

مقدمة والمشفر الجمعي:

مقدمة:

تشكل عام يشتمل النظام المعلوماتي على الأشخاص (مستخدمون ومدراء)، العتاد (Hardware)، المعلومات (Information) والمعلومات (Data)، والبرمجيات (Software) فهو عبأ المتعددة وعلى الشبكات (Networks) بأشكالها وأنواعها.

أمن المعلومات (Information Security):

بشكل عام يمكننا تعرف أمن المعلومات بأنه مجموعة الاستراتيجيات الواجب استخدامها لتجنب التهديدات الأهمية وتأمين الحماية للنظام المعلوماتي بكامل مكوناته (العنادية والبرمجية والبشرية) من مختلف أنواع التهديدات (Threats) والمخاطر (Risks) والهجمات (Attacks).

يتضمن أمن المعلومات قسمين هما:

- التقنيات: مثل التشفير (Encryption).

- الاستراتيجيات: مثل الأمن الفيزيائي (Physical Security) ويشمل:

- أمن المباني (Buildings): كاميرات المراقبة، حراس الأمن، ... الخ.

- أمن الأفراد (People): عن طريق المقاييس الحيوية (Biometrics)

- أمن التجهيزات العنادية (Hardware): مثلاً حفظ نسخ احتياطية

الأهداف الأساسية لأمن المعلومات:

السرية (Confidentiality): حماية المعلومات من الوصول غير المصرح به.

التكاملية (Integrity): إمكانية التعديل على البيانات بواسطة المستخدمين المختل لهم بذلك فقط.

التوافرية أو الإتاحة (Availability): ضمان أن تكون المعلومات متاحة للمستخدمين والتطبيقات بالوقت المطلوب وبالشكل المحدد.

وقد تم مؤخراً توسيعة هذه الأهداف لتشمل أيضاً:

المسائلة (Accountability): ضمان عدم الإنكار (non-Repudiation) ذلك من خلال وجود سجل نشاطات النظام، فمثلاً بفرض كان لدينا مستخدمين A و B حيث أن المستخدم A قد أرسل رسالة للمستخدم B على الشكل التالي:



حتى نستطيع ضمان عند الإنكار يجب ألا يستطيع المستخدم A أن ينكر أنه هو من قام بإرسال هذه الرسالة.

الاستيقان (Authenticity): التحقق من الرسالة وصحة نقلها وطريقة إنشائها والوثق بذلك.

لكل هدف من أهداف أمن المعلومات هجمات يمكن أن يتعرض لها وهي:

- الهجمات على السرية: (Sniffing)
- تحليل الاعتراض (Traffic Analysis)

- الهجمات على التكاملية:

- التعديل (Modification).
- إعادة الاستخدام (Replay).
- الإنكار (Repudiation).
- الخداع والتكرر (Masquerading).

- الهجمات على التوافقية:

- قطع الخدمة (Denial Of Service) أو (D.O.S).

التشفير (Encryption):

هو عملية تحويل النص الأصلي (Plain text) المفروء إلى نص مشفر (Cipher text) غير مفروء إلا لمن يملك معرفة خاصة أو مفتاحاً خاصاً لإعادة تحويل النص المشفر إلى نص مفروء.

مكتبة الحكمية

اللادهه - مدخل إسيرو - مقابل باب السكن الجامعي

0993 508 666 & 0937 508 666 & 041/2439 666

الجلسة الأولى

الإخفاء (Steganography): هو فن وعلم قديم جداً يعتمد على كتابة رسائل مخفية ضمن رسائل أخرى بحيث لا يمكن لأحد أن يلاحظ وجودها باستثناء المرسل للرسالة المخفية والمستقبل لها، تم تطوير هذا العلم ليعمل وفق خوارزميات محددة أهمها خوارزمية البت الأقل أهمية (Least Significant Bit) LSB.

سؤال: ما هو الفرق بين الإخفاء (Steganography) والتشفير (Encryption)?

