جامعة قاريونس - كلية تقنية المعلومات

مشروع تخرج بعنوان

خوارزميات التشفير

مقدم من الطالبتين:

لبنى عمر الكيخيا 16614

أسماء أبو بكر العشيبي 16790

تحت إشراف:

د . فرج المؤدب

تم إعداد هذا المشروع استكمالا لمتطلبات الحصول على درجة البكالوريوس في كلية تقنية المعلومات.

فصل الربيع 2006/2007

يا معشر الجن والإنس إن استطعتم أن تنفذوا من أقطار السماوات والأرض

فانفذوا لا تنفذون إلا بسلطان (31) فبأي آلاء ربكما تكذبان (32)

إ هـد ا ء

أهدي هذا الجهد المتواضع والذي أراه خطوة صغيرة أولى على درب طويل؛ ممهد بعضه وبعضه لا يزال ملئ بالعثرات... العثرات التي نأمل أن يكون عملنا هذا تذليلاً لها... أهدي هذا المشروع إلى والدي ووالدتي وأساتذتي.

أســمــاء

إ هـ د ا ء

اهدي نجاحي هذا إلى ينابيع الحنان المتدفق اللذان لا يملان من عطائهم...

إلى أبي الذي لم يبخل علي بشئ من اجل دفعي إلى طريق السعادة والنجاح... إلى أمي التي وفرت لي كل سبل الراحة لكي اصل إلى أفضل ما يمكن الوصول إليه.

كما اهديه إلى أخوتي الأعزاء اللذين عاشوا معي الحياة بحلوها ومرها...

وإلى كل من ساهم معي بوقته وجهده وكان سندا وعونا لي.

وأخيراً اهدي مشروعي هذا إلى خطيبي العزيز.

لبـــنـــى

الشكر والتقدير

بداية كان الدخول إلى عمل بهذا الحجم أشبه بالدخول لمتاهة لا علامات فيها ولا من مرشد . وحقيقة لم يكن ليتأتى لنا الخوض فيها دون ضوءٍ هاد ومصباح منير

د- فرج المؤدب

أ- عمران الأوجلي

أ- سمير بن علي

أ- وديع الأطرش

كانوا أكثر من ضوء هادي وأكثر من مصابيح أضاءت لنا العتمة وأبعدت بحنو وتعاطف كل هواجس المبتدئ عنا ومخاوف التائه.

الكلام لا يفي هؤلاء الأساتذة المرشدين حقهم لذا سنكتفي بالقول

شكراً . . . شكراً . . . شكراً .

فـهـرس المحـتـويـات

الموضوع **رقم الصفحة**

فـهـرس الأشكـال

الشكل **رقم الصفحة**

فـهـرس الجـداول

الجدول **رقم الصفحة**

التقديم

خلال السنوات الأخيرة شهدت صناعة الحواسيب تطورات كبيرة من حيث لغات البرمجة والبرامج التطبيقية وغيرها، كما أن استخدام الحاسوب دخل في عدة مجالات منها التجارية والمصرفية و غيرها من المجالات، كل هذه المجلات تحتاج إلى حماية بياناتها من التجسس والتطفل وهذا أدى إلى ضرورة استخدام برامج التشفير.

أسواق تقنيات وبرامج التشفير شهدت انتعاشاً مذهلاً بعد أن سُمح للشركات التجارية ببيع هذه التقنيات لعامة الناس بعدما كانت محصورة للاستخدامات العسكرية والحكومية لسنوات طويلة، وقد اتخذ هذا القرار لدعم الجانب الأمني لمجال التجارة الإلكترونية.

ونظراً إلى أن مناهجنا التدريسية والتعليمية ومكتباتنا العربية لا تعطي هذا الموضوع أهمية كبيرة من حيث توفير الكتب والبرامج اللازمة لفهم واستخدام تقنيات التشفير المختلفة، قمنا ببرمجة منظومة تقوم بتشفير الملفات باستخدام بعض خوارزميات التشفير المعروفة، كما قمنا في هذا الكتاب بشرح هذه الخوارزميات وبعض خوارزميات التشفير الأخرى مع توضيح كيفية عملها.

الفصل الأول

مقدمة عن علم التشفير

1.1 مقدمة

علم التشفير Cryptography أحد المجالات المهمة والمعقدة في الحاسوب، وقد ازداد الطلب على تقنيات التشفير في البرامج التي يستخدمها العامة من الناس مع انتشار الانترنت قبل عشر سنوات بسبب الحاجة لنقل المعلومات بسرية وخصوصية على شبكات عامة قابلة للاعتراض والتجسس.

هنالك الكثير من خطط التشفير Encryption Schemes، والتي تسمى أيضاً طرق أو أساليب التشفير، ولكل منها نقاط قوتها وضعفها واستخداماتها، وهذه الأساليب هي أفكار عامة وليست تطبيقات عملية للتشفير، أما التطبيقات العملية للتشفير فتسمى خوارزميات التشفير Encryption Algorithms.

في الماضي كان استخدام التشفير محصوراً على الحكومات والجيوش فقط، أما في الوقت الحاضر فقد أصبح استخدامه متوفراً لأي شخص، حيث يعتبر من أهم وسائل الحفاظ على سرية المعلومات، الخصوصية Privacy، التحكم في الوصول Access control، الفواتير الالكترونية Electronic payments، كذلك يستخدم في مجال التأكد من هوية الافراد Authentication، والعديد من المجالات الأخرى.

مصطلح Cryptography مشتق من الكلمتين اليونانيتين kryptós و gráfo وهما تعنيان الكتابة المخفية حيث أن kryptós تعنى مخفية و gráfo تعنى كتابة.

التشفير يستخدم بشكل كبير في الأغراض العسكرية، لذلك فإن العديد من التطورات التي حدثت في هذا المجال جاءت لتلبية احتياجات الجيوش، وأكثر آلات التشفير شيوعاً هي آلة إنقما Enigma التي استخدمت من قبل الجيش الألماني في الحرب العالمية الثانية والتي تم فك تشفيرها بواسطة محللي التشفير للجيش الإنكليزي والبولندي، شكل(1) يوضح آلة إنقما.

شكل(1): آلة إنقما (Enigma).

في عصر الالكترونيات الحالي ظهرت العديد من التطبيقات التي تستخدم تقنية التشفير، وأكثرها شيوعاً هو استخدام بطاقات الائتمان Credit cards والبطاقات الذكية Smart cards، شكل(2) يوضح بطاقة ائتمان مع بطاقة ذكية.

شكل(2): بطاقة ائتمان مع بطاقة ذكية.

1.2 دور العرب في علم التشفير

بدأ استعمال التشفير والكتابة السرية منذ بدء الحضارة الإنسانية، وقد استخدمها قدماء المصريين منذ حوالي 1900 سنة ق.م، كما استخدمها الإغريق القدماء.

كذلك قام يوليوس قيصر بابتكار طريقة تشفير بسيطة تعتمد على الإزاحة أو التبديل، حيث قام بإزاحة كل حرف ثلاثة خانات بعده، أي استبدل كل حرف بالحرف الذي يليه بثلاثة خانات.

ويقال أن التشفير كعلم مؤسس منظم لم يبدأ إلا عند العرب بعد بزوغ الحضارة الإسلامية العربية. و كما ورد في الموقع الالكتروني www.mawsoah.net أن المؤرخ الشهير لعلم التشفير ديفيد كاهن David Kahn قال في كتابه الشهير (كاسرو الشفرات) ما نصه:

"لم نجد في أي من الكتابات التي نقبنا عنها أي أثر واضح لعلم التشفير...علم التشفير الذي يشمل علم وضع الشفرة وعلم تحليلها (أي كسرها) ظهر في القرن السابع الميلادي، حيث ولد علم التشفير بشقيه بين العرب، فقد كانوا أول من اكتشف طرق تحليل التشفير وكتابته وتدوينه. إن هذه الأمة التي انبثقت من الجزيرة العربية في القرن السابع الميلادي، والتي انتشرت فوق مساحات شاسعة من العالم، أخرجت، وبسرعة، إحدى أرقى الحضارات التي عرفها التاريخ في ذلك الوقت."

في سنة 1992ف قام الدكتور إبراهيم القاضي بكتابة مقالة عن دور العرب في مجال التشفير ذكر فيها دور الفيلسوف العربي أبو يوسف يعقوب بن إسحاق الكندي (185-260هـ، 801-874ف)، شكل(3)، وهو أعظم العلماء العرب في هذا المجال، ضمت مؤلفاته الكثيرة أول كتاب معروف في علم التشفير "رسالة في استخراج المعمَّى"، استعرض في هذا الكتاب قواعد علم التشفير وأسرار اللغة العربية، واستخدم لأول مرة في التاريخ مفاهيم الإحصاء في تحليل النصوص المشفرة، وذلك قبل كتابات باسكال Pascal وفيرمات Fermat، التي يعتبرها مؤرخو الرياضيات الغربيين بداية علم الإحصاء والاحتمالات، بحوالي 800 سنة.

شكل (3): العالم العربي أبو يوسف الكندي

وكما ورد في الموقع الالكتروني www.ma3refah.net مخطوطة الكندي محفوظة في مكتبة آيا صوفيا ضمن خزائن المكتبة السليمانية في اسطنبول تحت رقم (4832)، وهى تعود إلى أوائل القرن الثالث الهجري وتضم 232 ورقة، وقد بقيت مجهولة لا تذكر في المراجع حتى عام 1985ف، حيث نجح الدكتور مراياتى، يرافقه الباحثان يحيى علم والدكتور محمد الطيان، في العثور عليها خلال بحث عن المخطوطات العربية في المكتبة.

المخطوطة مبنية منطقياً من فصول متسلسلة، يتناول الأول سبل تحليل التشفير، ويتطرق الثاني إلى طرق التشفير، ويعالج الثالث مناهج استخراج بعض أنواع التشفير، في حين يشرح الفصلان الأخيران مسألة دوران الحروف ومراتبها واقترانها.

ذكر الدكتور إبراهيم القاضي في مقالته أيضاً أن مصطلح "التشفير" المستخدم في أيامنا هذه للدلالة على إخفاء المعلومات هو مصطلح عربي الأصل. مع أن العديد من الناس يعتقدون أن كلمة "التشفير" وافدة من اللغات الأوروبية كترجمة للكلمة "Cipher"، ولكن في الحقيقة إن هذه الكلمة جاءت أصلاً من اللغة العربية ولكن بمعنى آخر لكلمة "الصفر". فكما هو معلوم أن العرب هم أول من اخترع مفهوم الصفر واستخدموه في الحساب، وهو ما لم يكن الأوربيون يعرفونه في القرون الوسطى، و كان مفهوم الصفر جديداً وغريباً لدرجة أنهم أخذوه بنفس الاسم فأسموه"Cipher"، ولأن مفهوم الصفر كان في منتهى التعقيد والغموض فقد

صاروا يستخدمون كلمة "Cipher" للدلالة على الأشياء غير الواضحة. ومن هنا تطور استخدام كلمة "Cipher"في جميع اللغات الأوروبية تقريباً لتعني إخفاء المعلومات.

ازدهر علم التشفير عند العرب كذلك في القرن الثالث عشر الميلادي، نتيجة لأسباب حضارية وعسكرية وسياسية، وظهرت في تلك الفترة مؤلفات كثيرة في علم التشفير منها: كتاب ابن دنينير (مقاصد الفصول المترجمة عن الترجمة)، وكتاب ابن عدلان المؤلف للملك الأشرف، وكتاب (مفتاح الكنوز في إيضاح المرموز) الذي كتبه علي بن الدريهم.

ثم ظهرت بعد ذلك مؤلفات في علم التشفير في أوروبا، بترجمات أو اقتباسات مما ترك العرب، مع زيادة وتطوير على أيدي بعض العلماء الأوروبيين. وبعد ذلك نشط هذا العلم قبيل الحرب العالمية الأولى، واستمر هذا النشاط العلمي يتزايد إلى يومنا هذا.

في عام 1949ف نشر العالم كلود شانون Claude Shannon بحثًا مبتكرًا عن نظرية الاتصالات السرية، ووضع بذلك الأساس النظري الرياضي لعلم التشفير الحديث.

في عام 1975ف ظهر نظام التشفير غير المتناظر (نظام تشفير بمفتاحين أحدها سري والآخر معلن). وفي عام 1977ف، ولأول مرة في التاريخ، تم اعتماد الخوارزمية DES (Data Encryption Standard) في الولايات المتحدة الأمريكية كخوارزمية قياسية لتشفير البيانات.

رغم استخدام التشفير في الحضارات القديمة، ورغم قدم جذور التشفير وعلم تحليل التشفير، إلا أن البحوث والدراسات تشير إلى تباطؤ نسبي لنمو هذين العلمين وذلك بالمقارنة مع العلوم الأخرى. ويعزى السبب في ذلك جزئيًا إلى ضيق نطاق استخدام التشفير قديماً حيث كان مقصورًا على المراسلات الدبلوماسية والعسكرية فقط، ولكن السبب الرئيسي يكمن في طوق السرية الذي كان مفروضًا على نشر وتداول الأفكار العلمية في هذا المجال.

هذا الأمر بدأ يشهد تحولاً كبيرًا؛ نتيجة ثورة المعلومات التي نعيشها والتقدم الكبير في الإلكترونيات والاتصالات والحواسيب، كذلك نتيجة الحاجة لأمن المعلومات في المجالات التجارية والمصرفية وشبكات الحاسوب وغيرها.

كما أدت نظرية شانون واختراع نظام التشفير غير المتناظر إلى نمو سريع ونشاط قوي في البحوث المتعلقة بعلم التشفير وعلم تحليل التشفير.

1.3 أهمية التشفير واستخداماته

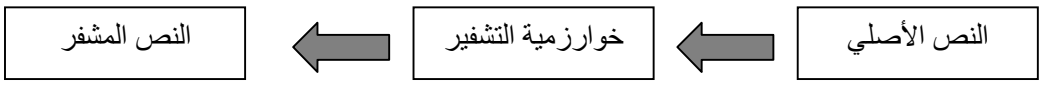
اكتسب علم التشفير وتطبيقاته أهمية بالغة منذ مطلع القرن العشرين، ففي منتصف السبعينيات استعمال التشفير تعدى الاتصالات والمراسلات العسكرية والدبلوماسية والأمنية ووصل إلى عدة مجالات وتطبيقات واستخدامات أُخرى، منها:

- في الصناعة والتجارة للمحافظة على الأسرار التجارية والعلمية والاختراعات والتصميمات والوضع المالي للمؤسسات وغيرها.
- في البث المرئي حيث يتم تشفير القنوات المرئية حتى لا يستطيع مشاهدتها إلا المشتركون الذين يدفعون اشتراكًا شهريًا مقابل المفتاح السري الذي يسمح بفك تشفير القنوات ومشاهدة البرامج.
- في المصارف للمحافظة على حسابات الزبائن وحمايتها من التلاعب أو الاختلاس خصوصاً مع تطور الخدمات المصرفية الإلكترونية.
- في شبكات الحواسيب والحواسيب الشخصية للمحافظة على المعلومات الحساسة، وحماية الملفات، وحماية الدخول إلى الشبكات عن طريق كلمة المرور الخاصة بكل مستخدم.
- في حماية الاتصالات السلكية واللاسلكية وهواتف السيارات من الالتقاط والتصنت والإطلاع على أسرار الآخرين الشخصية والعائلية.

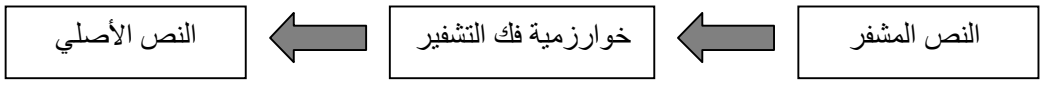
- في الكشف عن اللغات القديمة البائدة حيث كان لعلم تحليل التشفير أكبر الأثر في الكشف عن رموز اللغة الهيروغليفية في مطلع القرن التاسع عشر.

1.4 مصطلحات متعلقة بالتشفير

- التشفير Encryption يقصد به عملية تحويل المعلومات إلى صيغة غير مفهومة بهدف الحفاظ على سريتها وموثوقيتها عند إرسالها عبر قناة غير آمنة. أي أنه باستخدام تقنيات التشفير القوية المعلومات الحساسة والقيِّمة يتم حمايتها من الأشخاص الغير مسموح لهم بالإطلاع عليها مثل المخترقين والجواسيس والمتطفلين.



- فك التشفير Decryption وهى العملية العكسية للتشفير، أي هي عملية تحويل المعلومات الغير مفهومة إلي معلومات يمكن قراءتها وفهمها.



- إخفاء المعلومات Steganography هو علم يدرس كيفية حجب وإخفاء المعلومات بحيث تكون الرسالة غير مرئية Invisible.
- تحليل التشفير Cryptanalysis هو علم (أو فن) دراسة كيفية فك النص المشفر بدون معرفة تقنية التشفير المعتمدة.
- مفكك الشفرة Cryptanalyst وهو الشخص المتخصص في فك أو تحليل التشفير.

- علم التشفير وفكه Cryptology هذا العلم يجمع بين دراسة تشـفير النصـوص Cryptography ودراسة طرق تحليل شفرة النصوص Cryptanalysis. كلمة Cryptology تعنى دراسة الأسرار "the study of secrets".
- النص الأصلي Plaintext أو Clear text هو النص (الرسالة) المراد تشفيره.
- النص المشفر Cipher text أو Encrypted text هو النص الناتج عن تشـفير الـنص الأصـلي بإحدى طرق التشفير المعروفة.

1.5 نبذة مختصرة عن علم إخفاء أو حجب المعلومات Steganography

يختلف هذا العلم عن علم التشفير من حيث أن التشفير يجعل المعلومات مرئية ولكنها غير مفهومـة، بينمـا الحجب يجعل المعلومات غير مرئية للعيان، وهذه المعلومات من الممكن أن تكون مشفرة أو غير مشفرة.

إحدى الطرق التي كانت تستخدم في السابق لإخفاء المعلومات هي كتابة الرسالة بعصير الليمون، ولإظهار الكتابة نقوم بتسخين الورقة. بعد ذلك ظهر الحبر السري الذي استخدم لفترة في إخفاء المعلومات.

الآن بعد التطور الكبير الذي حدث في التكنولوجيا وأجهزة الحاسوب، ظهرت العديد من الطـرق لإخفـاء المعلومات باستخدام الحواسيب، مثل استخدام الصور أو الصوت لإخفاء الرسائل بداخلها.

1.5.1 طريقة إخفاء المعلومات داخل الصور Images

أي صورة موجودة في جهاز الحاسوب تتكون من مجموعة نقاط ضـوئية Pixels، كـل نقطـة ضـوئية تحتوي على لون من الألوان التي تكوِّن الصورة.

حجم النقاط الضوئية يختلف حسب دقة الصورة، فمن الممكن أن يكون حجمها 8 بت أو 24 بت أو غيرها، وكلما صغر حجمها كلما كانت الصورة أدق وأوضح، هذه النقاط الضوئية تخزن فـي جهـاز الحاسـوب كأرقام ثنائية Binary numbers. فإذا أردنا إخفاء رسالة داخل صورة معينة، نقـوم بتحويـل كـلاً مـن الصورة والرسالة إلى أرقام ثنائية، ومن ثم نستبدل أول رقم ثنائي من جهة اليمين لأول نقطة ضـوئية فـي

الصورة بأول رقم ثنائي من الرسالة المحولة، كذلك نستبدل أول رقم من جهة اليمين لثاني نقطة ضوئية بثاني رقم من الرسالة، وهكذا...

هذا التبديل سيحدث تغيير بسيط جدا في الألوان الموجودة داخل النقاط الضوئية، وبالتالي سيحدث تغيير طفيف جداً في الصورة لا يمكن ملاحظته بالعين المجردة.

يجب أن يتم التبديل من جهة اليمين حتى يحدث تغيير بسيط في اللون، بينما التبديل من جهة اليسار يغير اللون بأكمله.

مثال:

إذا كانت الصيغة الثنائية لجزء ما من الصورة المختارة هي:

```
11001001 00110101 00000010 11000011 00110110 01000101 10111111 00111000
```

والصيغة الثنائية لأول حرف في النص هي:

```
01101101
```

نقوم بوضع هذا الحرف داخل الصورة باستبدال أول بت من اليمين من كل نقطة ضوئية ببتات هذا الحرف بالترتيب، كالتالي:

```
11001000 00110101 00000011 11000010 00110111 01000101 10111110 00111001
```

نلاحظ أن أول بت لأول نقطة ضوئية تغير من1 إلى 0 وهذا سيحدث تغيير بسيط في درجة اللون الموجود في هذه النقطة الضوئية لن يتم ملاحظته بالعين المجردة.

بعد ذلك نرسل الصورة إلى الجهة الأخرى، التي تقوم بجمع الأرقام المخفية داخل الصورة للحصول على الرسالة الأصلية.

من عيوب هذه الطريقة أنه إذا تم اكتشاف الرسالة المخبأة داخل الصورة فسوف يتم معرفة محتواها بسهولة لأنها غير مشفرة، وعليه لضمان سرية الرسالة يتم تشفيرها قبل وضعها داخل الصورة وإرسالها.

يجب مراعاة النقاط التالية عند اختيار الصورة المراد إرسالها:

1. أن تكون الصورة مختارة بطريقة عشوائية.
2. ألا تكون صورة معروفة، مثلاً مأخوذة من الانترنت أو صورة لرسام مشهور، وذلك حتى لا يتم كشفها بسهولة.
3. أن تحتوي الصورة على عدة ألوان، أي لا تكون بيضاء أو لون واحد، حتى لا يكون التغيير المحدث فيها واضحاً.

