Covertness) – רמה מעל סודיות, חשאיות זו למעשה ההסתרה שכלל קיימת תקשורת, ועצם ההסתרה מונעת מראש ניסיונות פריצה. לרוב הסתרה של מידע מתבצעת בתוך איזשהו רעש סביבתי.
- 10- חיסיון (Confidentiality) – חיסיון מידע, למשל חיסיון מידע של ארגון שלא ניתן לבצע בו שימוש לא חוקי (מידע רפואי, מידע משפטי וכו').
- 11- פרטיות (Privacy) – היכולת והזכות להגיע על המידע האישי, למנוע כניסה למרחב האישי, וכמו כן, לשלוט לאיפה המידע האישי מגיע.
- 12- Subject – בן אדם פיזי.
- 13- Principal – שונה מהגדרה של Subject בכך שהגורם משתתף בתהליך שכולל אבטחה, למשל במקרה שבו בן אדם מקליד סיסמא למחשב, אז המחשב נחשב כ-Principle שעושה את העבודה עבור הבן אדם (עבודה יכולה להיות שיחת SSL, שליחת בקשה, הצגת תשובה וכו'). באופן כללי – Principal = מחשב ו-Subject = בן אדם.
- 14- Role – שיוך של תפקיד לישות, לרוב ה-Role משויך ל-Subject (למשל כאשר אליס מקבלת את תפקיד לקוח/מנהל), לאחר השיוך היכולות והמגבלות של ה-subject מוכתבות ממסגרת התפקיד. באופן כללי, תפקידים הם מאוד שימושיים, למשל כאשר רוצים לנהל המון משתמשים (subjects) קשה מאוד להחליט מה לתת לכל אחד ואחד (ברמה האישית), לכן מקובל שלא לבצע החלטה זו אלא לבנות תפקיד (Role) עם סט של יכולות ולשייך ישויות לתפקיד זה.
- 15- Group – קבוצה של משתמשים (Subjects), יכולות להיות מספר קבוצות לכל משתמש.
- 16- Trusted – רכיב במערכת שמניחים שעובד נכון, ססומכים עליו, אחרת – הכל אבוד.
- 17- Trustworthy – רכיב אמין שלא יקרוס, שיתנהג כמו שצריך.
- 18- Secrecy – סודיות, קריפטוגרפיה זו דרך לכתוב משהו מוסתר ולא מובן לקורא.
- 19- Integrity – יושרה, במידה והנתונים השתנו בדרך, מקבל הנתונים צריך לדעת זאת.
- 20- Authentication – אימות, עם איזה גורם מתקשרים בשיחה? ניתן להשגה ע"י אלגוריתמיקה של מנגנוני הצפנה.
- 21- Non-Repudiation – חוסר היכולת להכחשת עובדה אמיתית, למשל בעת שליחת מייל מגורם א' לגורם ב', גורם א' לא יוכל להכחיש שהוא שלח את המייל מכיוון שניתן יהיה להוכיח זאת, כך גם לקבל הצד המקבל שלא יוכל להכחיש את קבלת המייל.
- 22- Cryptographer – ממציא מערכות הצפנה.
- 23- Cryptanalyst – הפורץ למערכות הצפנה (בשני המקרים Cryptographer ו-Cryptanalyst הם אותו אדם מכיוון שקשה להיות ממציא מערכת שלא יודע לזהות פרצות אבטחה).
- 24- Cryptology – תורת ההצפנה.
- 25- Cipher – צופן, אלגוריתם מובנה עם צעדים מובנים אשר יודע להמיר מידע לפורמט אחר שאינו קריא. האלגוריתם מקבל מידע רגיל (plaintext), מעביר אותו תהליך שנקרא הצפנה לטקסט מוצפן ולא מובן (Ciphertext). אלגוריתם הופכי מבצע את פעולת הפענוח.



- 26- מדד אינדקס המקריות (IOC) – מדד הפיזור, אומדן לשונות של האותיות. הגדירו את ה-IOC הטיפוסי של טקסט ארוך באנגלית כ-0.068. ראשית, יש לנחש את אורך מילת הצופן, אורך הצופן מחלק את הטקסט המקורי לחתכים, ע"פ המיקום יש לבדוק עבור כל חתך האם ה-IOC שלו קרוב ל-0.068 ואם כולם פחות או יותר קרובים אליו זה אומר שהצלחנו לעלות על כמות האותיות הנכונה של המפתח.

- 27- One-Way Functions – פונקציה חד כיוונית היא פונקציה שממירה קלט לפלט באופן שקשה מאוד מבחינה חישובית להפוך את הפונקציה, דהיינו לשחזר את הקלט בהינתן פלט.
- 28- Trap Door Function – פונקציה שקלה לחישוב באופן חד-כיווני אבל קשה לחישוב בכיוון ההופכי ללא מידע קטן.
- 29- פרוטוקול – פורמט לשיחה, סוג של שיחה. מגדיר מה שולחים ומתי, איך עונים, איך שואלים, איך מגיבים, איך מפענחים וכו'. פרוטוקול עונה על כל השאלות האלה. פרוטוקול הוא מעין משחק של הודעות בין משתתפים, יש חוקים למשחק. הפרוטוקול אמור להגיע לאיזשהו יעד, הפרוטוקול מצליח כאשר מגיעים ליד. יעד יכול להיות למשל: אליס העבירה הודעה לברט וברט יודע את תוכן ההודעה, או למשל כאשר אליס שלחה הודעה וברט משוכנע לגמרי שאליס היא זו ששלחה את ההודעה. כאשר יש פרוטוקול ויש תוקף, התוקף מנסה להפריע ולגרום אולי לכישלון, אולי הוא רוצה לגרום לשיבוב (למשל שאליס תחשוב שיש שיחה בזמן שאין), הוא יכול לבצע את הפעולות הבאות: להפריע לשותפים בשיחה, לערוך אותם, להמציא הודעות, לענות להם, לחסום הודעות, לעכב הודעות, לחתוך את התקשורת, להציף את הרשת ועוד. מניחים שפרוטוקול מובן מאליו, לא צריכים ללמד את המשתתפים בשיחה על הפרוטוקול, מניחים שהם יודעים כבר.
- 30- Salt – הוספה של ביטים אקראיים לסיסמה בשביל להקטין את הסיכוי ששתי סיסמאות זהות יופיעו כערכים זהים בקובץ הסיסמה.