في التشفير:

تتم استخدام خوارزميات معينة تقوم بتغيير النص الأصلي وتحويله إلى نص مشفر ليس ذو معنى، أي إذا قام المهاجم باعتراض رسالة مشفرة فإنه يستطيع رؤية محتوى الرسالة ولكن المحتوى ليس نص ذو معنى حيث أن هذا النص يجب أن يتم فك تشفيره حتى يعود إلى نص أصلي ذو معنى.

في الإخفاء:

يتم استخدام خوارزميات إخفاء أجزاء معينة من الملف (يمكن أن يكون الملف صورة أو مقطع فيديو أو ملف نصي أو ...) ضمن ملف آخر مثلاً بذات من الصورة، مثل عليها أنه يمكننا استخدام صورة والتغليف على لون نقطة (Pixel) لكل مجموعة من النقاط لتقابل حرفًا أبجدياً من الرسالة المدعاة بكفاها.

أي أن الفرق الرئيسي بينهما هو أنه:

الإخفاء يتميز عن التشفير بأن الرسالة بحد ذاتها لا تجذب الاهتمام، الرسائل المشفرة تتدو كرموز مرئية وغير مقرئية (أي يبدو وبوضوح تشفير خوارزمية التشفير عليها) بينما الرسائل التي يطبق عليها أسلوب الإخفاء تتدو كرسائل عادية ولا تجذب أي اهتمام.

حتى نقول عن نظام ما أنه نظام تشفير (Crypto System) يجب أن يقوم بالعمليات التالية:

- معالجة النص (Text Processing): أي أنه يجب أن نقوم بمعالجة النص المراد تشفيره بشكل مناسب لكي يصبح دخل ملائم لخوارزمية التشفير.
- التشفير (Encryption)
- فك التشفير (Decryption)

مكتبة المكمية

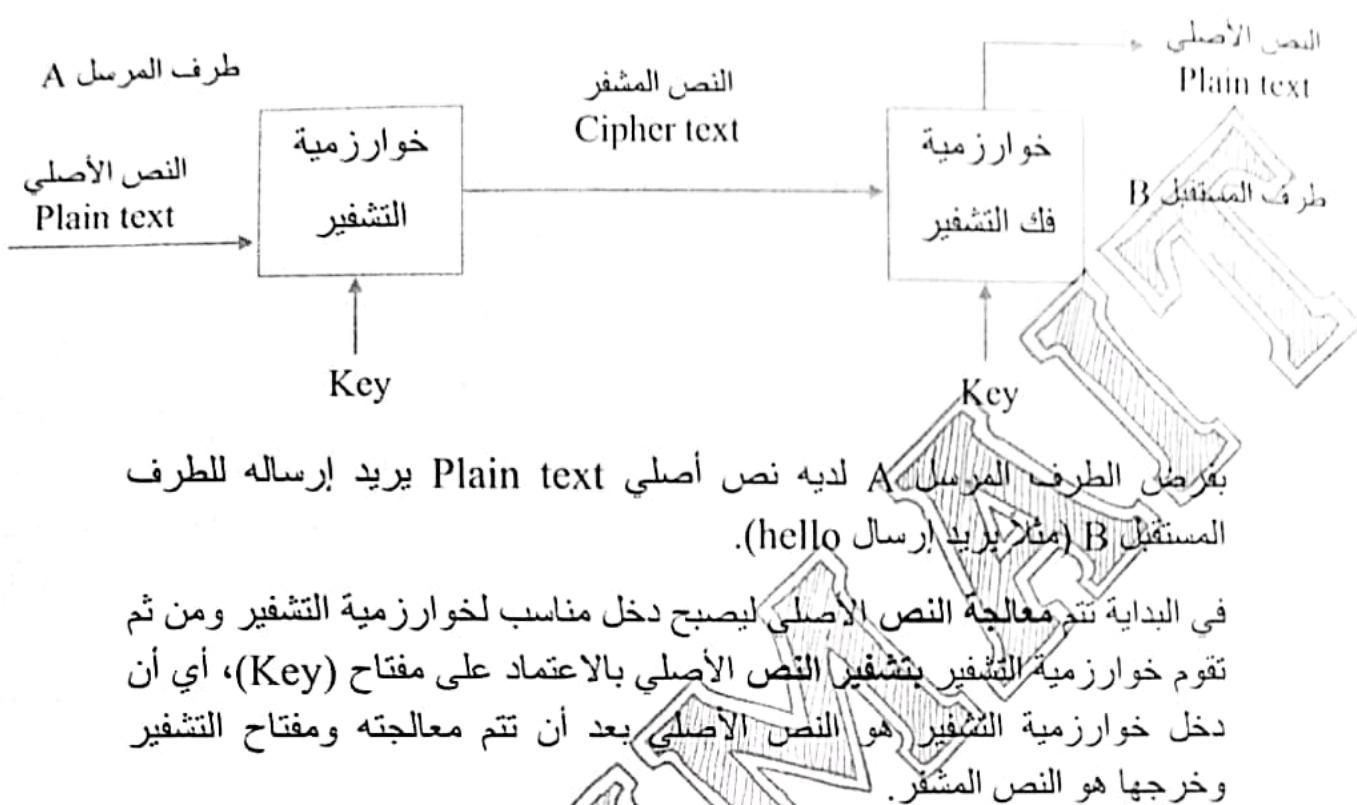
اللادفه - مدخل إسيرو - مقابل باب السكن الجامعي