1.5.2 طريقة إخفاء المعلومات داخل الصوت Sound

يتم إخفاء المعلومات داخل ملفات صوتية وذلك بعمل تغييرات بسيطة في هذه الملفات قبل إرسالها- كما في طريقة الصور- ولن يتم ملاحظة هذه التغييرات بواسطة السمع البشري.

الفصل الثاني

طرق التسفير المختلفة

هناك الكثير من الطرق المستخدمة للتشفير منها ما هو قديم ومنها ما هو حديث، وفي هذا الفصل نستعرض الأنواع المختلفة للتشفير.

2.1 التشفير بتبديل الحروف Substitution Cipher

في هذه الطريقة يتم تبديل كل حرف (أو حرفين أو كلمة) من النص الأصلي بحرف آخر (أو رمز أو رقم أو كلمة) بطريقة معينة، ومن الممكن تحديد هذه الطريقة باستخدام المفتاح السري.

التشفير بالتبديل يتم تمثيله بتكوين مصفوفة تحتوي على صفين وعدد من الأعمدة يساوي عدد الحروف الهجائية للغة المستخدمة في التشفير. يوضع في الصف الأول جميع الحروف الهجائية بشكل مرتب، ويوضع في الصف الثاني نفس العناصر ولكن بطريقة عشوائية. يتم تشفير الرسالة بأخذ كل حرف على حدة وتحويله إلى الحرف الذي يقابله في الصف الثاني من المصفوفة.

بشكل مبسط في طريقة التشفير بالتبديل حروف الرسالة الأصلية يتم استبدالها بحروف أخرى من أجل زيادة مستوى السرية.

هناك عدة طرق للتشفير بتبديل الحروف منها:

2.1.1 خوارزمية قيصر Caesar Cipher

أبسط أنواع طرق التبديل تنتج عن طريق تبديل أحرف الرسالة الأصلية بأحرف أخرى تبعد عنها مسافة ثابتة في الترتيب الأبجدي. أول من استخدم هذه الطريقة يوليوس قيصر.

قام القيصر بعمل إزاحة للأحرف ثلاث خانات، حيث استبدل الحرف A بالحرف D، والحرف B بالحرف E، والحرف C بالحرف F، وهكذا... كما موضح في الجدول(1):

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
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

جدول (1): تبديل الحروف بطريقة قيصر.

لنأخذ على سبيل المثال النص:

meet me after the party

ونقوم بتشفيره, حيث نقوم بتبديل كل حرف بثالث حرف بعده، كما هو موضح في الجدول(1)، إلى أن ينتج لنا النص المشفر:

phhw ph diwhu wkh sduwb

ولفك التشفير نقوم بإرجاع كل حرف ثلاث خانات قبله.

بعد ذلك تم استخدام هذه الطريقة مع تغيير مقدار الإزاحة حسب الاتفاق بين المرسل والمستقبل.

عيـــــــــــوب هذه الطريقة:

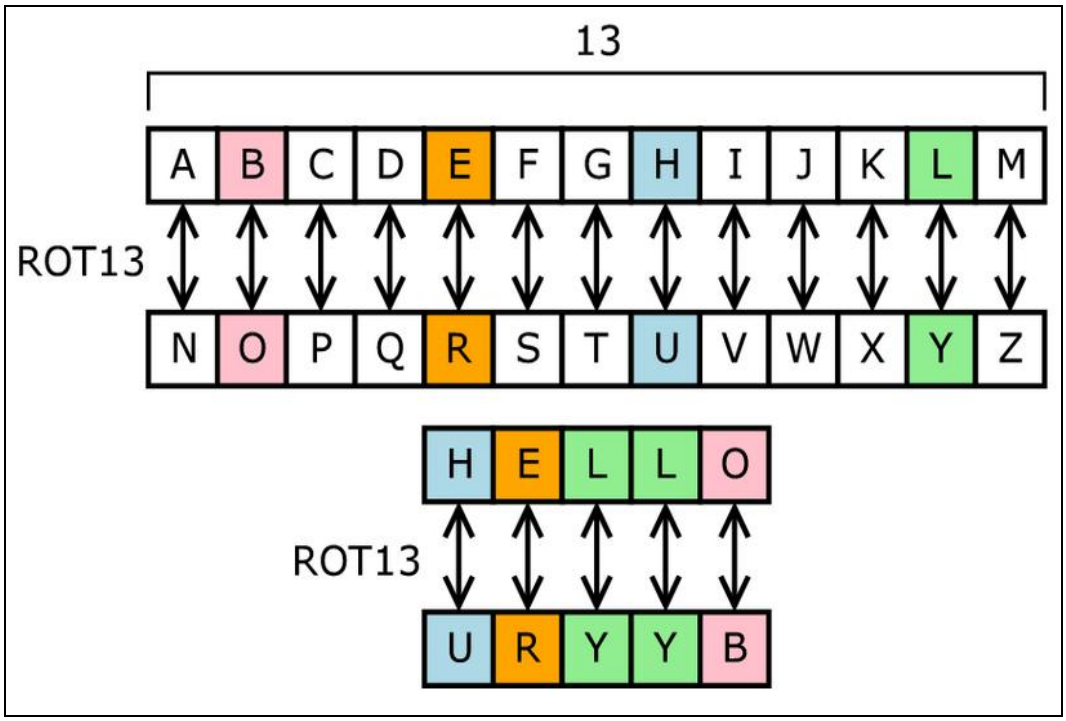
1. لو نظرنا إلى هذه الطريقة من جانب أمني لرأينا أنها سهلة الكسر، حيث انه لدينا 26 احتمال لكسر النص المشفر، وهو عدد الحروف الإنجليزية، أو بالأصح 25 احتمال لأن الحرف لا يساوي نفسه. ولنأخذ على سبيل المثال الحرف A لكسره نجرب كل الحروف ماعدا الحرف نفسه وهذه الطريقة معروفة لمحللي التشفير وتسمى طريقة البحث الشامل Brute force Search.
2. لا يوجد مفتاح في هذه الطريقة، أي أن هذه الطريقة ثابتة ويكفى كسر رسالة واحدة فقط لمعرفة جميع الرسائل الأخرى.

2.1.2 خوارزمية اتباش Atbash

في هذه الطريقة يتم استبدال الحرف A بالحرف Z، والحرف B بالحرف Y، الحرف C بالحرف X.. وهكذا، استخدمت هذه الطريقة في تشفير الرسائل المكتوبة باللغة العبرية، حيث أن الكلمة العبرية atbash تعبر عن أسلوب التبديل المتبع، ولفك التشفير نتبع نفس الأسلوب ولكن بالعكس.

2.1.3 خوارزمية الإزاحة 13 ROT13

في هذه الطريقة يتم إزاحة Rotate كل حرف من الرسالة الأصلية ثلاث عشرة مرة فنحصل على النص المشفر وإذا أردنا إرجاع النص المشفر إلى أصله نقوم بعمل إزاحة لكل حرف ثلاث عشرة مرة أخرى، شكل(4) يوضح عملية الإزاحة المتبعة.



شكل (4): عمل خوارزمية الإزاحة 13.

الخوارزميات السابقة لديها بعض العيوب نلخصها في النقاط التالية:

1. تكمن المشكلة الأساسية في هذه الخوارزميات في وجود التكرار للحروف، حيث أن النص الأصلي والنص المشفر يكونان متشابهان تقريباً، مما يجعل من مهمة محللي التشفير أكثر سهولة.
2. من أهم عيوب هذه الخوارزميات أنها تعتبر ضعيفة حيث يمكن استخدام طرق الإحصاء Statistics المختلفة لفك تشفير الرسالة، ومنها طريقة تحليل التكرار frequency analysis.

لذلك تم اختراع طرق للتشفير بتبديل الأحرف لا يمكن كسرها باستخدام الإحصاء، ومن هذه الطرق:

2.1.4 خوارزمية فيجينير Vigenere

في هذه الطريقة نقوم أولاً باختيار مفتاح التشفير، ثم نتبع الخطوات التالية لتشفير النص:

- نكرر كتابة المفتاح بعدد حروف النص الأصلي، بحيث أن كل حرف من المفتاح يناظر حرف من النص الأصلي.
- نساوي كل حرف من المفتاح والنص بقيمته العددية من 0 إلى 25 حسب ترتيب حروف أبجدية اللغة الإنجليزية، مثلاً نجعل: a = 0، b=1، c= 2 وهكذا...
- نجمع المفتاح مع النص الأصلي للحصول على النص المشفر، كما هو موضح في المعادلة التالية:

النص المشفر = النص الأصلي + الحرف الموازي له من المفتاح.

- لفك التشفير نستخدم المعادلة التالية:

النص الأصلي = النص المشفر – الحرف الموازي له من المفتاح.

مثال:

لنفرض أن المفتاح المستخدم للتشفير هو كلمة danger، والرسالة المراد تشفيرها هي:

Tomorrow is the time

نقوم بإتباع نفس الخطوات السابقة للتشفير ولفك التشفير:

1. نكرر المفتاح حسب عدد حروف النص الأصلي، كالتالي:

key : dangerdangerdange

plaintext : tomorrowisthetime

2. نجمع الحروف المتناظرة معاً للحصول على النص المشفر، بعد تحويل جميع الحروف إلى أرقام، مثلا: t+d تساوي 3+19=22 وهي قيمة الحرف w، إذا w=t+d، كذلك o+a تساوي 0+14=14 وهو حرف o، وهكذا... كما هو موضح في الجدول التالي:

d	a	n	g	e	r	d	a	n	g	e	r	d	a	n	g	e
t	o	m	o	r	r	o	w	i	s	t	h	e	t	i	m	e
w	o	z	u	v	i	r	w	v	y	x	y	h	t	v	s	i

وبعد تشفير الرسالة نحصل على النص المشفر:

Wozuvirwvyxyhtvsi

3. لفك التشفير نطرح كل حرفين متناظرين من بعضهما بعد تحويلهما إلى أرقام، مثلا: d - w = 3 - 22 = 19، والرقم 19 يساوي الحرف t، وهكذا حتى نحصل على النص الأصلي.

2.1.5 خوارزمية الكراسة الواحدة One time pad

ظهرت سنة 1917ف من قبل العالم فيرنام Vernam، في هذه الطريقة مفتاح التشفير يكون طويل جداً، أحيانا أطول من النص الأصلي. من الممكن استخدام كتاب أو قصة أو مجلة لتكون المفتاح، وذلك حسب الاتفاق بين المرسل والمستقبل.

يتم التشفير بأخذ جزء من المفتاح، جملة أو كلمة، بحيث يكون طوله مساوياً لطول النص الأصلي، وتحويل هذا الجزء إلى أرقام، بعد ذلك نزيح كل حرف من النص الأصلي بمقدار الرقم الذي يوازيه من الجزء المأخوذ من المفتاح. ولفك التشفير نستخدم نفس الطريقة ولكن بالمعكوس، أي أنه إذا كانت الإزاحة في التشفير إلى الأمام نرجع الأحرف إلي الخلف عند فك التشفير والعكس صحيح.

من المعروف أن أغلب طرق التشفير يمكن اختراقها، ماعدا هذه الطريقة التي برهن رياضياً على أنها آمنة تماما ويستعصى كسرها، وذلك لأن مفتاح التشفير المستخدم فيها هو مفتاح سري يتم اختياره بطريقة عشوائية تماماً، ويستخدم لمرة واحدة فقط، كذلك فإن حجم المفتاح كبير جدا.

2.2 التشفير بتبديل المواقع Transposition Cipher

في هذه الطريقة يقوم المشفر بتغيير مواقع حروف النص الأصلي وفق قاعدة محددة دون تغيير الحروف نفسها، ومن الممكن أن يتم تقسيم النص الأصلي إلى مجموعات كل منها ذات طول محدد، مثلاً مجموعات من خمسة أحرف، ثم يتم تغيير مواقع الحروف في كل مجموعة بطريقة معينة يمكن تحديدها بالمفتاح السري.

بشكل مبسط التشفير بتبديل المواقع هو التشفير الذي يتم فيه تغيير ترتيب حروف الرسالة.

هناك عدة أنواع من التشفير بتبديل المواقع منها:

2.2.1 خوارزمية ريل فينس Rail Fence

في هذه الخوارزمية نوزع النص الأصلي بشكل عمودي على عدد من الصفوف يتم الاتفاق عليه بين المرسل والمستقبل، وللحصول على النص المشفر ندمج هذه الصفوف معا بشكل أفقي.

مثال:

لتشفير النص Now is the right time نختار أي عدد من الصفوف وليكن صفين ثم نوزع النص عليهما مع إهمال الفراغات الموجودة في النص، وفي حالة كان عدد حروف النص أقل من خانات المصفوفة نقوم بإضافة حرف ليملأ الفراغات الموجودة في المصفوفة، في هذا المثال أضفنا الحرف x، كالتالي:

n	w	s	h	r	g	t	i	e
o	i	t	e	i	h	t	m	x

ثم نقوم بدمج الصفين بشكل أفقي فنحصل على النص المشفر nwshrgtieoiteihtmx

ولفك التشفير نقوم بتوزيع النص المشفر بشكل أفقي على مصفوفة تتكون من صفين ثم ندمج هذين الصفين بشكل عمودي فنحصل على النص الأصلي.

2.2.2 خوارزمية الاتجاه Route Ciphers

خوارزمية ريل فينس ما هي إلا شكل بسيط لخوارزمية الاتجاه، والتي انتشرت في بداية ظهور طرق التشفير. في هذه الخوارزمية يقوم المرسل والمستقبل بالاتفاق على كلمة مفتاحية معينة، ولتشفير الرسالة يقوم المرسل بتقسيمها إلى عناصر Elements كل عنصر عبارة عن حرف وحيد، ثم تقسم هذه العناصر على عدد من الصفوف يساوي عدد حروف الكلمة المفتاحية، بمعنى آخر يتم توزيع هذه العناصر على مصفوفة ثنائية الاتجاه، ومن ثم يتم وضع الكلمة المفتاحية على صفوف المصفوفة لتبين ترتيبها، ولفك التشفير نتبع نفس الطريقة.

مثال:

إذا كانت الجملة المراد تشفيرها هي Transposition ciphers can be broken easily والكلمة المفتاحية هي WAY، نقوم بتوزيع الجملة على مصفوفة بعمق 3 مع إهمال الفراغات، ثم نضع حروف الكلمة المفتاحية WAY في بداية كل صف لتوضيح ترتيب الصفوف، كما هو موضح:

W	t	n	o	t	n	p	r	a	e	o	n	s	y
A	r	s	s	i	c	h	s	n	b	k	e	i	x
Y	a	p	i	o	i	e	c	b	r	e	a	l	x

لاحظ أننا قمنا بوضع حرف X لملء الفراغات في المصفوفة.

وللحصول على النص المشفر نأخذ الصفوف بناءًا على الترتيب الأبجدي للكلمة المفتاحية، وبالتالي سيكون النص المشفر هو:

Rssichsnbkeixtnotnpraeonsyapioiecbrealx

الطرف المستقبل للنص المشفر سيقوم بملء ثلاثة صفوف تبعاً للترتيب الأبجدي للكلمة المفتاحية WAY:

A	t	n	o	t	n	p	r	a	e	o	n	s	y
W	r	s	s	i	c	h	s	n	b	k	e	i	x
Y	a	p	i	o	i	e	c	b	r	e	a	l	x

يمكن زيادة سرية الشفرة عن طريق إعادة تشفير الرسالة الناتجة عن هذه المصفوفة مرة أخرى، مما سيجعل الشفرة عسيرة على الكسر، ولكن هذا سيزيد من الزمن المستغرق في فك شفرة الرسالة عند الطرف المستقبل.

2.2.3 خوارزمية المربع square

في هذه الخوارزمية نضع النص الأصلي على هيئة مربع، وفي حالة كان عدد حروف النص الأصلي أقل من خانات المربع نقوم بزيادة حرف- مثلاً حرف x- لتعبئة هذه الخانات. أبعاد هذا المربع وكيفية استخدامه يتم الاتفاق عليها بين المرسل والمستقبل.

مثال:

إذا أردنا تشفير الجملة التالية:

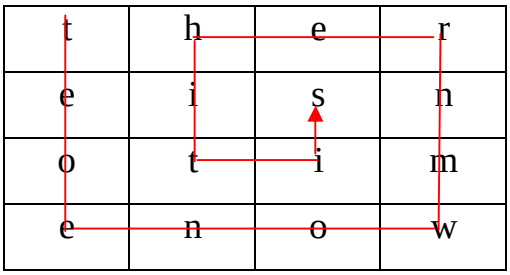
there is no time now

نقوم بتوزيعها مثلاً على 4 صفوف فيتكون لدينا مربع 4×4 كالتالي:

t	h	e	r
e	i	s	n
o	t	i	m
e	n	o	w

وهناك طريقتان لاستخدام المربع وهما:

- الطريقة الحلزونية:

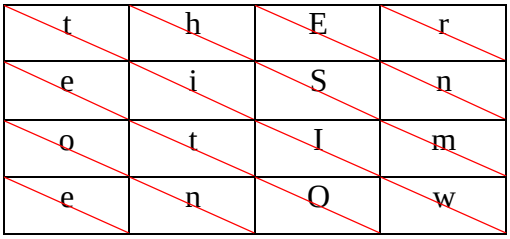


نبدأ من أول خانة ونتحرك بشكل حلزوني فنحصل على النص المشفر:

teoenowmnrehitis

ولفك التشفير نقوم بتوزيع الرسالة على مربع 4×4 بشكل حلزوني فنحصل على الرسالة الأصلية.

- الطريقة القطرية:



نبدأ من أقصى قطر على اليمين فيصبح النص المشفر كالتالي:

renhsmtiiwetoone

والمستقبل يقوم بفك التشفير بنفس الطريقة حيث يوزع الرسالة على مربع 4×4 بشكل قطري.

بعد ظهور أجهزة الحاسوب تم استخدامها في إنجاز عمليات التشفير حيث أن أي طريقة للتشفير تتكون من خوارزمية ومفاتيح للتشفير، قوة وفعالية طريقة التشفير تعتمد على قوة الخوارزمية وطول المفتاح (مقدرًا بالبت)، حيث أنه كلما زاد طول المفتاح زادت قوة التشفير.

طرق التشفير القديمة كانت في اغلب الأحيان تعتمد على سرية خوارزمية التشفير، بينما جميع طرق التشفير الحديثة تعتمد على سرية المفتاح، بدلا من سرية الخوارزمية نفسها.

طرق التشفير التي تعتمد على المفتاح تنقسم إلى:

- التشفير المتناظر.
- التشفير غير المتناظر.

2.3 التشفير المتناظر Symmetric (or secret-key)

شكل (5): التشفير المتناظر.

يسمى أيضاً التشفير بالمفتاح السري، يعتمد فيه كل من المرسل والمستقبل المفتاح السري ذاته، حيث يقوم الأول باستخدام المفتاح لتشفير الرسالة بينما يستخدمه الثاني لفك تشفير الرسالة وقراءتها، الشكل (5) يوضح هذا النوع من التشفير.

أي أنه في هذا النوع من التشفير يكون المفتاح المستخدم للتشفير هو نفس المفتاح المستخدم لفك التشفير، وفي بعض الأحيان يكونان مختلفان ولكن يمكن اشتقاق احدهما بسهولة من الآخر.

تدعى خطوات إعداد المفاتيح ونقلها وتخزينها عملية إدارة المفاتيح والتي يجب أن تتعامل معها كافة تقنيات التشفير، وجميع المفاتيح يجب أن تبقى سرية تماماً. طريقة المفتاح السري تواجه بعض الصعوبات أحياناً فيما يتعلق بإدارة المفاتيح وخصوصاً في الأنظمة المفتوحة التي تتضمن أكثر من مستخدم، ولكن على كل حال تبقى هذه الطريقة أسرع من طريقة التشفير بالمفتاح العام وأقل تعقيداً.

هنالك مشكلتين رئيسيتين تصاحبان التشفير المتناظر وهما:

المشكلة الأولى:

الطرفان اللذان يستخدمان هذا النوع من التشفير يجب أن يتفقان على مفتاح سري معين بدون أن يكشف أي شخص هذا المفتاح، وهذا الأمر يصعب تحقيقه، فكيف ننقل المفتاح من الجهاز (أ) إلى الجهاز (ب) بصورة آمنة إذا كان الاتصال بينهما قابلا للتجسس؟

تتفاقم هذه المشكلة إذا كان الطرفان في منطقتين بعيدتين جغرافياً عن بعضهما البعض، عندها سيتوجب عليهما إيجاد طريقة مضمونة للاتصال فيما بينهما، كوسيط موثوق أو نظام هاتفي آمن أو أي طريقة أخرى تضمن عدم تسرب المفتاح السري، لأنه في حال تسرب المفتاح إلى شخص ما أثناء إرساله فإنه سيغدو بإمكان ذلك الشخص فيما بعد قراءة وتعديل وتزوير جميع الرسائل المشفرة دون أن يشك احد الطرفين في أمره.

المشكلة الثانية:

إذا تصنت شخص ما على الاتصال بين الطرفين فانه يستطيع تخزين الرسائل المرسلة بينهما حتى وان كانت مشفرة، وبينما هو مستمر في مراقبتها وتخزينها، يحاول الحصول على المفتاح بطرق أخرى مثل اختراق الجهاز (أ) أو (ب)، وبمجرد أن يحصل على المفتاح فإن جميع البيانات التي تم تبادلها في السابق والتي سيتم تبادلها في المستقبل ستكون مكشوفة له، وقد يكون لذلك عواقب وخيمة.

والحل للمشكلة الثانية هو أن يتم تغيير هذا المفتاح بصفة دورية، بحيث لا تكون هنالك كمية كبيرة من البيانات الحساسة تعتمد على مفتاح واحد، لكن المشكلة الأولى تزيد المشكلة الثانية تعقيداً، حيث أنه من الصعب تغيير المفتاح بصفة دورية إذا لم تكن هنالك طريقة آمنة وسهلة وعملية لإعلام الطرفين بمفتاح التشفير الجديد.

تنقسم خوارزميات التشفير المتناظر إلى:

2.3.1 التشفير التدفقي Stream Cipher

في هذا النوع يتم تشفير كل بت bit من الرسالة الأصلية على حدة، ومن خوارزميات التشفير التدفقي:

2.3.1.1 خوارزمية RC4

صممت من قبل رون ريفست Ron Rivest لشركة RSA لحماية البيانات سنة 1987م، سميت RC4 اختصاراً لـ Rivest Cipher 4، كانت هذه الخوارزمية سرية حتى سنة 1994 حيث تم نشر شفرتها المصدرية على الانترنت بطريقة غير شرعية من قبل جهة مجهولة، الشفرة التي نشرت مطابقة تقريباً للخوارزمية RC4 وتؤدي نفس تطبيقاتها.

هذه الخوارزمية من أكثر خوارزميات التشفير التدفقي استخداماً، وهي تستخدم في البروتوكولات المعروفة مثل بروتوكول طبقة المداخل الآمنة Secure Sockets Layer (SSL) وذلك لحماية حركة سير الانترنت Internet traffic.

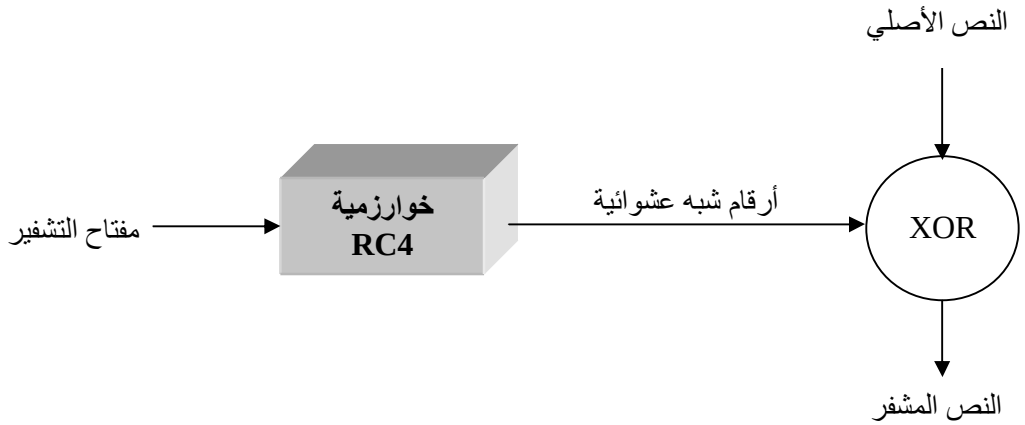
لم تعد هذه الخوارزمية تؤمن الحماية التامة لذلك لم يعد ينصح باستخدامها في الأنظمة الحديثة، ولكنها مازالت تستخدم بكثرة في التطبيقات المختلفة والعوامل الرئيسية التي أدت إلى انتشار استخدامها هي سهولتها حيث يمكن حفظ الخوارزمية بأكملها في دقائق، كذلك سرعتها في التشفير وفك التشفير.

فكرة عمل خوارزمية RC4

الخوارزمية RC4 بحد ذاتها لا تشفر شيئاً ولكنها تولد قيم شبه عشوائية pseudo-random values تسمى المفتاح المتدفق Keystream يستخدم في التشفير وذلك بعمل XOR له مع النص الأصلي، ويستخدم لفك التشفير بعمل XOR له مع النص المشفر، هذا المفتاح المتدفق يتولد بواسطة مفتاح آخر يحدده المستخدم.

عملية XOR هي عملية منطقية يتم فيها مقارنة رقمين ثنائيين، إذا كانا متشابهين تكون نتيجة هذه العمليـــة صفر، أما إذا كانا مختلفين فتكون النتيجة واحد.

عندما تزود الخوارزمية بالمفتاح الذي يحدده المستخدم يتم تكوين جدول يسمى جدول الحالة State Table يحتوي على جميع الحالات المحتملة، وبعد ذلك يتم توليد القيم شبه العشوائية التي تستخدم في التشفير وفـــك التشفير، والشكل(6) يوضح طريقة عمل هذه الخوارزمية.



شكل(6): طريقة عمل الخوارزمية RC4.

شرح خوارزمية RC4

الخوارزمية تنقسم إلى مرحلتين أساسيتين:

1. مرحلة جدولة المفتاح The key-scheduling phase (KSA)

وهي مرحلة تمهيدية يتم فيها ما يلي:

- يختار المستخدم المفتاح الذي يريد استخدامه في التشفير بحيث يكون طوله محصـــوراً بـــين 0 و 255 بايت، ولكن عادة يتم اختيار الطول من 5 إلى 16 بايت والتي تعادل 40 إلى 128 بت.
- يتم تكوين مصفوفة تسمى مصفوفة الحالة The state array، والتي تحتوي على جميع الحـــالات المحتملة أي جميع رموز لوحة المفاتيح, ويكون حجم هذه المصفوفة 256 بايـــت، ويـــتم تعبئتهـــا

بالترتيب من الرقم 0 إلى الرقم 255، أي أن الخانة 0 في المصفوفة تأخذ القيمة 0، والخانة 1 تأخذ القيمة 1... وهكذا.

- يتم إجراء عدة عمليات تبديل على مصفوفة الحالة، وذلك بجمع شفرة الأسكي ASCII Code لمفتاح التشفير مع خانات مصفوفة الحالة وفق القاعدة التالية:

```
For I = 0 to 255
j = (j + state ( I ) + Ascii (userKey)) Mod state.length
state( I ) = Swap(state( I ), state( j ))
```

للحصول على موقع عشوائي في مصفوفة الحالة، بحيث يتم استبدال العنصر الموجود في الموقع الحالي (I) مع العنصر الموجود في الموقع العشوائي (j). كل عنصر في المصفوفة يتم عمل استبدال له على الأقل مرة واحدة، حتى وإن استبدل العنصر مع نفسه. في نهاية هذه العمليات تنتج مصفوفة حالة جديدة تحتوي على جميع الحالات المحتملة ولكنها مرتبة بطريقة عشوائية.

2. مرحلة توليد القيم شبه العشوائية The Pseudo-Random Generation Algorithm (PRGA)

يتم في هذه المرحلة استخدام مصفوفة الحالة الجديدة لتوليد بايتات شبه عشوائية تمثل المفتاح المتدفق، وذلك بتوليد قيمة عشوائية مثلاً x، ومن ثم جمع العنصر الموجود في الموقع x من مصفوفة الحالة مع رقم عشوائي آخر وليكن y والناتج يوضع في y، بعد ذلك نقوم باستبدال العنصر الموجود في الموقع x من المصفوفة بالعنصر الموجود في الموقع y، نكرر هذه العملية حسب طول النص المراد تشفيره، وبعد الانتهاء من هذه العمليات نحصل على مصفوفة حالة جديدة بها قيم شبه عشوائية.

التشفير وفك التشفير

بمجرد الانتهاء من عملية توليد البايتات شبه العشوائية نستطيع البدء في عملية التشفير، والتي يتم فيها عمل XOR للمفتاح المتدفق مع النص الأصلي فينتج النص المشفر.

بعد ذلك عندما يستلم المستقبل النص المشفر يقوم بفك التشفير بنفس الطريقة وذلك بعمل XOR للمفتاح المتدفق مع النص المشفر فينتج النص الأصلي.

مميزات خوارزمية RC4

- صعوبة معرفة مكان وجود القيم في الجدول.
- التشفير بهذه الخوارزمية أسرع عشر مرات تقريباً من التشفير بالخوارزمية DES.
- كل مفتاح يستخدم لمرة واحدة فقط.

عيوب خوارزمية RC4

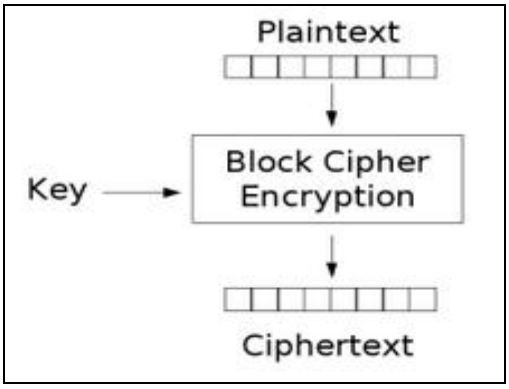
- جدول الحالة المستخدم في الخوارزمية عرضة لهجمات تحليل التشفير.
- مشكلة المفاتيح الضعيفة، فهناك على الأقل مفتاح واحد من 256 مفتاح يمكن أن يكون مفتاح ضعيف.
- تحليل التشفير لهذه الخوارزمية أسرع عشر مرات تقريباً من تحليل تشفير الخوارزمية DES.

2.3.1.2 خوارزمية سيل SEAL

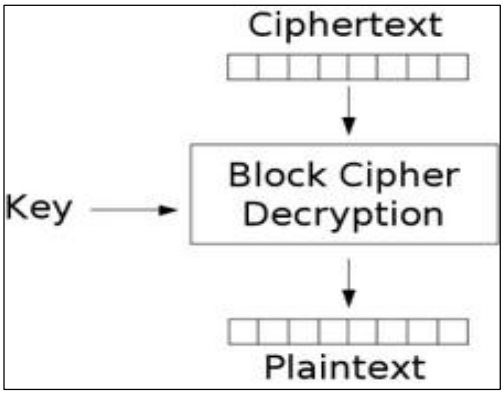
صممت من قبل دون كوبرسميث Don Coppersmith من شركة IBM، وهي تقريبا أسرع خوارزمية تشفير متوفرة حاليا، مفتاح الخوارزمية SEAL يأخذ بضعة كيلو بايتات من مساحة ذاكرة الحاسوب، وتوليده يتطلب خمسة عمليات والعديد من الحسابات التي تتضمن استخدام الخوارزمية SHA1.

2.3.2 التشفير الكتلي أو القوالبي Block Cipher

في هذا النوع يتم أخذ عدد أو كتلة من بتات النص الأصلي (غالبا يتم أخذ 64 بت خاصة في الخوارزميات الحديثة) وتشفيرها كوحدة واحدة single unit أو ككتلة واحدة، فينتج عنها نفس العدد من البتات ولكن مشفرة، ويتم ذلك باستخدام مفتاح التشفير الذي غالبا ما يتم اختياره بشكل عشوائي. الشكلان (7، 8) يوضحان طريقة التشفير وفك التشفير القوالبي.



شكل (7): التشفير القوالبي.



شكل (8): فك التشفير القوالبي.

من خوارزميات التشفير القوالبي:

2.3.2.1 الخوارزمية لوسيفر Lucifer

تعتبر من أوائل خوارزميات التشفير الحديث، صممت في شركة IBM من قبل هورست فيستل Horst Feistel صاحب شبكة فيستل المعروفة، اعتبرت لوسيفر البادرة الأولى التي أدت إلى اختراع DES.

هناك عدة نسخ لهذه الخوارزمية جميعها تحمل نفس الاسم مما أدى إلى حدوث إرباك شديد، وجميع هذه النسخ غير آمنة.

2.3.2.2 الخوارزمية ديس DES

سميت DES اختصاراً للعبارة Data Encryption Standard والتي تعني تشفير البيانات القياسي، ظهرت بسبب احتياج المكتب القومي للمعايير والتقنيات NIST(National Institute of Standards

(and Technology إلى خوارزمية قياسية للتشفير، وفي سنة 1976م تم اختيارها من قبل الحكومة الأمريكية لتصبح الخوارزمية القياسية للتشفير، بعدها أصبحت من أكثر الخوارزميات استخداماً في العالم.

مفتاح التشفير المستخدم في الخوارزمية DES قصير حيث أن طوله 56 بت، وبسبب قصر المفتاح تم اختراق DES في أقل من 24 ساعة، وذلك بواسطة آلة تكلفتها 220.000 دولار أمريكي تم إنشائها خصيصاً لهذا الغرض.

تم تطوير الخوارزمية لتصبح أكثر أمنا وذلك بإحدى طريقتين:

1. استخدام ديس الثلاثية Triple-DES، أي تشفير النص ثلاث مرات.
2. استخدام مفتاح تشفير بطول 128 بت.

ولكن نظراً لتعرض هذه الخوارزمية للاختراق تم استبدالها سنة 2002م بالخوارزمية AES لتصبح الخوارزمية القياسية للتشفير المعتمدة من الحكومة الأمريكية، ومع ذلك مازالت DES تستخدم بكثرة.

2.3.2.3 الخوارزمية RC2

في السابق كانت هذه الخوارزمية سرية إلى أن تم اكتشاف شفرتها من قبل بعض الأشخاص، أدى ذلك إلى شن هجوم على RC2 باختيار حوالي 2^{34} نص أصلي.

2.3.2.4 الخوارزمية ايديا IDEA

طورت في زيورخ/سويسرا سنة 1990م من قبل جيمس ماسي James Massey و اكسوجيا لاي Xuejia Lai كمحاولة لجعلها خوارزمية بديلة للخوارزمية DES، اعتبرت من أفضل الخوارزميات القوالبية وأكثرها أمناً. كذلك فهي سريعة في التشفير وفك التشفير مثل DES، وهذا جعل فرصتها لتحل محل DES كبيرة. المفتاح المستخدم في هذه الخوارزمية حجمه كبير 128 بت.

هناك مشكلة بسيطة في idea وهي مشكلة المفتاح الضعيف weak key problem، هذه المشكلة ليست خطيرة لأنها لا تظهر إلا إذا توفرت الشروط التالية:

- المفتاح المستخدم له تركيبة معينة.
- النص الأصلي يكون موضوع بطريقة معينة من قبل محلل الشفرة لكي يساعده على معرفة المفتاح.
- أن يقوم محلل الشفرة بإعطاء نص للشخص الذي يملك المفتاح لكي يشفره ويرجعه إليه.

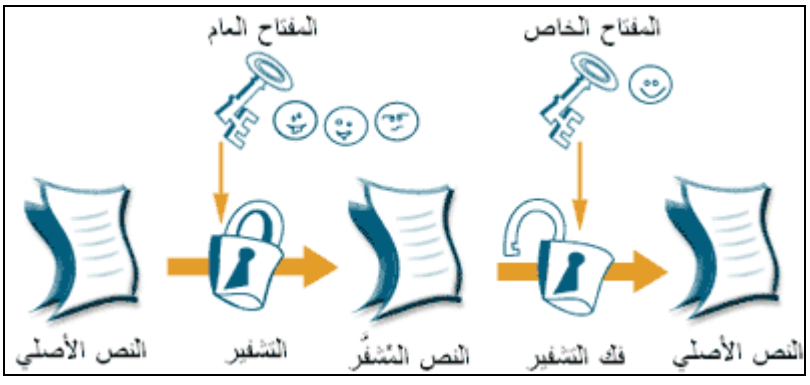
لتجنب مشكلة المفتاح الضعيف يجب ألا يقوم مالك المفتاح بتشفير أي نص لجهة غير معروفة.

2.3.2.5 الخوارزمية تيا TEA

سميت TEA اختصارا للعبارة Tiny Encryption Algorithm والتي تعنى خوارزمية تشفير مصغرة، صممت لتصغير المساحة المستخدمة من الذاكرة وتعظيم السرعة.

تم شن هجوم على هذه الخوارزمية باختيار 2^{23} نص أصلي، هذه المشكلة حدثت كنتيجة لجدولة المفاتيح المستخدمة في الخوارزمية، وهي تحول دون إمكانية استخدام TEA في مجال تأكيد الهوية.

2.4 التشفير غير المتناظر(Asymmetric (or public-key



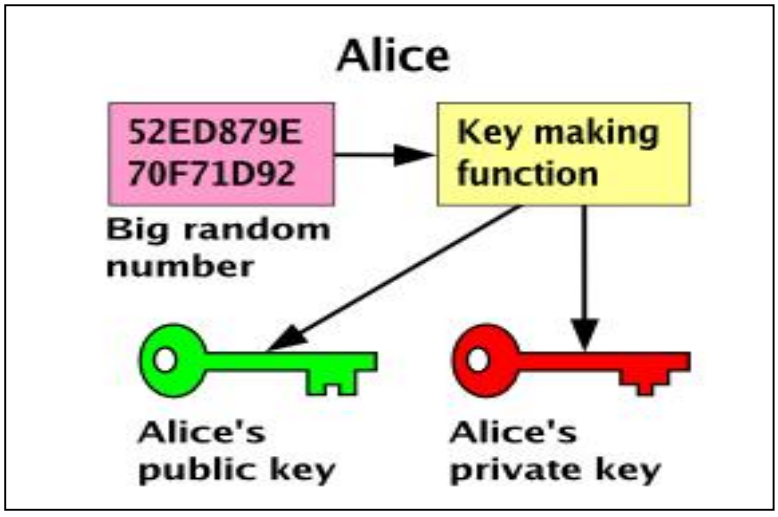
شكل (9): التشفير غير المتناظر.

يسمى أيضا التشفير بالمفتاح العام، الشكل(9) يوضح هذا النوع من التشفير، جاء هذا التشفير حلاً لمشكلة التبادل غير الآمن للمفاتيح في التشفير المتناظر، اكتشف من قبل العالمين ويتفيلد ديفي Whitfield Diffie ومارتن هيلمان Martin Hellman.

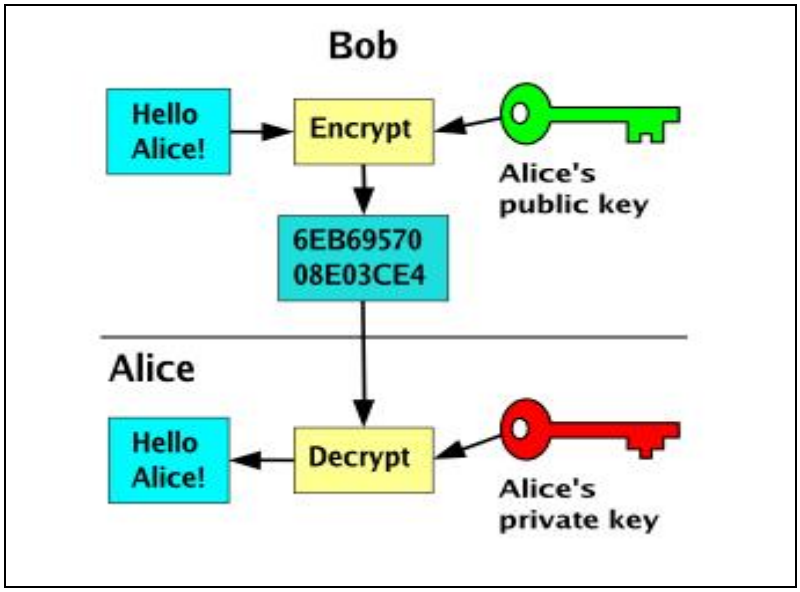
وهنا بدلاً من استخدام مفتاح واحد يتم استخدام مفتاحين، المفتاح الخاص private key يكون معروف لدى الجهة أو الشخص الذي يملكه فقط، بينما المفتاح العام public key يتم نشره، مثلاً على موقع في شبكة الانترنت، بحيث يستطيع أي شخص الحصول عليه، المعلومات المشفرة باستخدام المفتاح العام يمكن فك تشفيرها باستخدام المفتاح الخاص المرتبط به فقط، والعكس صحيح أي أن المعلومات المشفرة بالمفتاح الخاص يتم فكها باستخدام المفتاح العام المرتبط به، ولكن هذا المفتاح متوفر للجميع فيمكن لأي شخص معرفة هذه المعلومات، إذاً ما الفائدة من هذا التشفير؟

في الحقيقة أن أهمية هذا النوع من التشفير تكمن في إمكانية قيام المرسل بإرسال توقيعه الرقمي مشفراً بواسطة المفتاح الخاص ويقوم المستقبل بفك الشفرة باستخدام المفتاح العام فإذا وجد أن التوقيع صحيح فهذا يعني أن المرسل هو الطرف الصحيح المتوقع، وإذا وجد أن المفتاح خاطئ فهذا يعني أن هناك من ينتحل شخصية المرسل الحقيقي، سرية هذا التشفير تعتمد على سرية المفتاح الخاص.

يتم توليد المفتاحين الخاص والعام بطريقة معينة حتى يكونان مرتبطين ببعضهما البعض، الشكلان(10، 11) يوضحان عملية توليد المفتاحين واستخدامهما في التشفير.



شكل (10): الرقم العشوائي يستخدم لإنتاج زوج المفاتيح.



شكل (11): التشفير بالمفتاح العام.

يتطلب التشفير غير المتناظر عمليات أكثر من العمليات التي يتطلبها التشفير بالمفتاح المتناظر لذلك لا تعتبر طريقة المفتاح العام مناسبة عند وجود كمية كبيرة من المعلومات ولكن من الممكن استخدامه لتشفير المفتاح السري لخوارزميات التشفير المتناظر وبالتالي نستطيع إرساله عبر الشبكة بأمان.

من أهم خوارزميات التشفير غير المتناظر الخوارزمية RSA التي سنتناول شرحها بالتفصيل.