0993 508 666 & 0937 508 666 & 041/2439 666

الجلسة الأولى



الشكل التالي يظهر طريقة عمل المشفرات:



بفرض الطرف المرسل A لديه نص أصلي Plain text يريد إرساله للطرف المستقبل B (مثلاً يريد إرسال hello).

في البداية تتم معالجة النص الأصلي ليصبح دخل مناسب لخوارزمية التشفير ومن ثم تقوم خوارزمية التشفير بتشифر النص الأصلي بالاعتماد على مفتاح (Key)، أي أن دخل خوارزمية التشفير هو النص الأصلي بعد أن تتم معالجته ومفتاح التشفير وخرجها هو النص المشفر.

يتم إرسال النص المشفر إلى الطرف المستقبل B ليقوم بعدها بإدخال النص المشفر إلى خوارزمية فك التشفير، ونلاحظ أن دخل خوارزمية فك التشفير هو النص المشفر والمفتاح وخرج هذه الخوارزمية هو النص الأصلي (أي hello).

عندما يكون مفتاح التشفير مشترك بين المرسل والمستقبل عندها يدعى بالتشفير المتناظر (عملية التشفير وفك التشفير تتم باستخدام نفس المفتاح).

عندما تكون مفاتيح المرسل مختلفة عن مفاتيح المستقبل، ويكون لكل منها روح من المفاتيح (مفتاح خاص ومفتاح عام) عندها يدعى هذا النوع بالتشفير غير المتناظر. (عملية التشفير تتم باستخدام المفتاح العام للمرسل وعملية فك التشفير تتم باستخدام المفتاح الخاص للمستقبل أي أن مفتاح التشفير يختلف عن مفتاح فك التشفير).

التشفير المتناظر يقسم إلى نوعين:

○ حديث

○ تقليدي

مكتبة الحكمة

اللادفه - مدخل اسيرو - مدخل باب السكن الجامعي
0993 508 666 & 0937 508 666 & 041/2439 666

الجلسة الاولى

- التشفير الحديث يقسم إلى نوعين:

- كتّي: مثل: DES, AES

- سلسلى: مثل: RC4

- التشفير التقليدي يقسم إلى نوعين:

- مشفرات التبديل: مثل: المشفر الجمعي، الضربى

- مشفرات تغيير الموضع: مثل: RailFence

سوف نبدأ مع مشفرات التبديل (ضمن المشفرات التقليدية) والتي لا تقدم مستوى الأمان المطلوب ويمكن كسرها بسيولة.