الخوارزمية RSA

خوارزمية RSA من أشهر الخوارزميات المستخدمة في التشفير غير المتناظر، وتحتوي هذه الطريقة على مفتاحين عام وخاص, المفتاح العام يكون معروفاً من قبل أي شخص ويستعمل لتشفير الرسائل, بينما يكون الخاص سرياً ويستخدم لفك تشفير الرسائل.

اخترعت هذه الخوارزمية عام 1977م من قبل العلماء الثلاثة رون ريفست Ron Rivest، وادي شامير Adi Shamir، وليونارد ادلمان Leonard Adleman، في معهد ماسشيوتس للتكنولوجيا في الولايات المتحدة الأمريكية وسميت RSA اختصاراً لأسمائهم، التقنية المستخدمة في خوارزمية RSA اكتشفها العالم البريطاني كليفورد كوك Clifford Cocks سنة 1973م ثم قام هؤلاء العلماء الثلاثة بتطويرها حتى تصبح خوارزمية تشفير. لم تنشر هذه الخوارزمية حتى عام 1997 بسبب سريتها.

تختلف هذه الخوارزمية عن بقية الخوارزميات في أنها تعتمد على أساس رياضي جديد وهو الرفع إلى القوى (الأسس)، كذلك تم استخدام النظرية الرياضية Euler's Theorem، بالإضافة إلى أنها تحتوي على مفتاحين يتم توليدهما كالتالي:

طريقة توليد المفتاحين العام والخاص

- نختار رقمين أوليين عشوائيين مثلاً س، ص.
- نحسب حاصل ضربهما ن= س×ص، حيث تكون قيمة ن عامة وليست سرية، وهي تمثل جزء من المفتاحين العام والخاص. يجب أن يكون ن أكبر من عدد الحروف الهجائية للغة المستخدمة (أي أكبر من 26 إذا كنت تستخدم اللغة الإنجليزية).
- نحسب فاي Phi للقيمة ن، والذي يرمز له بالرمز Φ، وهو يمثل العدد الكلي للأعداد الأصغر من ن والأولية بالنسبة له، يكون العدد أولي بالنسبة للعدد ن عندما لا يوجد بينهما عامل مشترك اكبر غير الواحد الصحيح، ويتم حساب Φ من خلال المعادلة الرياضية التالية:

$$\Phi(ن) = (س-1) \times (ص-1)$$

هذه المعادلة لا يمكن استخدامها إلا إذا كان س و ص أوليان.

- نختار أي رقم صحيح وليكن ع بحيث Φ(ن) > ع >1، ويكون أولي بالنسبة لـــ Φ(ن) أي أن العامل المشترك الأكبر بين ع، Φ(ن) يساوي الواحد الصحيح. ع يمثل جزء من مفتاح التشفير.
- نحسب قيمة الرقم الصحيح خ الذي يمثل جزء من مفتاح فك التشفير، بحيث يكون Φ(ن) > خ >1، ويحقق المعادلة:

$$(ع \times خ) \bmod \Phi(ن) = 1$$

أي أن:

$$ع \times خ = ك \times \Phi(ن) + 1$$

حيث ك هو أول عدد صحيح يجعل خ عدد صحيح أولي بالنسبة لـــ Φ(ن) وبالنسبة للعدد ع.

المفتاح العام سيكون (ن، ع)، والفتاح الخاص سيكون (ن، خ)، ويجب أن يكون خ سريا، بينمـــا ع و ن يكونان عامان.

يزداد أمان التشفير عند اختيار رقمين أوليين كبيرين، مع مراعاة ألا يكون الفارق بينهما صغيراً جـــداً أو كبيراً جداً، بعد توليد المفاتيح نستطيع نسيان هذين الرقمين لأننا لن نحتاج لهما بعد الآن.

هناك سببان رئيسيان لاختيار الأرقام الأولية وهما:

1. نستطيع حساب Φ(ن) بسهولة.
2. يصبح ن صعب التحليل، فكلما قل عدد عوامله يزداد الوقت اللازم لإيجاده.

طريقة التشفير بهذه الخوارزمية

نقوم بترميز كل حرف من الحروف الهجائية بإعطائه رقم معين، مع مراعاة ما يلي أثناء عملية الترميز:

- عدم استخدم الواحد الصحيح لأن أي رقم يرفع للأس واحد يبقي كما هو.
- عدم تجاوز العدد (ن-1).
- عدم استخدام مضاعفات العددين س و ص.

يتم التشفير برفع ترميز كل حرف من حروف النص الأصلي إلى الأس ع، ثم نقسم الناتج على ن، ونأخـــذ باقي القسمة لنحصل في النهاية على النص المشفر، والمعادلة التالية توضح هذه العملية، حيث أن ت هـــي ترميز النص الأصلي:

$$\text{النص المشفر} = \text{ت}^{\text{ع}} \bmod \text{ن}$$

طريقة فك التشفير بهذه الخوارزمية

عندما يستقبل الطرف الآخر النص المشفر يقوم برفع كل حرف من حروفه إلى الأس خ، ثم يأخذ باقي قسمة الناتج على ن ويحوله إلى الحرف الذي يقابله، والمعادلة التالية توضح ذلك، حيث أن ش هو ترميز النص المشفر:

$$\text{ش}^{\text{خ}} \bmod \text{ن} = \text{النص الأصلي}$$

مثال على هذه الطريقة:

1. نختار أي رقمين أوليين وليكونا 5 و 11.
2. ن= 5 ×11 = 55.
3. Φ (55) = (5-1) × (11-1) = 4 × 10 = 40، حيث أن الأعداد الأصغر من 55 والأولية بالنسبة له عددها 40 وهي 1، 2، 3، 4، 6، 7، 8، 9، 12، 13،...،54.

يكون العدد أولي بالنسبة للعدد 55 عندما لا يوجد بينهما عامل مشترك اكبر غير الواحد الصحيح.

4. نختار أي رقم أولي بالنسبة للعدد 40 ليكون المفتاح العام ع، مثلا نختار الرقم 7، ومن ثم نحسب قيمة خ من المعادلة:

$$7 \times \text{خ} = \text{ك} \times 40 + 1$$

$$\text{خ} = (\text{ك} \times 40 + 1) \div 7$$

نبدأ البحث من الرقم واحد عن قيمة ك التي تجعل خ عدد صحيح أولي بالنسبة للعددين 7 و40، فنجد أن ك =4، وتكون قيمة خ:

$$\text{خ} = (40 \times 4) + 1 \div 7 = 23.$$

الآن تحصلنا على العددين ع و خ، فيصبح المفتاح العام (55، 7)، والمفتاح الخاص (55، 23).

5. نقوم بترميز الحروف الهجائية على ألا نزيد عن الرقم 54 كالتالي:

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
2	3	4	6	7	8	9	12	13	14	16	17	18	19	21	23	24	26	27	28	29	31	32	34	36	37

6. إذا أردنا تشفير كلمة ما، مثلا الكلمة hi، فإننا نقوم بالعمليات الحسابية التالية:

$h = 12^{7} \pmod{55} = 35831808 \pmod{55} = 23 = p$

$i = 13^{7} \pmod{55} = 62748517 \pmod{55} = 7 = e$

يصبح النص المشفر pe.

7. لفك تشفير الرسالة pe نقوم بالحسابات التالية:

$P = 23^{23} \pmod{55} = 12 = h$

$e = 7^{23} \pmod{55} = 13 = i$

وبالتالي نحصل على الرسالة الأصلية hi.

كيف نجعل من تشفيرنا تشفيراً صعب الكسر؟

في الحقيقة تكمن الصعوبة في كسر تشفير الخوارزمية RSA في الرقم ن (والذي يمثل حاصل ضرب الرقمين الأوليين), فكلما زادت قيمة ن زادت صعوبة الحصول على المفتاح الخاص, كذلك فإننا سنحصل على Φ(ن) أكبر وبالتالي سنستطيع أن نغطي حروف وأرقام ورموز أكثر في جدول التشفير ومن الممكن أن نشفر كل حرف أكثر من مرة، مثلا نشفر الحروف الكبيرة والحروف الصغيرة كلاً على حدة، وهذا سيزيد صعوبة كسر الشفرة. ولكن المشكلة هنا هي أننا لن نصبح قادرين على إيجاد المفتاحين بطريقة يدوية أي باستخدام القلم والورقة, لذا سنستخدم الحاسب لهذه العملية، وهنا سوف تظهر لنا مشكلة أخرى وهي أننا لن نستطيع استخدام متغير نضع فيه القيمة الكبيرة الناتجة من رفع الحرف لقوة كبيرة جداً, مما سيعطل عمل التشفير بالأرقام الكبيرة.

الحل لهذه المشكلة يكون باستخدام بواقي القسمة لأن رياضيات بواقي القسمة تصغر الأرقام الكبيرة، وطريقة استخدام باقي القسمة لإيجاد الأس مشروحة بالأرقام في المثال التالي.

مثال:

إذا أردنا إيجاد ناتج العملية $4^{13} \bmod 497$ فإننا نستخدم المعادلة س=(4× أ) $\bmod 497$ ثلاثة عشر مرة، حيث أن أ يبدأ من الرقم 1 ثم يأخذ قيمة س، كما يلي:

- س= (4×1) $\bmod 497 = 4 \bmod 497 =$ 4.
- س= (4×4) $\bmod 497 = 16 \bmod 497 =$ 16.
- س= (4×16) $\bmod 497 = 64 \bmod 497 =$ 64.
- س= (4×64) $\bmod 497 = 256 \bmod 497 =$ 256.
- س= (4×256) $\bmod 497 = 1024 \bmod 497 =$ 30.
- س= (4×30) $\bmod 497 = 120 \bmod 497 =$ 120.
- س= (4×120) $\bmod 497 = 480 \bmod 497 =$ 480.
- س= (4×480) $\bmod 497 = 1920 \bmod 497 =$ 429.
- س= (4×429) $\bmod 497 = 1716 \bmod 497 =$ 225.
- س= (4×225) $\bmod 497 = 900 \bmod 497 =$ 403.
- س= (4×403) $\bmod 497 = 1612 \bmod 497 =$ 121.
- س= (4×121) $\bmod 497 = 484 \bmod 497 =$ 484.
- س= (4×484) $\bmod 497 = 1936 \bmod 497 =$ 445.

وإذا قمنا بحساب $4^{13} \bmod 497$ بطريقة يدوية سنجد أن الناتج يساوي 445.

بالرغم من كل ذلك فإن نظام RSA ليس عصياً على الاختراق، إذ إن اختراقه أمر ممكن إذا توفر ما يلـزم لذلك من وقت ومال، ولذلك تم تطوير نظام PGP(Pretty Good Privacy) الذي يعـد نموذجـاً محسـناً ومطوراً من نظام RSA، ويستخدم PGP مفتاحا بطول 128 بت، إضافة إلى استخدامه البصمة الإلكترونية للرسالة message digest، ولا يزال هذا النظام منيعاً على الاختراق حتى يومنا هذا.

استخدامات الخوارزمية RSA

- تستخدم لتشفير المفتاح السري للخوارزمية DES قبل إرساله عبر الشبكة، في حين تسـتخدم DES لتشفير وفك تشفير النصوص. فبالرغم من أن خوارزمية RSA أفضل وأكثر أمناً من خوارزمية DES إلا أنها أبطأ بكثير، لذلك لا تستخدم بكثرة في التشفير.
- تستخدم في مجال تأكيد الهوية حيث تقوم هذه الخوارزمية بإنشاء تواقيع رقمية، وهنا يـتم اسـتخدام المفتاح الخاص في التشفير والمفتاح العام يستخدم في فك التشفير.

2.5 التشفير باتجاه واحد One way hash

التشفير باتجاه واحد One way encryption هو أسلوب من أساليب التشفير يتم فيه أخذ الرسـالة المـراد تشفيرها وتحويرها لإنتاج مفتاح التشفير Hash Key. سمي هذا الأسـلوب بالتشـفير باتجـاه واحـد لأن النصوص المشفرة بهذه الطريقة لا يمكن فك تشفيره أبداً.

هذا الأسلوب هو في الواقع أكثر الأساليب استخداماً، ومن أشهر خوارزمياتـه المسـتخدمة حاليـا md5، ويعتبر البعض خوارزمية sha1 خليفة للخوارزمية md5، وقد بدأت بعض المشـاريع بالانتقـال تـدريجيا لاستخدامها بدلا من md5.

لماذا نقوم بتشفير البيانات إذا لم نكن قادرين بعد ذلك على فك تشفيرها؟

ذلك لأن تشفير نفس الرسالة بنفس الخوارزمية ينتج مفتاح الشفرة نفسه في كل مرة، لـذلك يسـتخدم هـذا الأسلوب في بعض الأنظمة للتحقق من صحة معلومات ما دون الحاجة لمعرفة محتوى هذه المعلومات.

استخدامات التشفير باتجاه واحد

- **تخزين كلمات المرور في ملف قابل للاختراق**

يستخدم التشفير باتجاه واحد من قبل أصحاب المواقع في تشفير كلمات المرور للمشتركين، وتخزين مفتاح الشفرة في ملف، وبذلك من المستحيل أن يتمكن المخترق من معرفة كلمات المرور حتى ولو تحصل على الملف الذي يحتوي على مفاتيح الشفرة لكلمات المرور.

وفي المقابل عندما يقوم المستخدم الذي يعرف كلمة المرور الصحيحة الخاصة به بمحاولة الدخول على الموقع، يقوم برنامج معين بتشفير كلمة المرور المدخلة ومقارنة مفتاح الشفرة الناتج بمفتاح الشفرة المخزن في ملف المستخدم، فإذا تطابق المفتاحان نعلم بأن كلمة المرور صحيحة.

- **إرسال كلمة المرور بصورة مشفرة على الشبكة**

إذا كان الملف الذي يحتوى على كلمات المرور المشفرة مخزن على الخادم server، في حالة حدوث اختراق فإن المخترق لن يتمكن من معرفة كلمة المرور الحقيقية، لكن هنالك مشكلة أخرى وهي أن البيانات في أغلب المواقع ترسل على الشبكة عبر اتصال غير آمن، وذلك يعني بأنه من السهل معرفة كلمة المرور التي أرسلت عبر الشبكة عن طريق التصنت على الاتصال.

الحل لهذه المشكلة أن نقوم بتشفير كلمة المرور قبل إرسالها، ومقارنتها بالنسخة المشفرة عند وصول كلمة المرور إلى هناك، وهنا نكون قد تأكدنا من أن المتجسسين على الاتصال لن يعرفوا كلمة المرور.

- **التأكد من عدم التلاعب ببيانات ما**

من استخدامات نظام التشفير باتجاه واحد هو التحقق من عدم التلاعب ببيانات أو ملفات ما، سواء كان هذا التلاعب متعمد، أو أنها أصيب بفيروس، أو حدث بها تغيير غير متعمد من شخص ما، فنحن بحاجة هنا لطريقة نتأكد بها من تطابق نسخة الملف التي لدينا مع النسخة الأصلية.

لهذا السبب تقوم الكثير من الشركات المصممة للبرامج بوضع مفتاح تشفير تنتجه بنظـــام md5 أوsha1 مع البرنامج، ليتمكن المستخدم من تشفير نسخة الملف التي لديه ومقارنة مفتاح التشفير النـــاتج بمفتـــاح التشفير المرفق مع البرنامج، وبالتالي التأكد من أن البرنامج سليم ولم تحدث عليه أية تغييرات.

كذلك فإن هذه الطريقة تستخدم كأسلوب أمني لأجهزة الحاسوب للتأكد من عدم حـــدوث تغييـــرات فـــي الأجزاء الحساسة من النظام.

الفصل الثالث

تحليل التشفير

3.1 تحليل أو كسر التشفير

كلمة Cryptanalysis مشتقة من الكلمة اليونانية kryptós analýein والتي تعني تفكيك أو تحليل الكتابة المخفية. تحليل التشفير هو دراسة طرق استنتاج معنى المعلومات المشفرة، بدون معرفة المعلومات السرية المطلوبة لإنجاز ذلك مثل معرفة المفتاح المستخدم في التشفير.

تحليل التشفير قد يستغل من قبل مخترق ما لتخريب نظام التشفير، أو من قبل مصمم نظام التشفير لتطوير قابلية النظام ضد الاختراق، حيث إن خوارزميات التشفير يجب أن تفحص وتختبر جيدا قبل استخدامها حتى يكون نظام التشفير موثوقاً به، فبدون هذه الاختبارات لن يكون هناك ثقة في جودة نظام التشفير.

الهدف من علم تحليل التشفير هو العثور على بعض نقاط الضعف (الأماكن غير المحمية) في الخوارزمية المستخدمة للتشفير.

معظم طرق التشفير يمكن اختراقها بعدد من التخمينات باستخدام طريقة البحث الشامل أو أي طريقة أخرى، ولكن عدد التخمينات أو المحاولات المطلوبة تعتمد على حجم المفتاح، فكلما كان المفتاح أطول كلما كانت محاولات تحليل التشفير أصعب.

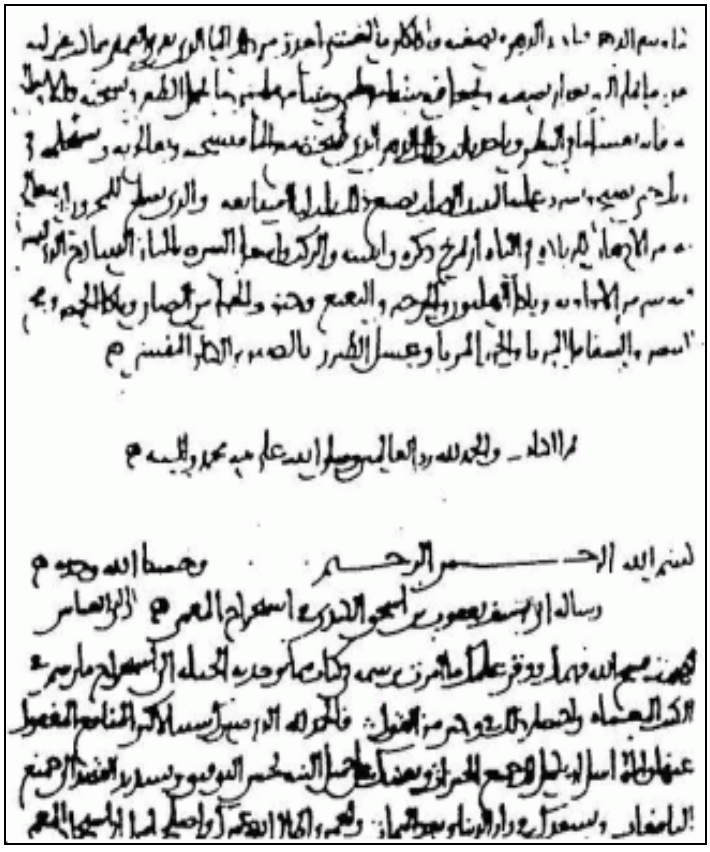
3.2 تاريخ تحليل التشفير History of cryptanalysis

تحليل أو كسر التشفير ارتبط منذ البداية بظهور التشفير، فمعظم طرق التشفير القديمة تم إيجاد طرق لكسرها وهذا أدى إلى تصميم وتطوير طرق جديدة للتشفير لتحل محل الطرق المكسورة والتي لم تعد آمنة الاستخدام، وبالتالي أدى هذا إلى اختراع تقنيات جديدة لتحليل التشفير لكسر هذه الطرق المطورة.

بمعنى آخر التشفير وتحليل التشفير متلازمان دائما، وعليه لكي نستطيع بناء نظام تشفير آمن يجب أن يصمم بطريقة تجعله مضاد لجميع محاولات تحليل التشفير المحتملة.

3.3 الطرق القديمة لتحليل التشفير Classical cryptanalysis

على الرغم من أن كلمة تحليل التشفير تعتبر حديثة حيث ابتكرها العالم ويليام فريدمان William Friedman سنة 1920م، ولكن الحقيقة إن طرق تحليل الشفرات قديمة جداً، فأول شرح مسجل لطرق كسر الشفرات كتب في القرن التاسع من قبل العالم العربي أبو يوسف الكندي، في مخطوطة توضح طريقة فك شفرة الرسائل. هذه المخطوطة تشتمل أيضاً على وصف لطريقة تحليل التكرار Frequency Analysis، والذي سنتناول شرحه بالتفصيل، مخطوطة الكندي موضحة في الشكل(12).



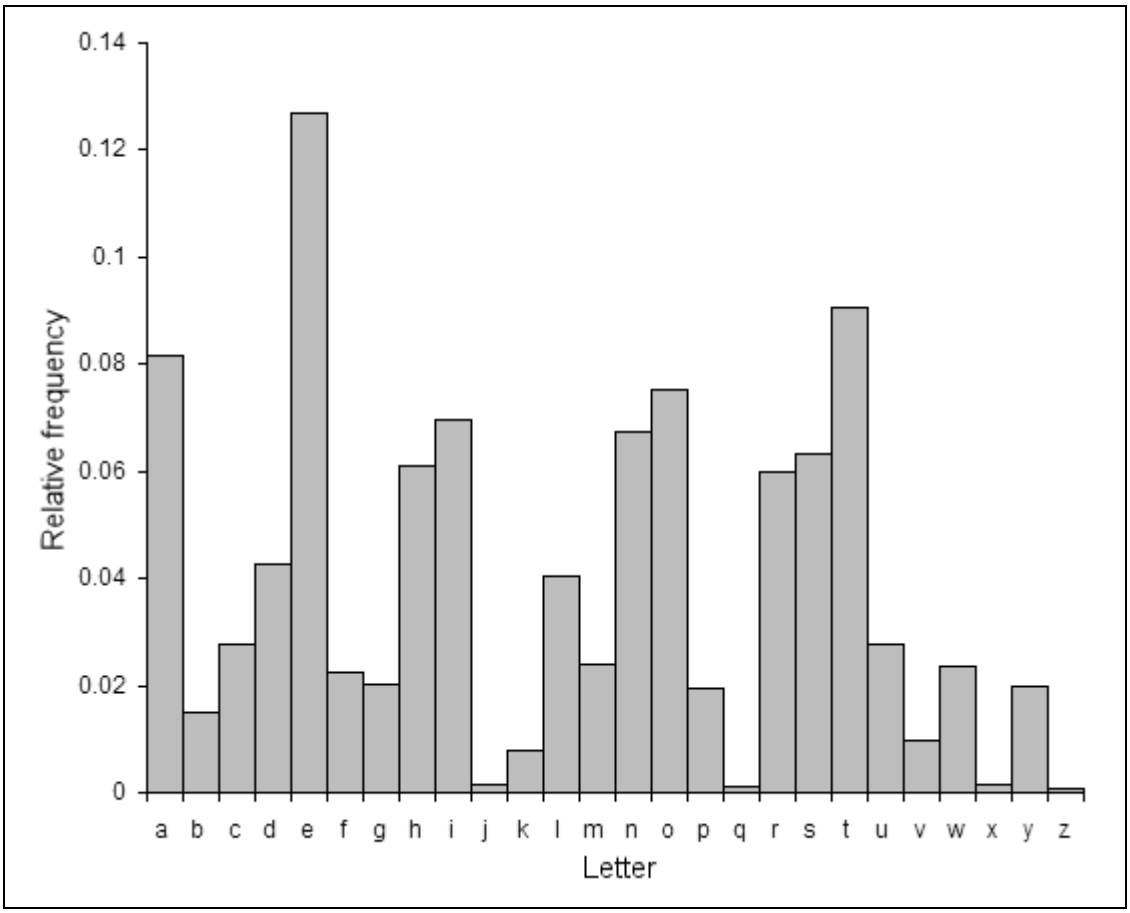
شكل (12):

الصفحة الأولى من مخطوطة الكندي التي توضح كيفية تحليل النصوص المشفرة

تحليل التكرار

تحليل التكرار هو أحد الطرق التقليدية لكسر النصوص المشفرة والذي يعتمد على الإحصاء بشكل أساسي.

في اللغات الطبيعية هناك حروف هجائية معينة تظهر بشكل متكرر، فمثلاً أكثر الحروف تكراراً في اللغة الإنجليزية هو الحرف E حيث أنه يظهر بكثرة في جميع النصوص (يتم استخدامه بمعدل كل 8 أحرف)، ثم يأتي من بعده الحروف: T، I، A، O، N، S، R، H على الترتيب، بينما الحروف نادرة الاستخدام هي: Z، J، K، Q، X على الترتيب، الشكل(13) والجدول(2) يوضحان نسب مئوية تقريبية لتكرار الحروف في اللغة الإنجليزية Letters frequencies.



شكل(13): رسم بياني يوضح نسب تكرار الأحرف في اللغة الإنجليزية.

الحرف	نسبة التكرار
E	0.127
T	0.097
I	0.075
A	0.073
O	0.068
N	0.067
S	0.067
R	0.064
H	0.049
C	0.045
L	0.040
D	0.031
P	0.030
Y	0.027
U	0.024
M	0.024
F	0.021
B	0.017
G	0.016
W	0.013
V	0.008
K	0.008
X	0.005
Q	0.002
Z	0.001
J	0.001

جدول(2): النسب المئوية لتكرار الأحرف الأحادية.

كذلك فإن أكثر حرفان متتاليان يتكرران هما الحرفان TH، ويأتي من بعدهما الحروف: IN، AN, ER، RE، الجدول(3) يوضح النسب المئوية لاستخدام مجموعات مكونة من حرفين في اللغة الإنكليزية.

الأحرف	التكرار
TH	%3
IN	%1.5
AN	%1

جدول(3): النسب المئوية لتكرار بعض الأحرف الثنائية.

كذلك أكثر ثلاث حروف متتالية تتكرر هي: THE، ING، AND، ION، FOR.

الجدول(4) يوضح النسب المئوية لاستخدام مجموعات مكونة من 3 أحرف في اللغة الإنكليزية.

الأحرف	التكرار
The	%6.5
ing	%1.5
And	%0.5
For	%0.67

جدول(4): النسب المئوية لتكرار بعض الأحرف الثلاثية.

تحليل التكرار يعتمد على ضعف التشفير في إخفاء هذه الإحصائيات، فمثلا طرق التشفير بتبديل الحروف والتي يتم فيها تبديل حرف بحرف آخر يتم كسر شفرتها عن طريق استخدام الإحصائيات السابقة. تحليل التكرار يعتمد أيضا على معرفة اللغة المستخدمة في التشفير.

أمثلة على استخدام تحليل التكرار في كسر الشفرات

مثال1:

نفرض أن لدينا النص المشفر التالي:

Ab rtx cxor abrxsxoro mp rtx qxmpux

قبل أن تفكر بعملية فك الشفرة يجب أن تنمي عندك موهبة تحليل نظم التشفير، وهذا يحتاج لكثير من القراءة والفهم لأنظمة التشفير والطرق المستخدمة للتشفير وعمل تطبيقات عليها.

العبارة السابقة مشفرة بنوع من أبسط أنواع التشفير والذي يحتاج فقط إلى وقت قليل لكسره، وهذا النوع هو التشفير بتبديل الأحرف.

الخطوات اللازمة لتحليل التشفير باستخدام طريقة تحليل التكرار

1. إذا تأملت قليلا في العبارة السابقة ستجد فيها كلمتين متشابهتين وهي rtx وبفرض إطلاعك على أنظمة التشفير وعلوم اللغات ستعرف أن أكثر الكلمات شيوعا في اللغة الإنكليزية هي كلمة the كما أن الحرف E هو أكثر الحروف استعمالاً في الإنكليزية، كما هو موضح في الجدول (2).
2. ستلاحظ أيضا أن أكثر حرف تكرر في العبارة المشفرة السابقة هو الحرف X، وهنا يمكن افتراض أن حرف X يرمز إلى حرف E، وهذا يدعم بقوة الافتراض السابق وهو احتمال كون rtx هي كلمة the، هنا نقوم باستبدال rtx في العبارة المشفرة بكلمة the، وكذلك كل حرف X بحرف E، وبذلك نحصل على الجملة التالية:

Ab the ceor abreseoro mp the qemque

3. أيضا على افتراض أن rtx هي كلمة the نقوم باستبدال كل حرف r في العبارة السابقة بحرف t، والنتيجة تكون كالتالي:

Ab the ceot abteseoro mp the qemque

4. إلى هنا نكون قد قطعنا مرحلة جيدة في فهم الشفرة والإمساك ببعض الخيوط الأساسية لفكها، ولنكمل البقية نقوم بالإطلاع على الجدول(2) وتطبيق النسب المذكورة فيه على العبارة المشفرة بنفس الطريقة التي بدأنا بها، وبعد عدة محاولات سنستطيع كسر الشفرة والحصول على النص الأصلي وهو:

In the best interests of the people

مثال 2:

إذا كان النص المشفر هو:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ.

لمعرفة النص الأصلي نتبع الخطوات التالية:

1. نحسب الحروف المتكررة في النص بأكبر تكرار.
2. بالنظر إلى الجدول(2) نستطيع التخمين أن P=e و Z=t.
3. نحسب التكرار لكل حرفين متتاليين يظهران في النص أكثر من مرة.
4. بالرجوع إلى الجدول(3) يمكننا التخمين أن ZW=th وبالتالي يكون ZWP=the.

وبعد عدة محاولات نحصل على النص التالي :

it was disclosed yesterday that several informal but direct contacts have been made with political representatives of the viet cong in Moscow.

إذا أصبح النص المشفر أكثر تعقيداً فإن الرياضيات تصبح هي المسيطرة على عمليات تحليل التشفير، حيث أن الإحصائيات لن تكون مفيدة في هذه الحالة.

هذا التغيير برز خلال الحرب العالمية الثانية حيث أن محاولات كسر النصوص المشفرة تطلبت استخدام مستوى جديد من الطرق الرياضية. علاوة على ذلك تم تطبيق الميكنة لأول مرة لتحليل التشفير باستخدام جهاز يدعى بومبا Bomba والموضح في الشكل (14)، بالإضافة إلى حواسيب أولية.

شكل (14): جهاز بومبا.

3.4 الطرق الحديثة لتحليل التشفير Modern cryptanalysis

في الحرب العالمية الثانية الحسابات أو التخمينات computation كانت تستخدم بشكل كبير في تحليل التشفير ونتائجها كانت جيدة، كذلك فإنها ساعدت على ظهور خوارزميات تشفير أكثر تعقيداً من السابق مثل الخوارزمية AES، هذه الخوارزميات الحديثة تعتبر محصنة أكثر ضد الكسر أو الخرق، ولكن هذا لا يعنى أنه تم التغلب على تحليل التشفير نهائياً حيث من الممكن جداً ظهور طرق جديدة لتحليل التشفير، خاصة إن الشفرات الضعيفة لم تنقرض بعد.

هنالك عدة طرق أو هجمات حديثة لكسر التشفير نذكر منها:

3.4.1 هجوم معرفة النص الأصلي Known-plaintext attack

في هذا النوع من الهجمات يستطيع المهاجم معرفة النص الأصلي أو التكهن به من خلال أجزاء معينة من النص المشفر، والمهمة هنا تكون فك تشفير بقية النص المشفر باستخدام هذه المعلومات، ومن الممكن إنجاز ذلك عن طريق تحديد المفتاح المستخدم في التشفير.

من أفضل طرق هذا النوع من الهجمات هي طريقة تحليل التشفير الخطي التي تستخدم ضد التشفير القوالبي.

3.4.2 هجوم اختيار النص الأصلي Chosen-plaintext attack

المهاجم هنا يكون قادراً على الحصول على أي نص مشفر يريده ولكنه لا يعرف المفتاح، فتصبح مهمته هي تحديد المفتاح المستخدم في التشفير.

من طرق هذا الهجوم طريقة تحليل التشفير بالتفاضل والتي يمكن تطبيقها ضد التشفير القوالبي.

بعض أنظمة التشفير، وخاصة نظام RSA، غير محصنة ضد هجوم اختيار النص الأصلي، ولذلك عند استخدام هذه الخوارزميات يجب اتخاذ الحيطة والحذر أثناء تصميم التطبيقات التي تستخدم هذه الخوارزميات، حتى لا ينجح المهاجم في تحليل النص باستخدام هذه الطريقة.

3.4.3 هجوم الرجل الوسطي Man-in-the-middle attack

هذا الهجوم يستخدم ضد طرق تشفير الاتصالات وطرق تبادل مفاتيح التشفير. فعندما يقوم طرفان مثلا أ و ب، بتبادل مفتاح التشفير للحفاظ على سرية الاتصالات بينهما، فإن المهاجم يتمركز على خط الاتصالات بين هذين الطرفين، ثم يقوم باستقبال الإشارات التي يبعثها الطرفان إلى بعضهما، وبعد ذلك يقوم بتبادل مفتاح(من اختياره) مع الطرف أ ثم مع الطرف ب كل على حدة، وبالتالي فإن الطرفين أ و ب سوف يستخدمان مفتاحين مختلفين للتشفير، هذين المفتاحين معروفان لدى المهاجم. بهذه الطريقة يستطيع المهاجم فك تشفير أي اتصال مرسل من أ باستخدام المفتاح الذي تبادله معه ومعرفة محتوى هذا الاتصال، ثم يقوم بتشفيره مرة أخرى بالمفتاح الذي تبادله مع ب ويرسله إلى الطرف ب، الطرفان أ و ب سيعتقدان أن الاتصال بينهما آمن، ولكن في الحقيقة فإن المهاجم يشاركهما في جميع اتصالاتهما.

أفضل طريقة لتفادي هجوم الرجل الوسطي هي استخدام التشفير بالمفتاح العام، لأن هذه الطريقة تؤكد هوية مرسل الرسالة، ففي البداية يجب أن يعرف كل طرف المفتاح العام للطرف الآخر، بعد ذلك عند بدء الاتصال بينهما يجب على كل طرف تأكيد هويته للطرف الآخر بإرسال توقيعه الالكتروني الذي أنشأه باستخدام مفتاحه الخاص، في هذه الحالة المهاجم الموجود في الوسط (الرجل الوسطي) سوف يفشل في تنفيذ هجومه لأنه غير قادر على معرفة التوقيع الالكتروني للطرفين بدوم معرفة مفتاحيهما السريين المستخدمان في إنشاء التواقيع.

3.4.4 هجوم النص المشفر فقط Ciphertext-only attack

يستخدم هذا الهجوم عندما يكون المهاجم لا يعرف شيئا عن محتوى الرسالة الأصلية، وبالتالي فإنه لا يملك إلا العمل على الرسالة المشفرة. معظم طرق التشفير الحديثة محصنة ضد هذا النوع من الهجمات.

3.4.5 هجوم البحث الشامل Brute Force attack

يتم فيه كسر التشفير بتجربة عدد كبير من الاحتمالات، مثلا تجربة جميع المفاتيح المحتملة بالترتيب لفك تشفير الرسالة المشفرة، في معظم مخططات التشفير الاحتمالات النظرية لهجوم البحث الشامل تؤخذ في الاعتبار.

اختيار الطول المناسب للمفتاح يقلل وقد يمنع من استخدام هجوم البحث الشامل لأنه كلما زاد طول المفتاح يزداد الوقت اللازم لإيجاده باستخدام هذا الهجوم.

3.5 ملخص

تحليل التشفير يحتاج إلى دراسة واسعة وخبرة طويلة ومثابرة غير عادية ومعرفة بطبيعة المراسلة والمتراسلين. كما أن الحظ المجرد قد يؤدي دوراً في حل بعض طرق التشفير ففي بعض الأحيان يكون من المستحيل حل رسالة مشفرة قصيرة حتى ولو كان نظامها بسيطاً جداً، ولكن محللي التشفير الذين تتوافر لديهم المهارة والقدرة والوسائل التحليلية والوقت الكافي إضافة إلى عدد من الرسائل والمعلومات الجانبية عن طبيعة الرسائل والمتراسلين، قد يستطيعون حل أنظمة التشفير البالغة التعقيد.

يعتمد كسر التشفير في كثير من الأحيان على دراسة الخصائص الإحصائية للغة المستخدمة، إذ تمتاز اللغات الطبيعية بأنها ذات تركيب إحصائي بالغ التعقيد. وقد يختلف التوزيع الإحصائي وعدد الحروف من لغة إلى أخرى، ولكن جميع اللغات تمتاز بتوزيع غير منتظم وبتكرارية عالية نسبيًا، تتيح للناطقين بهذه اللغات معرفة المعنى المقصود حتى لو فقدت أو تغيرت بعض أجزاء الرسالة أو مقاطع الكلام. وفي اللغة العربية يختلف التكرار النسبي من حرف إلى آخر، إذ إن بعض الحروف كالألف مثلاً، ترد في النصوص والرسائل أكثر بكثير من حروف أخرى كحرف الظاء أو الغين. بل إن احتمال ورود حرف معين في نص مكتوب يعتمد اعتمادًا كبيرًا على الحرف أو الحروف السابقة له. وفي العصر الحديث، فإن تحليل التشفير علم قائم بذاته ويعتمد على علوم أخرى كثيرة أهمها علوم الرياضيات وعلوم الإحصاء، والحاسوب،

والهندسة الإلكترونية وعلوم اللغة والأصوات. كما قد تتم الاستعانة بمتخصصين في علوم أخـــرى مســـاندة كعلم الاجتماع وعلم النفس والتاريخ وغيرها، حسب طبيعة الموضوع وأهمية الرسالة.

الفصل الرابع

التحليل والتصميم

باستخدام لغة UML

يعتبر هذا الفصل مجهوداً شخصياً من قبلنا لذلك لا يمكن استخدامه كمرجع.

4.1 مقدمة

تم استخدام لغة النمذجة الموحدة ((Unified Modeling Language (UML) في تحليل وتصـــميم هـــذه المنظومة. لتفصيل أكثر على لغة النمذجة الموحدة انظر الملحق الأول.

4.2 أهداف استخدام لغة النمذجة الموحدة في تحليل وتصميم منظومتنا (منظومة التشفير)

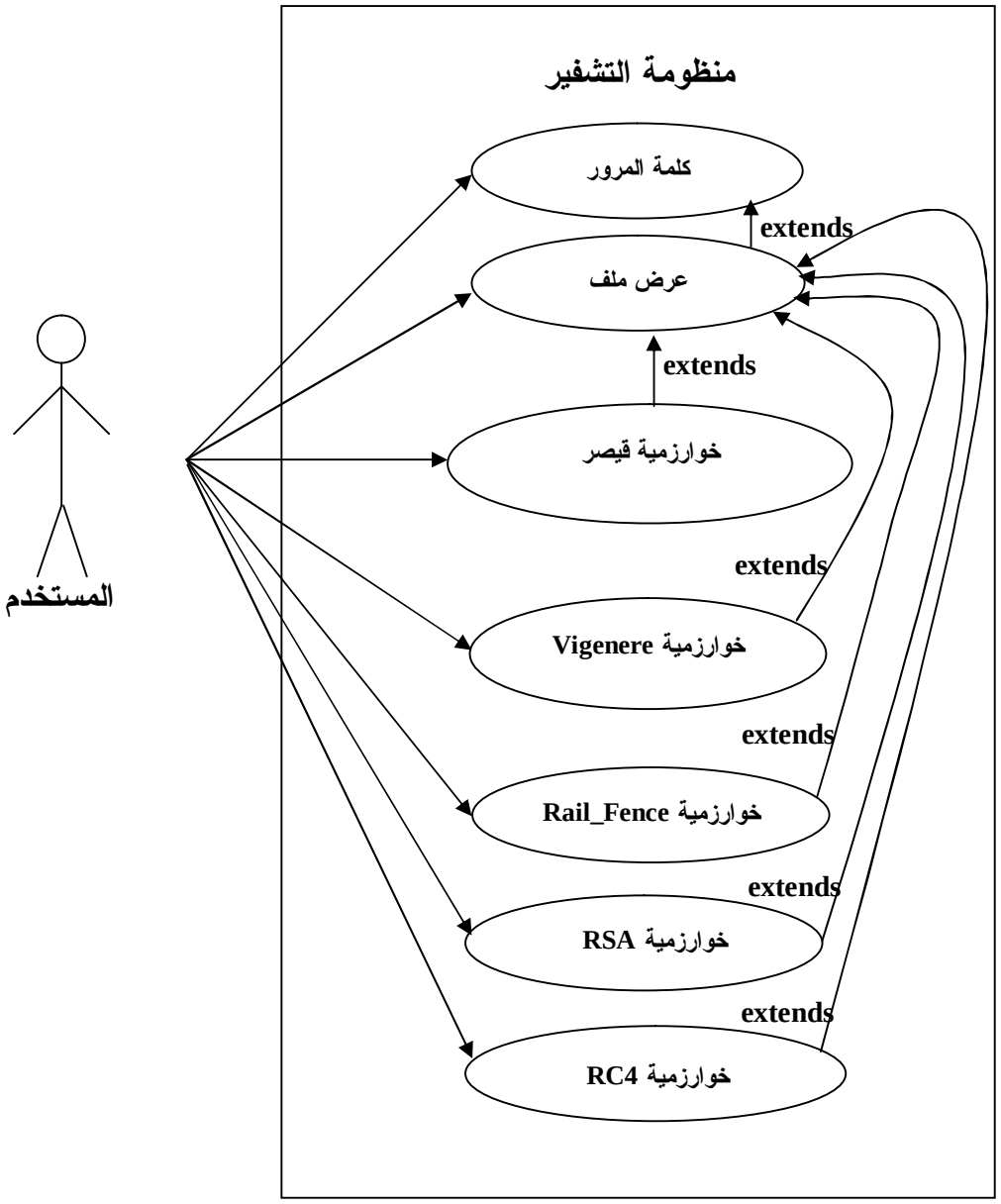
- تتبع عمل خوارزميات التشفير الموجودة في المنظومة والتي تقـــوم بتشـــفير البيانـــات مـــن اجـــل المحافظة على سريتها.
- الرغبة في فهم التحليل والتصميم باستخدام لغة النمذجة الموحدة واستخدام مخططاتهـــا والتـــدريب عمليا عليها.

4.3 التحليل باستخدام لغة النمذجة الموحدة

قمنا باستخدام مخططات لغة النمذجة الموحدة في وصف وتحليل منظومتنا، وسوف نقوم بإعطاء نبذة موجزة عن كل مخطط.

- **مخطط حالة الاستخدام Uses Case Diagram**

في هذا المخطط قمنا بتوضيح العمليات التي تقوم بها المنظومة من منظور المستخدم الخارجي **ACTOR**، مخطط حالة الاستخدام لمنظومتنا موضح في الشكل(15).



شكل (15): مخطط حالة الاستخدام.