مكتبة الحكمة

اللادفه - مدخل اسيرو - مقابل باب السكن الجامعي

0993 508 666 & 0937 508 666 & 041/2439 666

الجلسة الاولى



التشифير الجمعي:

لدينا النص الأصلي الذي نريد تشفيره ولدينا مفتاح التشفير.

النص يتبع إلى الأبجدية معينة، حيث الأبجدية التي سوف نتعامل معها هي محارف اللغة الإنجليزية (يمكن أن تكون الأبجدية مثلاً محارف اللغة والأرقام من 0 إلى 9).

في هذه الطريقة يعطى كل حرف من أحرف الأبجدية قيمة رقمية تُسند إليه حسب ترتيبه في سلسلة الحروف، كما في الجدول التالي:

Plain	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
Value	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

مفتاح التشفير هو رقم ضمن المجال $[1, n]$ حيث n هي طول الأبجدية (في حال أن الأبجدية هي اللغة الإنجليزية تكون $n = 26$).

للقيام بعملية التشفير نأخذ قيمة كل حرف من النص ونجمعها مع مفتاح التشفير ليكون الناتج هو الحرف المقابل في النص المشفى.

$$C = (P + K) \bmod n$$

النص المشفر = (النص الأصلي + مفتاح التشفير) باقي القسمة على 11

قمنا بأخذ باقي القسمة على 26 (حيث 26 هي عدد أحرف الأبجدية) لنبقى ضمن المجال الصحيح، لأن بفرض لدينا قيمة المحرف 15 ولدينا المفتاح هو 22 عندها ناتج الجمع 37 وهي قيمة خارج المجال (لأن الأبجدية تحوي 26 حرف فقط).

وبكلام آخر يمكن التعبير عن مبدأ عمل الخوارزمية وفق التالي:

المشفى الجمعي يكون دخله سلسلة من المحارف (النص الأصلي P) ويكون لدينا مفتاح تشفير K وخرجته هو سلسلة من المحارف (النص المشفر C).

مبدأ عمله سهل حيث نقوم باستبدال كل حرف بالحرف الذي يليه وبين مقدار K أي من أجل $C = P + K$ يمكننا تمثيل العملية بـ إزاحة وحيدة أي:

مكتبة المحمية

اللادهـ - مدخل إسبرو - مقابل باب السكن الجامعي

0493 508 666 & 0937 508 666 & 041/2439 666

الجلسة الأولى



ملاحظة هامة: النص الأصلي نكتبه بالحرف صغيرة ولكن النص المشفر نكتبه بالحرف

مثل: بفرنس $P = \text{zed} \& K = 7$

$$c_1 = (z + 7) = (25 + 7) \bmod 26 = 32 \bmod 26 \approx 6 = G$$

$$c_2 = (e + 7) = (4 + 7) \bmod 26 = 11 \bmod 26 \approx 11 = L$$

$$c_3 = (d + 7) = (3 + 7) \bmod 26 = 10 \bmod 26 \approx 10 = K$$

$$\rightarrow C = "GLK"$$

كيف حصلنا على القيم السابقة؟

لأخذ مثلاً المحرف z والمفتاح هو 7، المحرف z قيمته هي 25 (من الأبجدية abcd...z نبدأ بالعد من a أي 0 وهي 1 وهكذا) ثم نقوم بجمع القيمة 25 مع المفتاح 7 فنحصل على الرقم 32 ونقوم بعدها بأخذ باقي قسمته على 26 وهي 6، نبحث عن المحرف الذي قيمته هي 6 فنلاحظ أنه

ملاحظة: لو كان عدد المحارف في الأبجدية المستخدمة هو 36، تقوم عندها بأخذ باقي القسمة على 36.

عندما $k=3$ يدعى التشفير الجمعي بـمشفر فيسر.