مخطط حالة الاستخدام يتكون من العمليات التالية:

- **كلمة المرور**

لكي يستطيع المستخدم استعمال المنظومة لابد أن يقوم بإدخال اسم المستخدم وكلمة المرور، وإذا كـــان المستخدم جديداً يجب أن يقوم بعملية التسجيل حيث يختار اسم مستخدم وكلمة مرور خاصة به ويضغط على زر "تسجيل" لكي تتم إضافته، بعد ذلك يستطيع استخدام المنظومة، كذلك يستطيع المستخدم تغييـــر

كلمة المرور الخاصة به وذلك بكتابة اسم المستخدم وكلمة المرور القديمة ثم يدخل كلمة المرور الجديدة ويضغط زر "تغيير"، كذلك يستطيع المستخدم إلغاء الاسم الخاص به وذلك بكتابة اسم المستخدم وكلمة المرور والضغط على زر "حذف".

- **عرض ملف**

هناك ثلاث اختيارات لعرض الملف وهي:

- كتابة نص باستخدام لوحة المفاتيح، ونستطيع تخزين هذا النص إذا أردنا بالضغط على زر "تخزين النص".
- فتح ملف نصي سبق تخزينه بشرط ألا يكون فارغاً، وذلك بالضغط على زر "فتح ملف".
- اختيار ملف من الملفات التي سبق معالجتها (تشفيرها أو فك تشفيرها)، ونستطيع حذف أي ملف من هذه الملفات بالضغط على زر "حذف".

بعد عملية عرض الملف يتم اختيار خوارزمية تشفير المراد استخدامها، والخوارزميات المتوفرة هي: قيصر، Vigenere، Rail Fence، RSA، RC4.

- **خوارزمية قيصر**

تقوم هذه الخوارزمية بتشفير أوفك تشفير الملف المعروض وذلك بالضغط على أحد الزرين "تشفير أو "فك تشفير" على الترتيب، ولكن قبل إنجاز هاتين العمليتين لابد من إدخال مقدار الإزاحة، بعد ذلك النص الذي ينتج نستطيع تخزينه بالضغط على زر "تخزين".

- **خوارزمية Vigenere**

تقوم هذه الخوارزمية بتشفير أوفك تشفير الملف المعروض وذلك بالضغط على أحد الزرين "تشفير أو "فك تشفير" على الترتيب، ولكن قبل إنجاز هاتين العمليتين لابد من إدخال الكلمة المفتاحية، بعد ذلك النص الذي ينتج نستطيع تخزينه بالضغط على زر "تخزين".

- **خوارزمية Rail_Fence**

تقوم هذه الخوارزمية بتشفير أوفك تشفير الملف المعروض وذلك بالضغط على أحد الزرين "تشـــفير" أو "فك تشفير" على الترتيب، ولكن قبل إنجاز هاتين العمليتين لابد من إدخال عدد صفوف المصفوفة، بعـــد ذلك النص الذي ينتج نستطيع تخزينه بالضغط على زر "تخزين".

- **خوارزمية RSA**

تقوم هذه الخوارزمية بتشفير أوفك تشفير الملف المعروض وذلك بالضغط على أحد الزرين "تشـــفير" أو "فك تشفير" على الترتيب، ولكن قبل إنجاز هاتين العمليتين لابد من إجراء عملية توليد المفتاحين العـــام والخاص وذلك بالضغط على زر "توليد المفتاحين"، بعد ذلك النص الذي ينتج نستطيع تخزينه بالضـــغط على زر "تخزين".

- **خوارزمية RC4**

تقوم هذه الخوارزمية بتشفير أوفك تشفير الملف المعروض وذلك بالضغط على أحد الزرين "تشـــفير" أو "فك تشفير" على الترتيب، ولكن قبل إنجاز هاتين العمليتين لابد من إدخال مفتاح التشفير، بعد ذلك النص الذي ينتج نستطيع تخزينه بالضغط على زر "تخزين".

4.3.2 مخطط الصنف Class Diagram

يبين هذا المخطط البنية الثابتة للأصناف الموجودة في نظامنا، والجدول (5) يوضح هذه الأصناف.

اسم الصنف	وصف الصنف
إدخال ملف	يحتوي على المعلومات الخاصة بالملف المطلوب معالجته.
خوارزمية قيصر	يحتوي على المعلومات الخاصة بخوارزمية قيصر.
خوارزمية Rail Fence	يحتوي على المعلومات الخاصة بخوارزمية Rail Fence.
خوارزمية vigenere	يحتوي على المعلومات الخاصة بخوارزمية Vigenere.
خوارزمية RC4	يحتوي على المعلومات الخاصة بخوارزمية RC4.
خوارزمية RSA	يحتوي على المعلومات الخاصة بخوارزمية RSA.

جدول(5): الأصناف الموجودة في المنظومة.

وسوف نقوم بعرض كل صنف مع المتغيرات والعمليات الخاصة به.

عرض ملف
+ اسم الملف
+ نص الملف
+ مشيد

خوارزمية قيصر
- مقدار الإزاحة
+ تشفير
+ فك تشفير
+ مشيد

خوارزمية Rail Fence
- عدد الصفوف
+ تشفير
+ فك تشفير
+ مشيد

خوارزمية Vigenere
- الكلمة المفتاحية
+ تشفير
+ فك تشفير
+ مشيد

خوارزمية RC4
- مفتاح التشفير
+ تشفير + فك تشفير + مشيد

خوارزمية RSA
- المفتاح العام - المفتاح الخاص - حاصل الضرب
+ تشفير + فك تشفير + مشيد

4.3.3 مخطط الكائن Object Diagram

هو عبارة عن صورة لحظية لمنظومتنا.

عرض ملف:
اسم الملف= " test.rtf "
نص الملف=" this is a test "

خوارزمية قيصر:
مقدار الإزاحة= 6

خوارزمية Rail Fence:
عدد الصفوف= 4

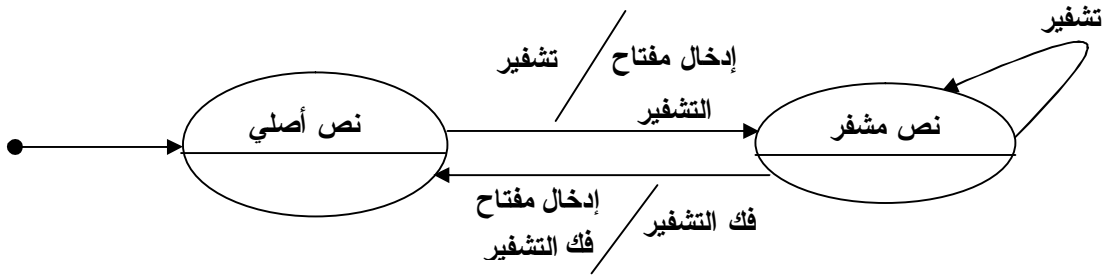
خوارزمية Vigenere:
الكلمة المفتاحية= " welcome "

خوارزمية RC4:
مفتاح التشفير= " test123 "

خوارزمية RSA:
المفتاح العام= 53
المفتاح الخاص = 4969
حاصل الضرب = 264457

4.3.4 مخطط الحالة State diagram

الشكل (16) يبين الحالات الممكنة للعمليات التي يتم إجرائها على الملفات بعد اختيار خوارزمية التشفير المطلوبة، ففي البداية تكون الحالة أن النص غير مشفر ولكن بعد تنفيذ حدث تشفير وإدخال مفتاح التشفير تصبح الحالة أن النص مشفر وعند تنفيذ الحدث تشفير مرة أخرى لا تتغير الحالة، ولكن عند تنفيذ حدث فك التشفير تصبح الحالة أن النص أصلي (غير مشفر).



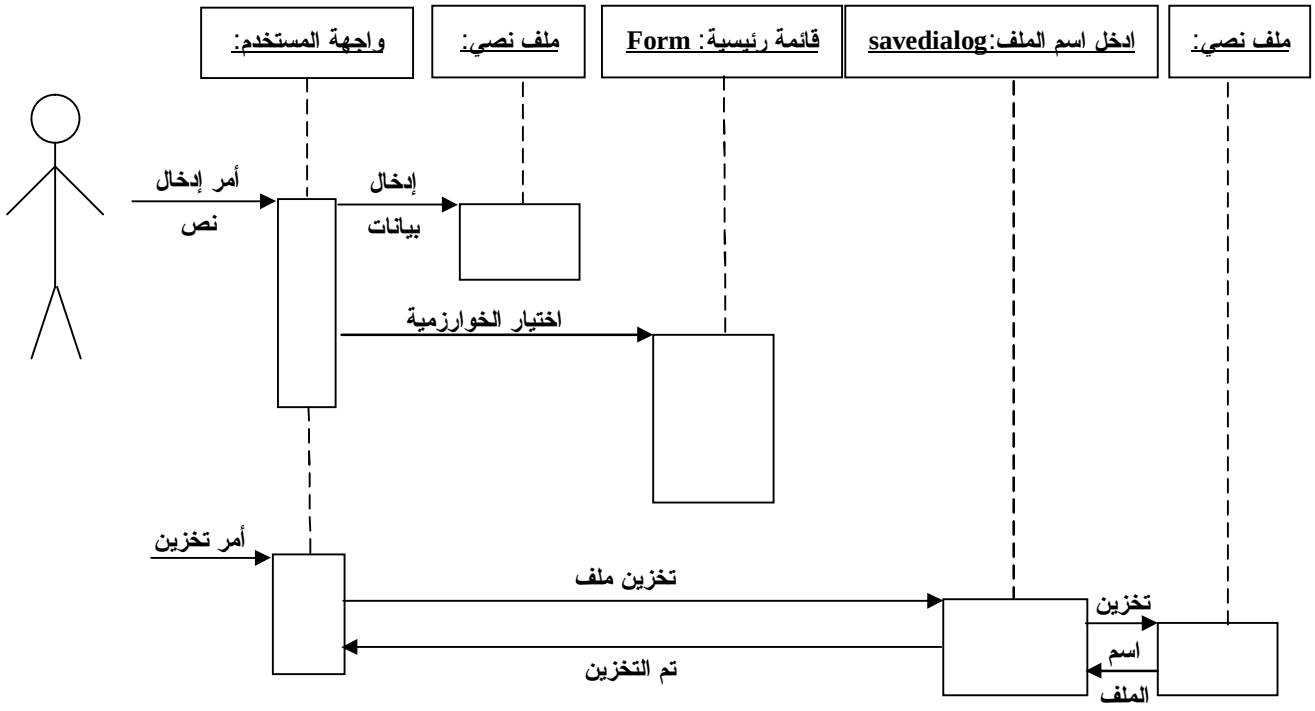
شكل(16): مخطط الحالة.

4.3.5 مخطط التتابع Sequence Diagram

مخطط التتابع يوضح كيفية تفاعل الكائنات مع بعضها البعض مع التركيز على الوقت، وهو موضح في الأشكال من (17) إلى (25).

- **في حالة إدخال نص**

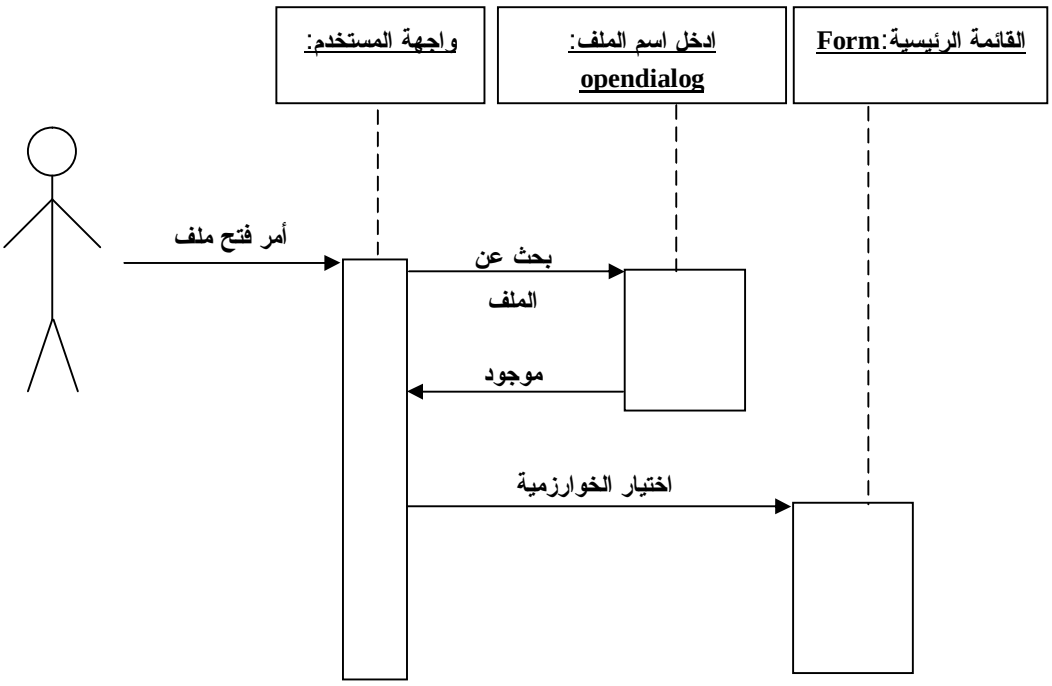
في الشكل (17) يقوم المستخدم بكتابة النص الذي يريد تشفيره في الكائن ملف نصي، بعد ذلك يختار الخوارزمية التي يريد استخدامها.



شكل(17): مخطط التتابع في حالة إدخال ملف.

- **في حالة فتح ملف**

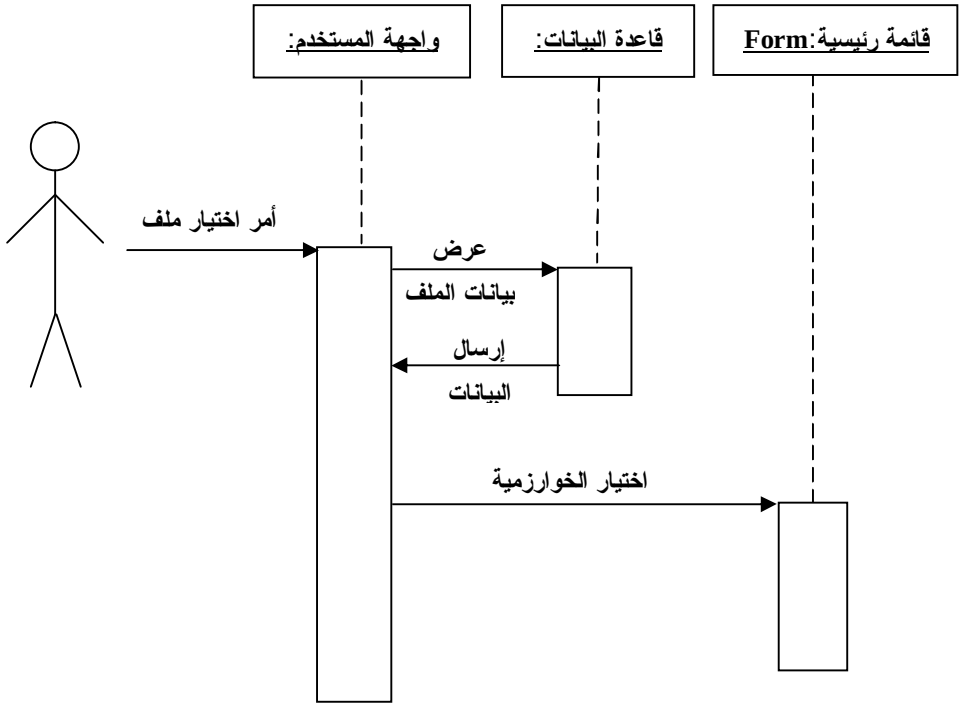
في الشكل (18) يقوم المستخدم بفتح الملف المطلوب تشفيره ويتم البحث عنه فإذا كان موجود يقوم المستخدم باختيار خوارزمية التشفير.



شكل(18): مخطط التتابع في حالة عرض ملف.

- **في حالة اختيار ملف معالج من قبل**

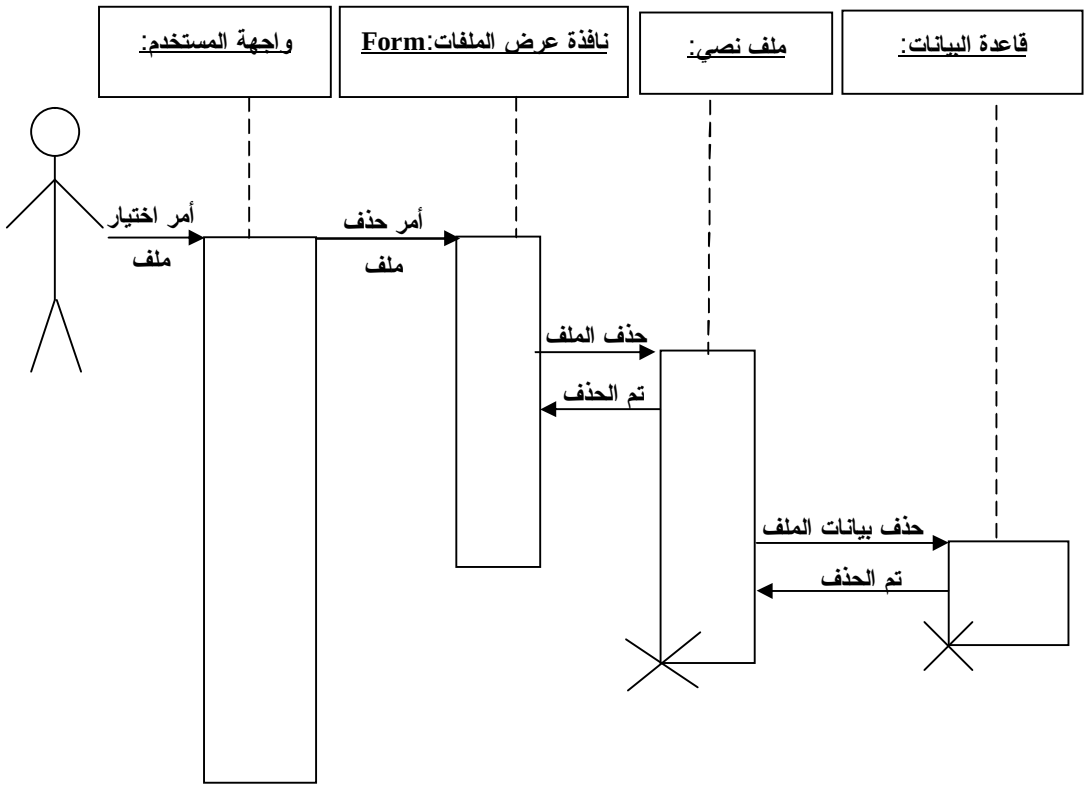
في الشكل (19) يقوم المستخدم باختيار أحد الملفات المعالجة مسبقاً، ومن ثم يتم جلب بيانات هذا الملف من قاعدة البيانات، وبعد ذلك يقوم المستخدم باختيار خوارزمية التشفير.



شكل(19): مخطط التتابع في حالة اختيار ملف معالج من قبل.

- **في حالة حذف ملف معالج**

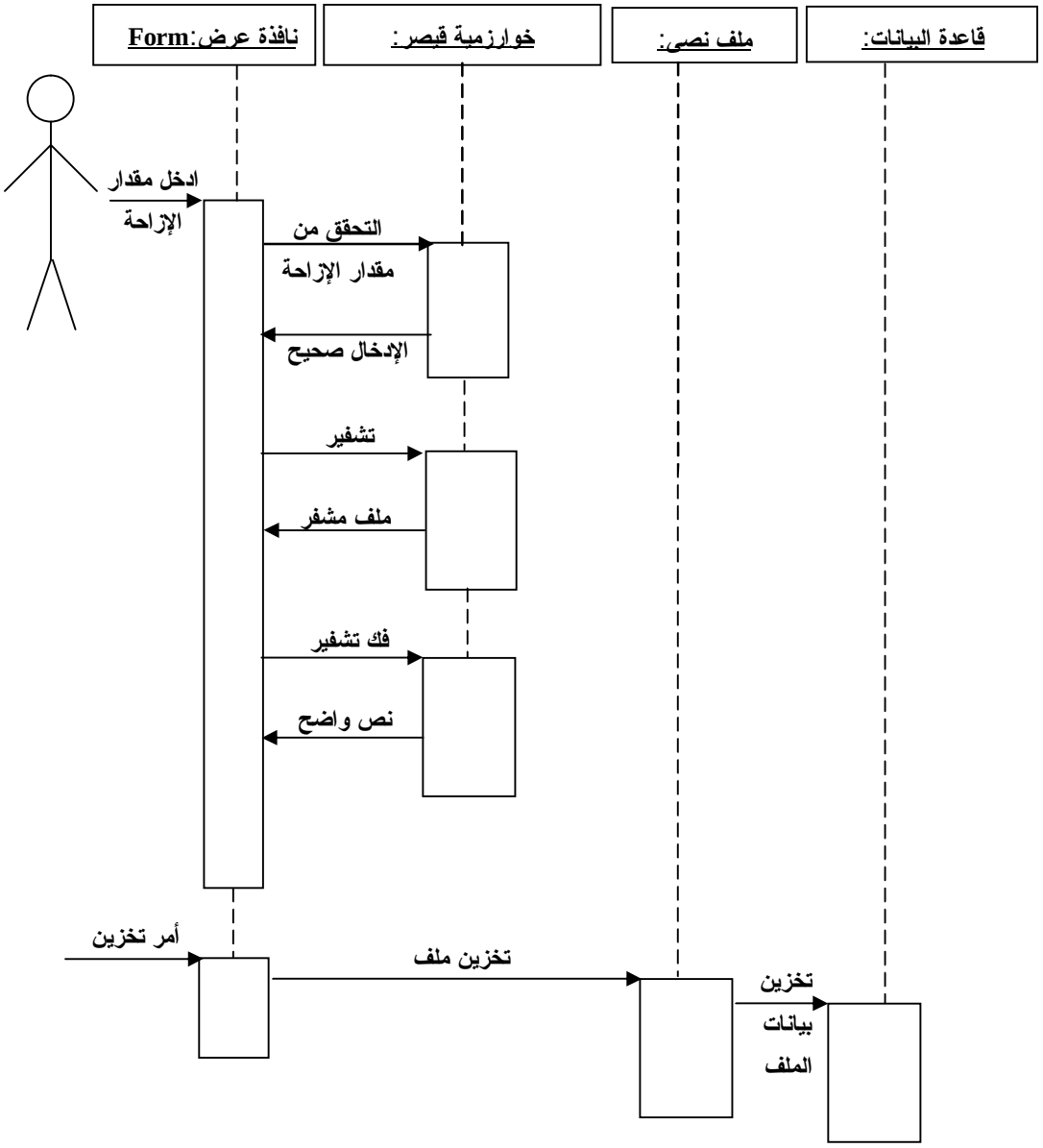
في الشكل (20) يقوم المستخدم باختيار أحد الملفات المعالجة مسبقاً ثم يقوم بتنفيذ أمر حذف ملف فيتم حذف الملف من الكائن ملف نصي ثم يتم حذف البيانات الخاصة به من قاعدة البيانات.