مثال 2: ليكن لدينا النص الأصلي هو zend ومفتاح التشفير هو 3 فيكون النص المشفر هو

Plain text	z	e	n	d
Value	25	04	13	03
+3	2	07	16	06
Cipher text	C	H	Q	G

$$\rightarrow C = CHQG$$

مكتبة المحمية

الادهه - مدخل اسيرو - مقابل باب السكن الجامعي

0993 508 666 & 0937 508 666 & 041/2434 666

الجلسة الاولى

أمتى:

```

>> AdditiveCipher("ahmad", 4)
ans = ELQEH
>> AdditiveCipher("ahmad", 10)
ans = KRWKN
>> AdditiveCipher("zein", 16)
ans = FUYD
>> AdditiveCipher("informationsecurity", 23)
ans = FKCLOXJQFLKPBZROFQV
>> AdditiveCipher("informationsecurity", 17)
ans = ZENFIRDZFEJVTLIZKP
>> AdditiveCipher("informationsecurity", 2)
ans = KPHQTICOVKQPUGEWTKVA
>> AdditiveCipher("informationsecurity", 0)
ans = INFORMATIONSECURITY
    
```

نلاحظ عندما $k = 0$ يكون النص المشفر ذاته النص الأصلي (عملياً لم تتحقق أي تشفير للرسالة الأصلية).

فك التشفير الجمعي:

لدينا النص المشفر ولدينا مفتاح التشفير وفريدة لتجد النص الأصلي.
عملية فك التشفير هي عملية معاكسة لعملية التشفير.
لقيام بعملية فك التشفير الجمعي نقوم بحساب النظير الجمعي لمفتاح التشفير.

النظير الجمعي لمفتاح التشفير = - مفتاح التشفير

$$\text{مفتاح التشفير} + (\text{النظير الجمعي له}) \Leftrightarrow 0 = (k + -k = 0)$$

$$P = (C - K) \bmod n$$

النص الأصلي = (النص المشفر - مفتاح التشفير) باقي القسمة على n

مثال:

لفك تشفير الرسالة "C = "GLK" باستخدام المفتاح 7

$$P_1 = (G - 7) = (6 - 7) \bmod 26 = -1 \bmod 26 = 25 = z$$

$$P_2 = (L - 7) = (11 - 7) \bmod 26 = 4 \bmod 26 = 4 = e$$

$$P_3 = (K - 7) = (10 - 7) \bmod 26 = 3 \bmod 26 = 3 = d$$

$$\rightarrow M = "zed"$$



ملاحظة:

من أجل P_1 فإن الناتج عدد سالب

ولكن كيف يتم حساب باقي قسمة عدد سالب x على عدد n (في المثال السابق 26)

???

- قواعد حساب $x \bmod n$ -

لنفرض أن $n = 26$

لدينا 3 حالات وهي:

$$1. 0 \leq x < 26$$

$$x \bmod 26 = x$$

$$7 \bmod 26 = 7$$

$$2. x \geq 26$$

يمكن كتابة x وفق الشكل التالي

بما أن $26 \geq x$ عندها x هي من مضاعفات الـ 26 إضافة إلى قيمة b والذي

يمثل مقدار الزيادة عن أحد المضاعفات (b حكماً أقل من 26).

$$(30 = 26 * 1 + 4 ; a = 1 \& b = 4)$$

$$x = a * 26 + b$$

$$x \bmod 26 = (a * 26 + b) \bmod 26$$

$$= (a * 26) \bmod 26 + b \bmod 26$$

$$= 0 + b = b$$

مكتبة الحكمة

اللاذقية - مدخل إسيبرو - مقابل باب السكن الجامعي

0993 508 666 & 0937 508 666 & 041/2439 666

الجلسة الأولى



$$30 \bmod 26 = (26*1 + 4) \bmod 26 = 4$$

$$60 \bmod 26 = (26*2 + 8) \bmod 26 = 8$$

3. $x < 0$

$$x \bmod 26 = 26 - (|x| \bmod 26)$$

$$-7 \bmod 26 = 26 - (7 \bmod 26) = 26 - 7 = 19$$

$$-1 \bmod 26 = 26 - (1 \bmod 26) = 26 - 1 = 25$$

$$-100 \bmod 26 = 26 - (100 \bmod 26) = 26 - 22 = 4$$

أو يمكننا استخدام الآلة الحاسبة 😊

أمثلة على فك التشفير الجمعي:

>> AdditiveDecryption("ELQEH", 4)

ans = ahmad

>> AdditiveDecryption("KRMKN", 10)

ans = ahmad

>> AdditiveDecryption("PUYD", 16)

ans = zein

>> AdditiveDecryption("FKCLOXJQFLKPZROFQV", 23)

ans = informationsecurity

>> AdditiveDecryption("ZENFIRDKZFEJVTLIZKP", 17)

ans = informationsecurity

>> AdditiveDecryption("KPHQTCOVKQPUGEWTKVA", 2)

ans = informationsecurity

مكتبة المحمية

اللادفه - مدخل اسيرو - مقابل باب السكن الجامعي

0993 508 666 & 0937 508 666 & 041/2439 666

الهجمات على المشفر الجماعي:

يعتبر المشفر الجماعي سهل المهاجمة، ويتم كسره بسهولة، ولدينا طريقة لمحاجمة:

ـ الهجوم الأعمى أو الهجوم الشامل (Bruteforce): وفيه يقوم المهاجم باختراض الرسالة المشفرة ويقوم بتجربة كل المفاتيح الممكنة في خوارزمية فلتر التشفير حتى يحصل على نص ذو معنى.

إليه عمل الهجوم الأعمى: بفرض لدينا نص مشفر BODDK، نقوم أولاً بتجربة المفتاح $A = k$ ونحاول فك تشفير النص باستخدام هذا المفتاح ونرى إذا كان النص الناتج ذو معنى وإذا لم يكن كذلك نأخذ المفتاح $B = k + 1$ ونحاول فك تشفير النص وهكذا (أي أننا نقوم بتجربة كل الاحتمالات الممكنة للمفتاح).

ملحوظة: في المثال السابق المفتاح الذي يفك تشفيره إلى نص ذو معنى هو 10.

ـ الهجوم الإحصائي: يعتمد على تكرارات الأحرف.

إليه عمل الهجوم الإحصائي: يعتمد الهجوم الإحصائي على جدول تكرارات الأحرف، فمثلاً أكثر حرف في أي نص يكرر بما في اللغة الإنكليزية هما e ثم t لذلك سنستفيد من هذه المعلومة في مهاجمة الرسالة المنشورة.

مثال: بفرض أن المهاجم اعترض الرسالة التالية BODDK، الهجوم المناسب على هذه الرسالة هو الهجوم الإحصائي وذلك بسبب وجود حرف مكرر في الرسالة، أكثر حرف مكرر في اللغة الإنكليزية هو الحرف e لذلك سنفترض أن أكثر حرف مكرر في الرسالة المشفرة (أي الحرف D) ناتج عن تشفير الحرف e.

$$e \rightarrow D$$

إذا افترضنا أن الكلام السابق صحيح (أي الحرف D ينتج عن e) عندما سيكون مفتاح التشفير هو:

$$\text{key} = (D - e) \% 26 = (3 - 4) \% 26 = 25$$

نجرب فك تشفير الرسالة السابقة باستخدام المفتاح 25 فنحصل على $p = cpeel$ وهو ليس نص ذو معنى.

مكتبة المحميّة

اللادفه - مدخل إسپرو - مقابل باب السكن الجماعي

0993 508 666 & 0937 508 666 & 041/2439 666

الجلسة الأولى

إذاً فإن المحرف D ليس ناتجاً عن تشفير المحرف e، نجرب الحرف التالي الأكثر تكراراً في اللغة الإنجليزية وهو المحرف t:

$$t \rightarrow D$$

يكون المفتاح الناتج في هذه الحالة:

$$key = (D - t) \% 26 = (3 - 19) \% 26 = 10$$

نحاول فك تشفير الرسالة السابقة باستخدام المفتاح 10 فنحصل على $p = retta$ وهو بالفعل نفس ذي معنى، إذاً المفتاح 10 هو المفتاح الصحيح.