شكل(20): مخطط التتابع في حالة حذف ملف.

- **في حالة التشفير وفك التشفير بخوارزمية قيصر**

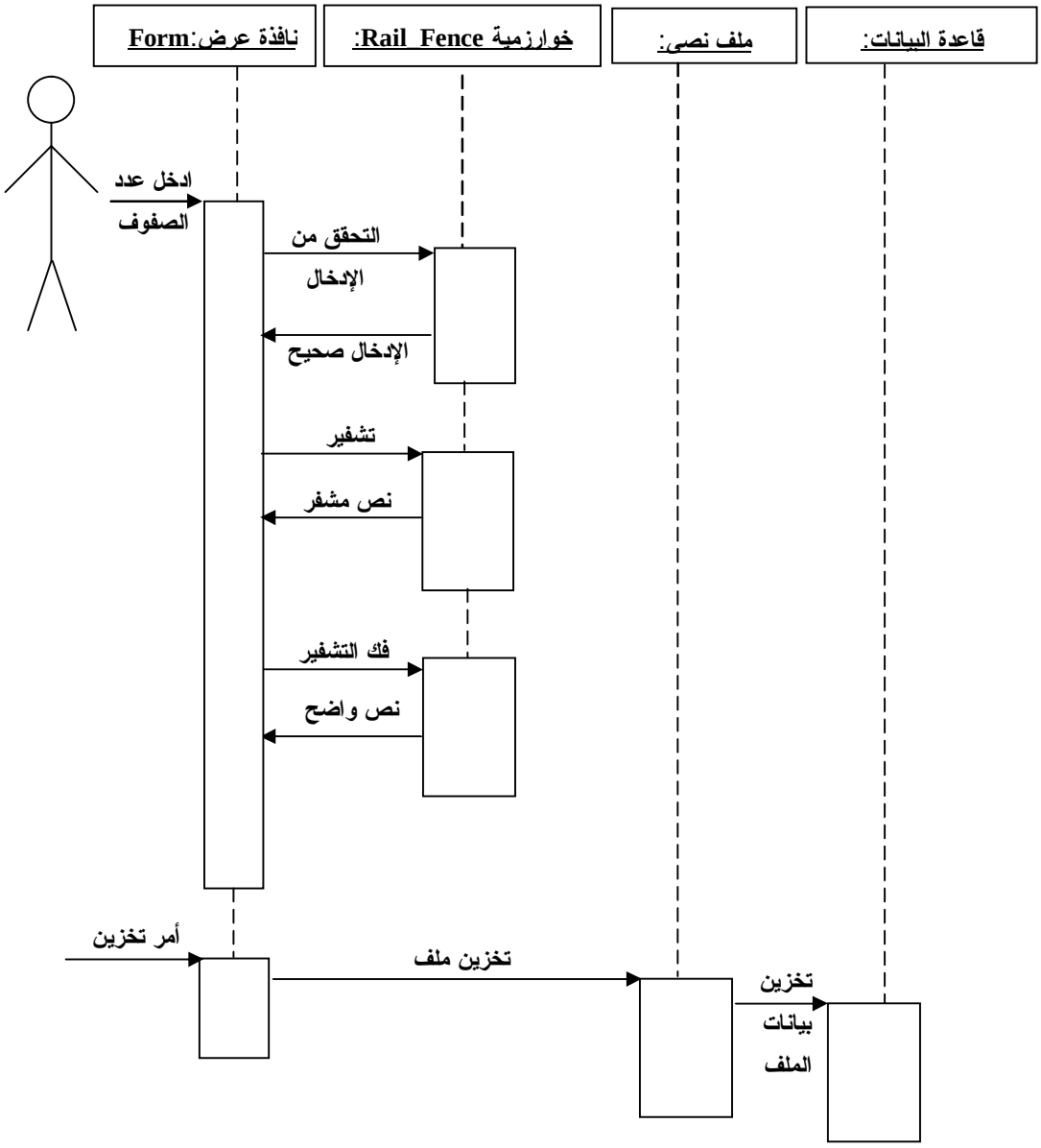
في الشكل (21) يقوم المستخدم بإدخال مقدار الإزاحة حيث يتم التأكد من صحته (التأكد من أنـــه رقـــم وليس حرف أو رمز)، بعد ذلك يقوم المستخدم بتشفير النص فيتم إرجاع النص المشفر أو فـــك تشـــفير النص فيتم إرجاع النص الواضح، ثم يتم تخزين الملف المعالج في ملف وبيانات الملـــف المعـــالج فـــي قاعدة البيانات.



شكل (21): مخطط التتابع لخوارزمية قيصر.

- **في حالة التشفير وفك التشفير بخوارزمية Rail Fence**

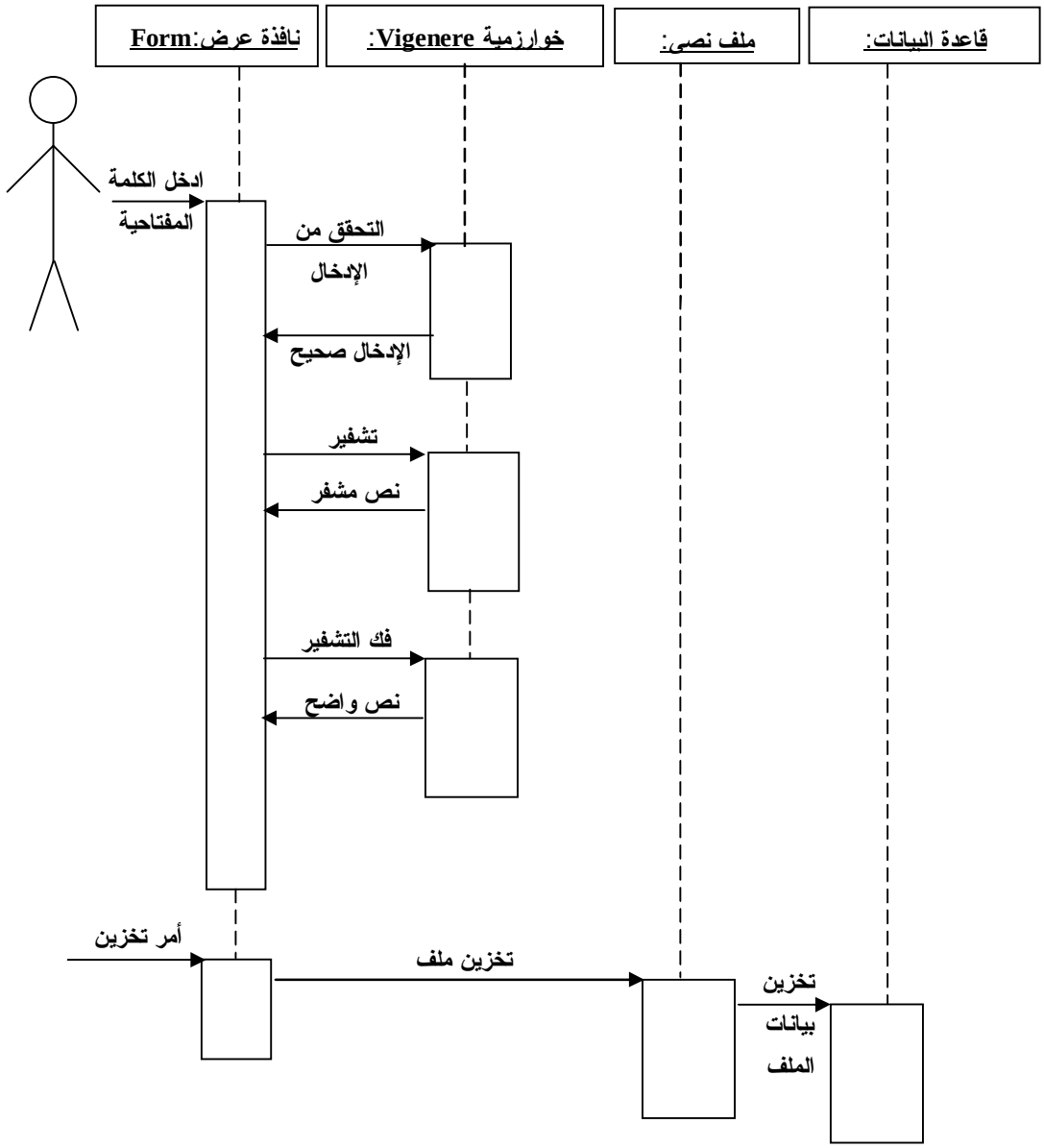
في الشكل (22) يقوم المستخدم بإدخال عدد الصفوف حيث يتم التأكد من صحة الإدخال (التأكد من أنه رقم وليس حرف أو رمز)، بعد ذلك يقوم المستخدم بتشفير النص فيتم إرجاع النص المشفر أو فك تشفير النص فيتم إرجاع النص الواضح، ثم يتم تخزين الملف المعالج في ملف وبيانات الملف المعالج في قاعدة البيانات.



شكل(22): مخطط التتابع لخوارزمية Rail Fence.

- **في حالة التشفير وفك التشفير بخوارزمية Vigenere**

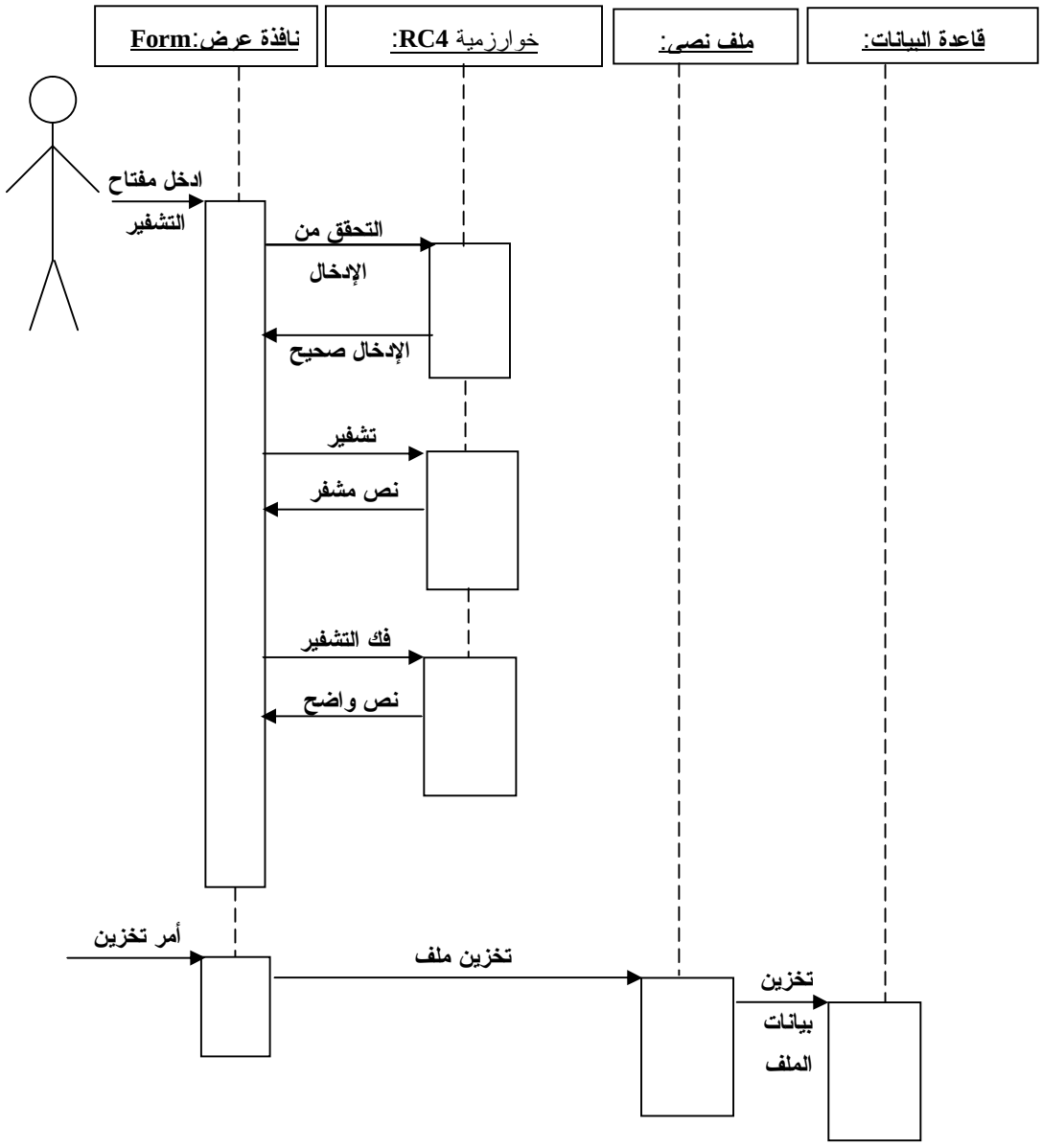
في الشكل (23) يقوم المستخدم بإدخال الكلمة المفتاحية ثم يتم التأكد من صحتها (التأكد من أنها تتكــون من حروف فقط)، بعد ذلك يقوم المستخدم بتشفير النص فيتم إرجاع النص المشفر أو فك تشفير الــنص فيتم إرجاع النص الواضح، ثم يتم تخزين الملف المعالج في ملف وبيانات الملف المعــالج فــي قاعــدة البيانات.



شكل(23): مخطط التتابع لخوارزمية Vigenere.

- **في حالة التشفير وفك التشفير باستخدام خوارزمية RC4**

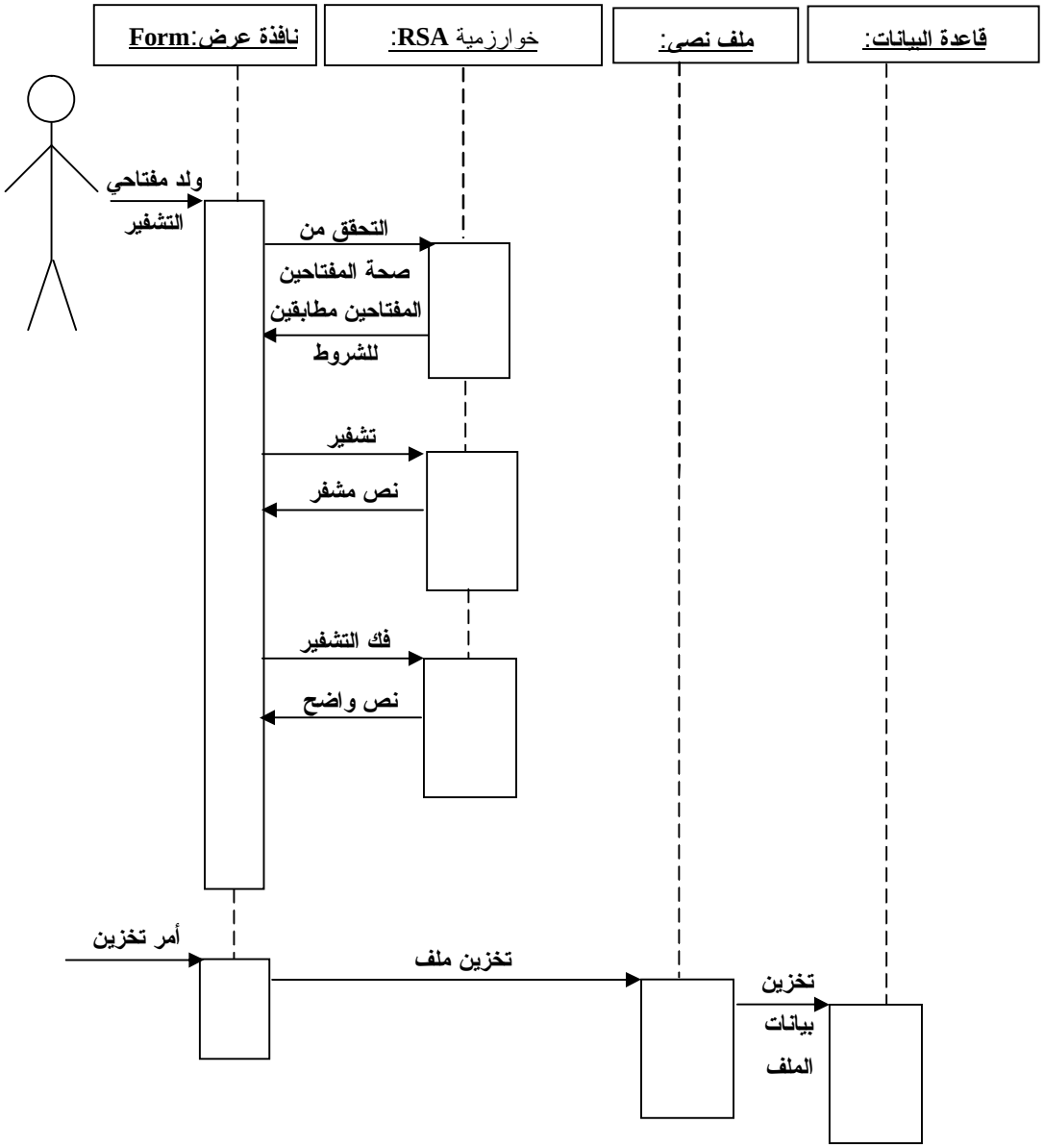
في الشكل (24) يقوم المستخدم بإدخال مفتاح التشفير ثم يتم التأكد من صحة الإدخال، بعد ذلــك يقــوم المستخدم بتشفير النص فيتم إرجاع النص المشفر أو فك تشفير النص فيتم إرجاع النص الواضح، ثم يتم تخزين الملف المعالج في ملف وبيانات الملف المعالج في قاعدة البيانات.



شكل(24): مخطط التتابع لخوارزمية RC4.

- **في حالة التشفير وفك التشفير بخوارزمية RSA**

في الشكل (25) يقوم المستخدم بتوليد مفتاحي التشفير أولاً، بعد ذلك يقوم المستخدم بتشفير النص فيتم إرجاع النص المشفر أو فك تشفير النص فيتم إرجاع النص الواضح، ثم يتم تخزين الملف المعالج في ملف وبيانات الملف المعالج في قاعدة البيانات.



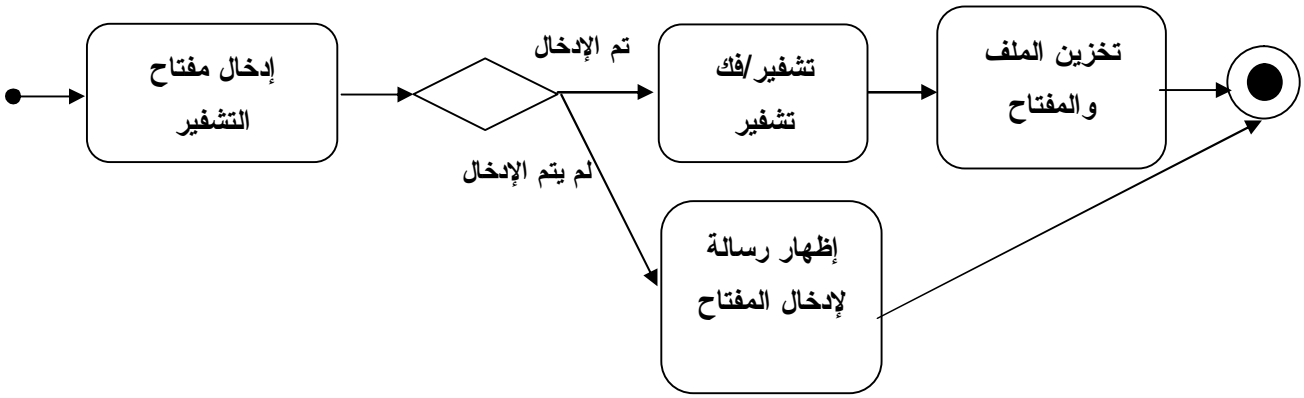
شكل (25): مخطط التتابع للخوارزمية RSA.

4.3.6 مخطط النشاط Activity Diagram

يتم استخدام هذا المخطط في وصف الأنشطة المختلفة التي يقوم بها النظام.