ملاحظة:

نأخذ باقي القسمة على 26 لأن طول الأبجدية (عدد مهارف الأبجدية) هو 26.
لو كانت الأبجدية هي: مهارف اللغة الإنجليزية + الأرقام (0 - 9) كنا أخذنا باقي القسمة على 36 لأن عدد المهارف والأرقام (طول الأبجدية) هو 36، مثلاً: في حال

ورود هذه الأبجدية في الامتحان:

{ a ... z , * , @ , , / }

خوارزمية التشفير (باستخدام Pseudo Code)

```
function additiveEnc(String plaintext, char key) {
    1. ciphertext = "";
    2. keyAsNumber = get order of key;
    3. foreach character in plaintext:
        a. c = get order of current character;
        b. sumAsNum = (c + keyAsNumber) % 26;
        c. currentEncryption = convert sumAsNum to
            character;
        d. ciphertext += currentEncryption;
    4. return ciphertext
}
```



يمكنا كتابة كود خوارزمية التشفير باستخدام لغة الجافا:

```
public String Encrypt (String plaintext, int key) {
    String alphabetic = "abcdefghijklmnopqrstuvwxyz";
    String ciphertext = "";
    String lowercased = plaintext.toLowerCase();
    int sum, c;
    for (int i = 0; i < plaintext.length(); i++) {
        c = alphabetic.indexOf(lowercased.charAt(i));
        sum = (c + key) % 26;
        ciphertext += alphabetic.charAt(sum);
    }
    return ciphertext.toUpperCase();
}
```

شرح الكود السابق:

قبل البدء بالشرح سنعرف على التابعين `charAt()` و `indexOf()` - `indexOf()`: نمرر له محرف ويعين لنا موقع المحرف في串ة النصية.

مثال: بفرض لدينا:

`String s = "hello";`

`s.indexOf(h); -> 0`

`s.indexOf(l); -> 2`

(أي أنه لم يجد المحرف المطلوب) - `s.indexOf(H); -> -1;`

- `charAt()`: نمرر له رقم ويعين لنا المحرف الموجود في هذا الموقع.

مثال:

`s.charAt(0); -> h`

مكتبة المكتبة

اللادهه - مدخل إسبرو - مقابل باب السكن الجماعي

0993 508 666 & 0937 508 666 & 041/2439 666

الجلسة الأولى



الآن سنقوم بشرح الكود السابق:

أولاً نقوم بتبيين المتحول alphabetic وقيمة هي أحرف الأبجدية كاملة.

نقوم بتهيئة النتيجة ciphertext التي سنقوم بإعادتها من التابع بسلسلة فارغة.

نقوم بتحويل النص الذي نريد تشفيره إلى أحرف صغيرة (لأنه من الممكن أن يقوم المستخدم مثلاً بإدخال Best بدلاً من best وعندها لن يعمل التابع indexOf بشكل صحيح) ولذلك عن طريق التابع `toLowerCase()`.

نستخدم حلقة for للمرور على المحارف الموجودة في السلسلة والحصول عليها من خلال التعليمية `charAt(i)` التي ستعيد المحرف الموجود في الموقع 0 ثم 1 ثم 2 وهكذا، حيث في تلك هو أن حلقة for تبدأ من الـ 0 وتنتهي عند `plaintext.length()` (طول السلسلة) أي إذا كان لدينا الكلمة hello فهي ستبدأ من الـ 0 وتنتهي عند 4 لأن `5 = plaintext.length` في هذه الحالة.

القيمة التي يعيدها التابع `charAt(i)` ستمرر مباشرةً إلى التابع `indexOf()` وذلك للحصول على موقع المحرف في الأداة، مثلاً بفرض أن النص هو `hello` عندها في أول تكرار من الحلقة ستكون التعليمية على الشكل التالي:

```
c = alphabetic.indexOf(lowercased.charAt(0));  
c = alphabetic.indexOf('h');  
c = 7;
```

وبهذا نكون قد حصلنا على القيمة العددية للمحرف، الان ننتقل إلى الخطوة التالية وهي جمع هذه القيمة مع المفتاح (بفرض هو 1) وحساب باقي القسمة على 26.

```
sum = (c + key) % 26;
```

```
sum = (7 + 1) % 26;
```

$$\text{sum} = 8;$$

الخطوة الأخيرة هي أننا نستخدم التابع `indexOf` لكي نحصل على المحرف الموجود في الموقع 8 وهو أ ونضيفه إلى المتحول `ciphertext`.

عند انتهاء تنفيذ الحلقة نقوم بإعادة النص المشفر ولكن نستخدم التابعtoUpperCase لكي نحصل على النص المشفر باحرف كبيرة.

مكتبة الكوفة

اللادفه - مدخل إسپيرو - مقابل باب السكن الجامعي

0993 508 666 & 0937 508 666 & 041/2439 666

الجلسة الاولى

خوارزمية فك التشفير (باستخدام Pseudo Code)

```
function additiveDec(String ciphertext, char key) {
    1. plaintext = "";
    2. keyAsNumber = get order of key;
    3. foreach character in ciphertext:
        a. c = get order of current character;
        b. sumAsNum = (c - keyAsNum) % 26;
        c. currentDecryption = convert sumAsNum to
            character;
        d. plaintext += currentDecryption;
    4. return plaintext;
}
```

يمكننا كتابة كود خوارزمية فك التشفير باستخدام لغة الجافا:

```
public String Decrypt (String ciphertext, int key) {
    String alphabetic = "abcdefghijklmnopqrstuvwxyz";
    String plaintext = "";
    String lowercased = ciphertext.toLowerCase();
    int sum, c;
    for (int i = 0, i < ciphertext.length(); i++) {
        c = alphabetic.indexOf(lowercased.charAt(i));
        sum = (c - key);
        if (sum < 0) { sum = sum + 26; }
        else { sum = sum % 26; }
        plaintext += alphabetic.charAt(sum);
    }
    return plaintext;
}
```

مكتبة المكتبات

اللادقية - مدخل إسيرو - مقابل باب السكن الجامعي

0993 508 666 & 0937 508 666 & 041/2439 666

الجلسة الأولى

نلاحظ بعض الاختلافات عن خوارزمية التشفير وهي:

- أولاً الوسيط الممر للتابع هو **ciphertext** وليس **plaintext** (وعلى الرغم من أنه يمكننا استخدام مثلاً الاسم **x** للوسيط وإن تؤثر في عمل التابع ولكن تعتبر أسماء المتغيرات مهمة للدلالة على ما تحاول القيام به لذلك من الأفضل انتقاء اسم للمتغير يعبر عن القيمة التي سيحتويها)، ينطبق تغيير الأسماء على عدة متغيرات في التابع منها المتتحول في شرط التوقف الموجود في حلقة **for** ومنها السلسلة التي نهيتها في بداية التابع لكي تقوم بارجاعها و ... إلخ.

- أصبحت العلاقة التي نحسب منها المتتحول **sum** إشارتها هي الطرح وليس الجمع (وذلك لأننا نقوم بفك التشفير)، وأيضاً نلاحظ بعد حساب المتتحول **sum** أننا نقوم بمعالجة حالة إذا كان أصغر من **0**.

تمرين: قم بإضافة الكود اللازم لكي نستطيع الهجوم على نص مشفر باستخدام طريقة الهجوم الأعمى (Bruteforce)

الحل:

بساطة يمكننا وضع حلقة **for** تمر على كل المفاتيح الممكنة (من 1 وحتى 25) ومن أجل كل مفتاح تقوم باستدعاء خوارزمية فك التشفير، أي سيكون الكود على الشكل التالي:

```
String ciphertext = "BODDK";
For (int i = 1; i < 26; i++) {
    System.out.println(Decrypt(ciphertext, i));
}
```

أنجزت بالأسلاك

مكتبة الحكمة

اللاديفه - مدخل إسپرو - مقابل باب السكن الجامعي

0993 508 666 & 0937 508 666 & 041/2439 666

الجلسة الأولى