- **في حالة استخدام مفتاح واحد للتشفير**

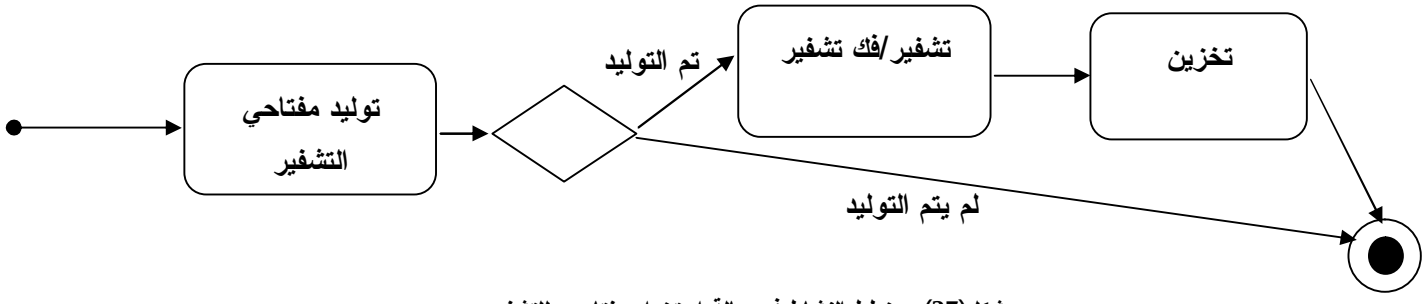
في الشكل (26) نلاحظ أن هناك أربعة عمليات ولكن بسبب وجود نقطة القرار فهناك احتمال أن يتم تنفيذ ثلاثة عمليات فقط وذلك في حالة إدخال مفتاح التشفير، أما إذا لم يتم إدخال مفتاح التشفير فسوم يتم تنفيذ عمليتين فقط، وتوجد نهاية واحدة فقط.



شكل(26): مخطط النشاط في حالة استخدام مفتاح واحد للتشفير.

- **في حالة استخدام مفتاحين للتشفير العام والخاص**

في الشكل (27) نلاحظ أن هناك ثلاث عمليات ولكن بسبب وجود نقطة القرار فهناك احتمال أن يتم تنفيذ عملية واحدة فقط وذلك في حالة لم يتم توليد مفتاحي التشفير، أما إذا يتم توليد المفتاحين فسوف يتم تنفيذ العمليات الثلاث كلها، وتوجد نهاية واحدة فقط.

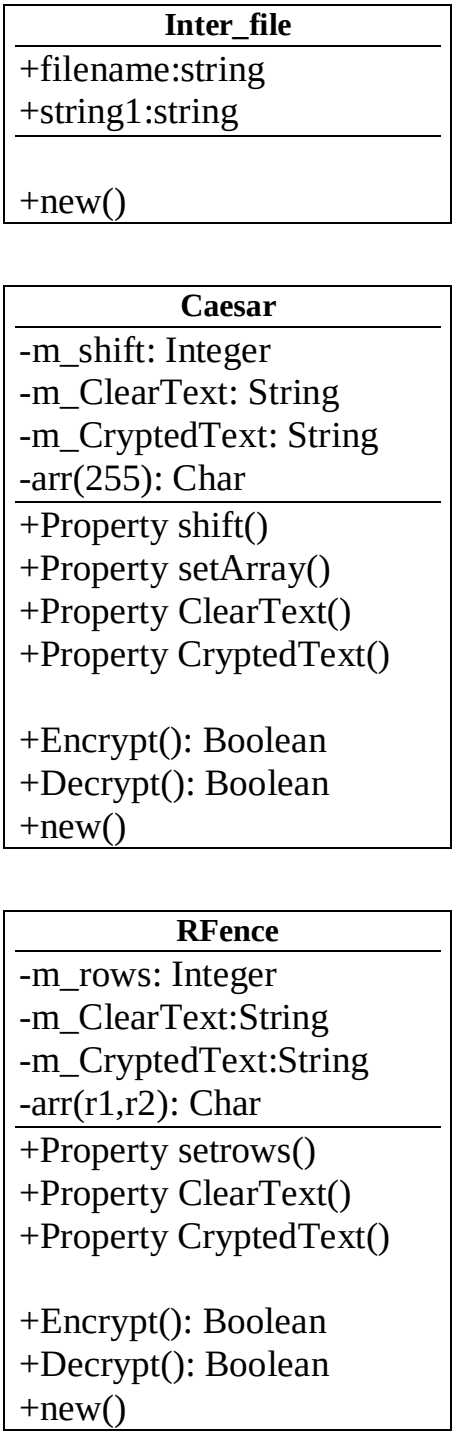


شكل(27): مخطط النشاط في حالة استخدام مفتاحين للتشفير.

4.4 التصميم باستخدام لغة النمذجة الموحدة

4.4.1 مخطط الصنف Class Diagram

الأصناف الموجودة في المنظومة هي:



vigenr
-m_EncKey: string -m_ClearText: String -m_CryptedText: String -text(255): Char
+Property EncryptionKey() +Property setArray() +Property ClearText() +Property CryptedText() +Encrypt(): Boolean +Decrypt(): Boolean +new()

RC4Engine
+CryptArray(): Long -max: Byte -state(max): Byte -userKey: String
-KSA() +CryptReturnString(val): String -Swap(a,b): byte +New()

RSAEngine
-p: integer -q: integer +publickey: integer +privatekey: integer + mult: integer
+Encrypt(): Boolean +Decrypt(): Boolean +New()

وسوف نقوم بتوضيح فكرة عمل كل الدوال والإجراءات الموجودة في كل صنف:

- الصنف RC4Engine يحتوي على:

KSA(): يقوم هذا الإجراء بإعداد مصفوفة الحالة.

()CryptReturnString: تقوم هذه الدالة بإنتاج القيم شبه العشوائية ومن ثم تقوم بالتشفير أوفك التشفير.

()Swap: تقوم هذه الدالة بتبديل قيمتين.

- ()Encrypt: هذه الدالة موجودة في جميع الأصناف ماعدا RC4Engine وهي تقوم بعملية تشفير الملف المطلوب.
- ()Decrypt: هذه الدالة موجودة في جميع الأصناف ماعدا RC4Engine وهي تقوم بعملية فك تشفير الملف المطلوب.
- ()new: هو إجراء استخدم مع جميع الأصناف، وهو عبارة عن مشيد يستخدم لتشييد الأصناف في الذاكرة، ويستدعى عند تكوين الكائن أول مرة.
- ()Property….: هذه الدالة استخدمت مع جميع الأصناف، وهي تعتبر خاصية لكل متغير موجود في الصنف، حيث تستخدم لقراءة قيمة المتغير وكذلك لإرسال قيمة المتغير مع إمكانية جعلها تؤدي وظيفة واحدة من هذه الوظيفتين وهنا استخدمت لتؤدي الوظيفتين.

4.4.2 مخطط الكائن Object Diagram

:inter_file
Filename= " test1.rtf " string1= " this is s test "

:Caesar
m_shift=6

:RFence
m_rows= 4

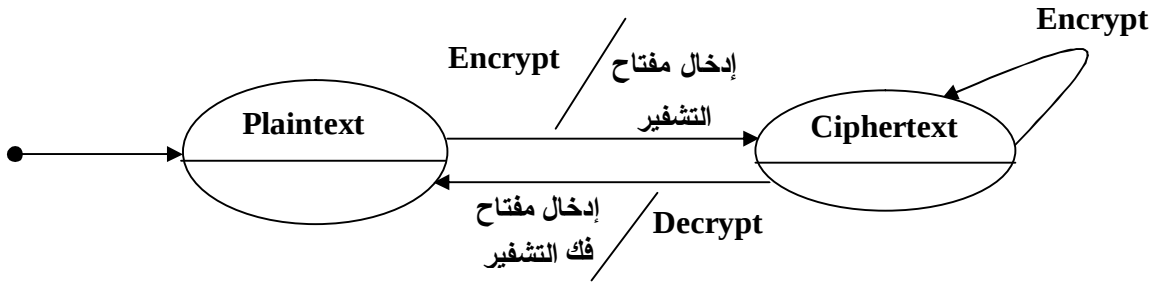
:vigenr
m_EncKey = " welcome "

:RC4Engine
userKey= " 123test "

:RSAEngine
publickey=53 privatekey=4969 mult=4969

4.4.3 مخطط الحالة State diagram

الشكل (28) يبين مخطط الحالة.

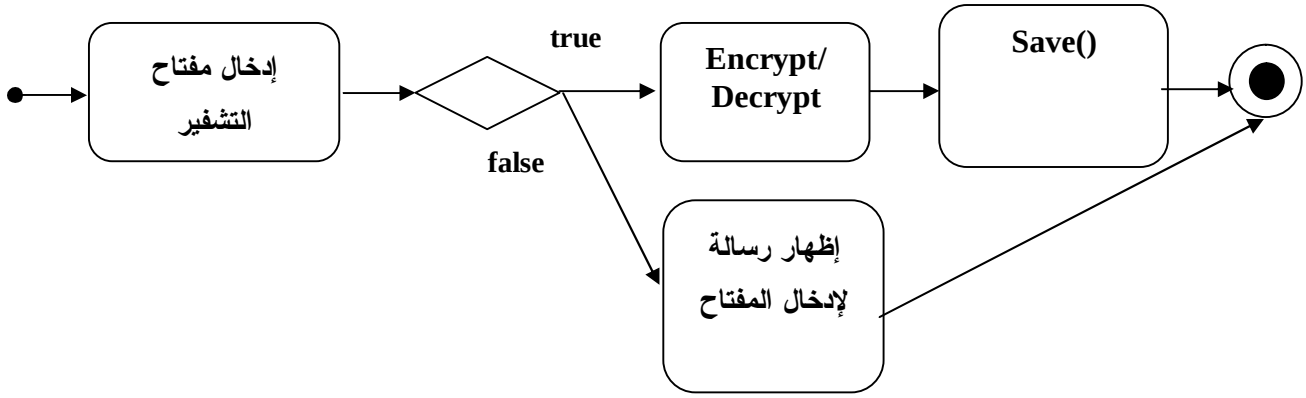


شكل(28): مخطط الحالة.

4.4.4 مخطط النشاط Activity Diagram

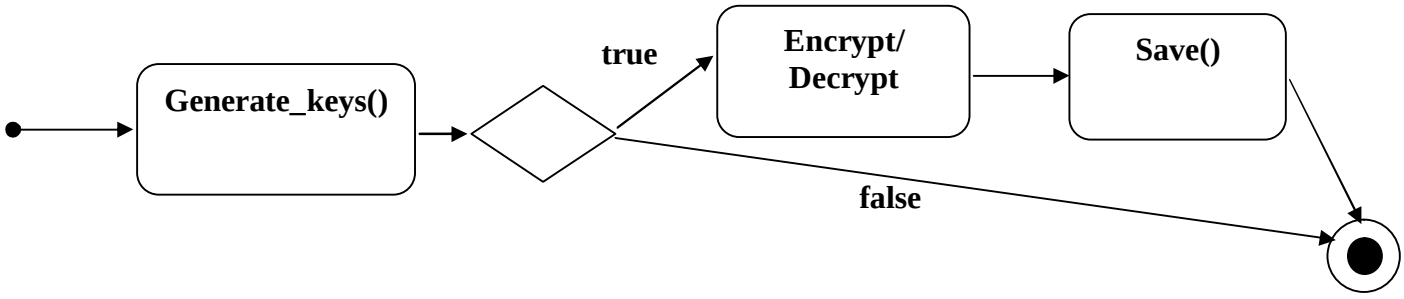
الشكلان (29، 30) يوضحان مخطط النشاط.

- **في حالة استخدام مفتاح واحد للتشفير**



شكل(29): مخطط النشاط في حالة استخدام مفتاح واحد للتشفير.

- **في حالة استخدام مفتاحين للتشفير العام والخاص**

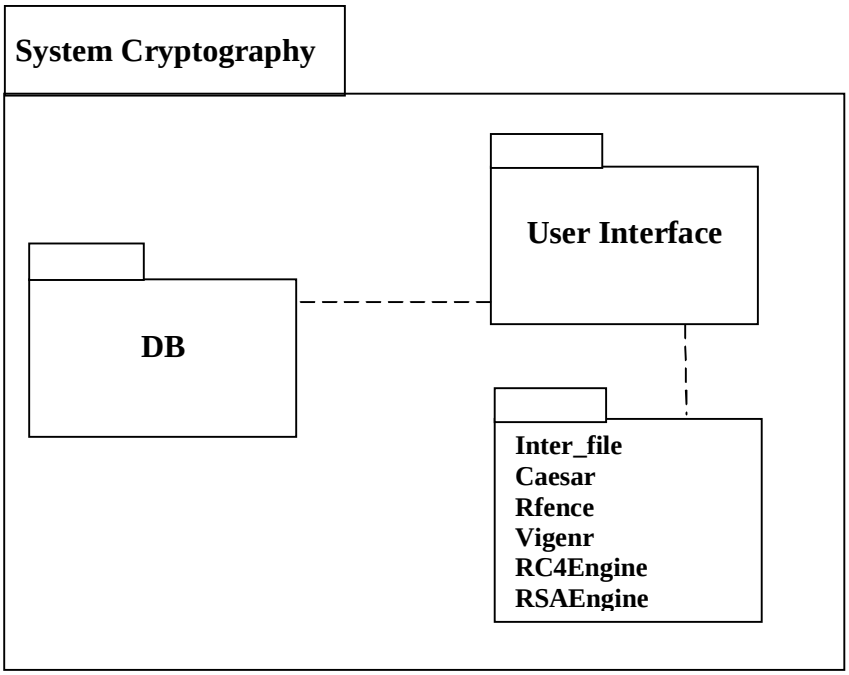


شكل(30): مخطط النشاط في حالة استخدام مفتاحين للتشفير.

4.4.5 مخطط الحزم packages diagram

يعرفنا بمحتويات المنظومة حيث يتم تقسيم المنظومة إلى أجزاء صغيرة يمكن فهمها بسهولة.

منظومتنا تتعامل مع الملفات كما أنها تحتوي على قاعدة بيانات وهذا ما هو موضح في الشكل (31).

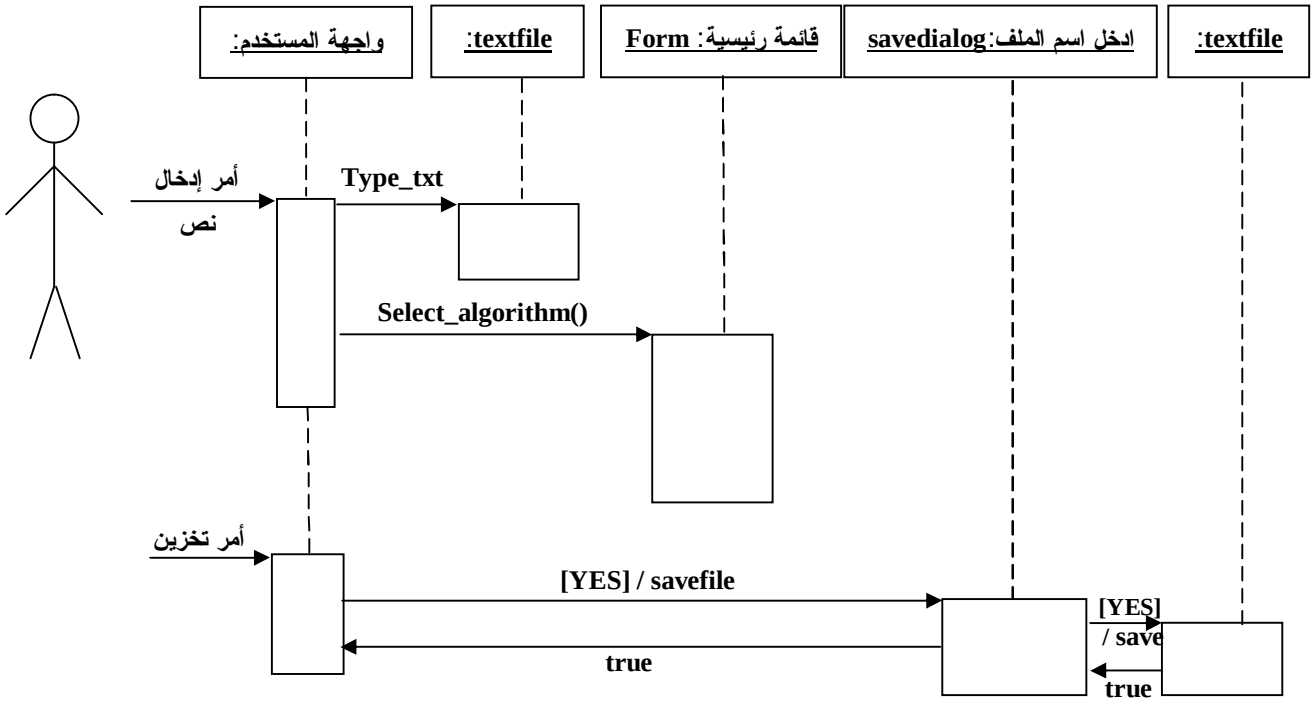


شكل(31): مخطط الحزم.

4.4.6 مخطط التتابع Sequence Diagram

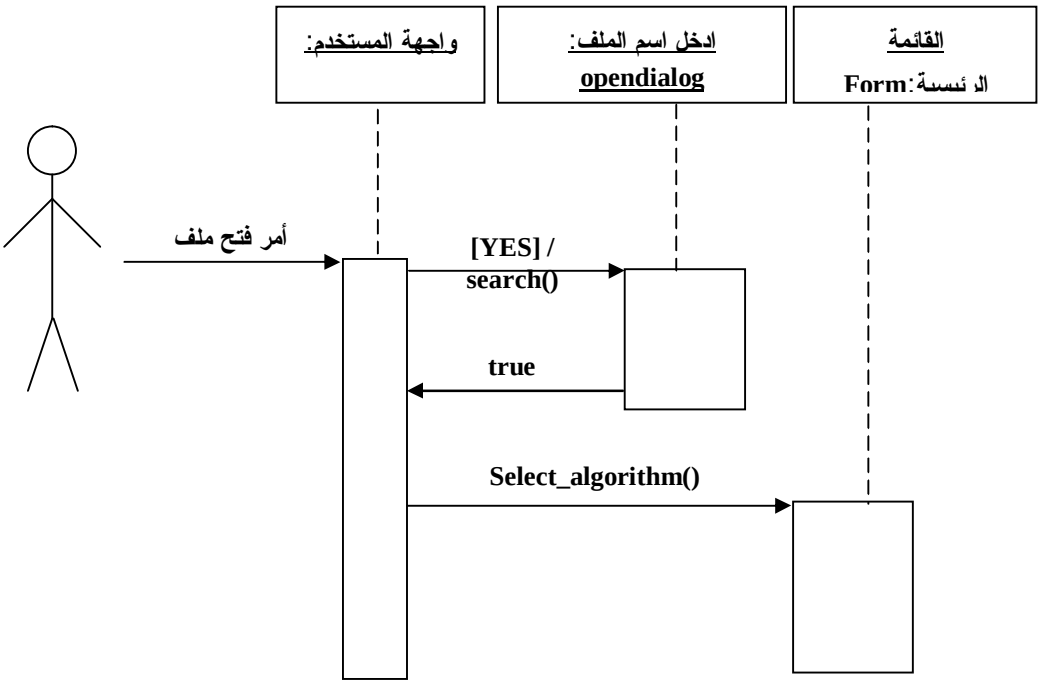
مخطط التتابع موضح في الأشكال من (32) إلى (40).

- **في حالة إدخال نص**



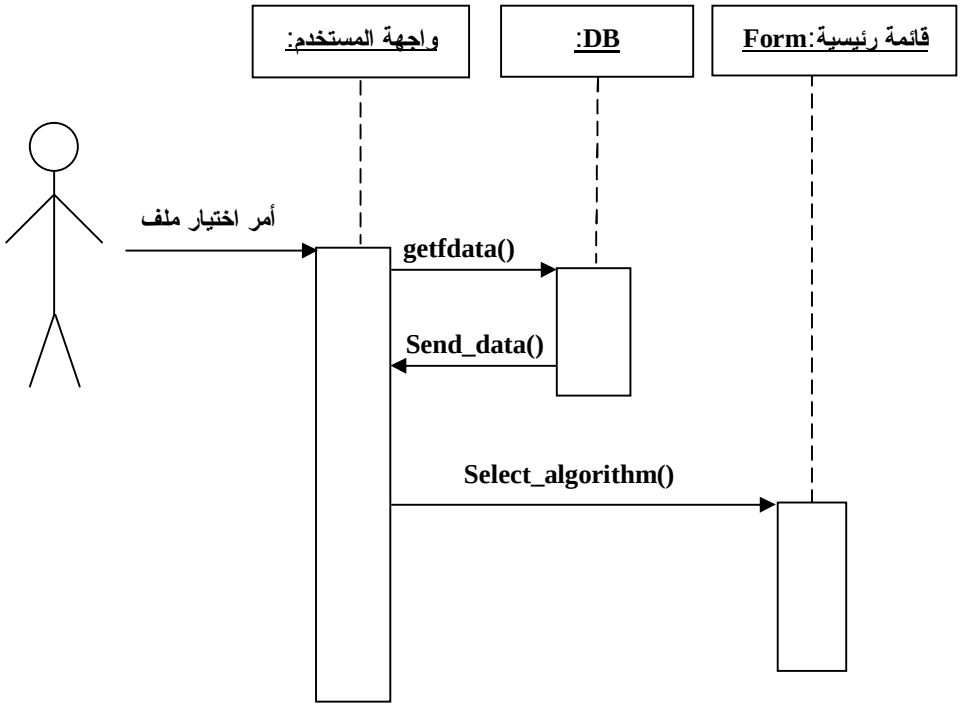
شكل(32): مخطط التتابع في حالة إدخال ملف.

- **في حالة فتح ملف**



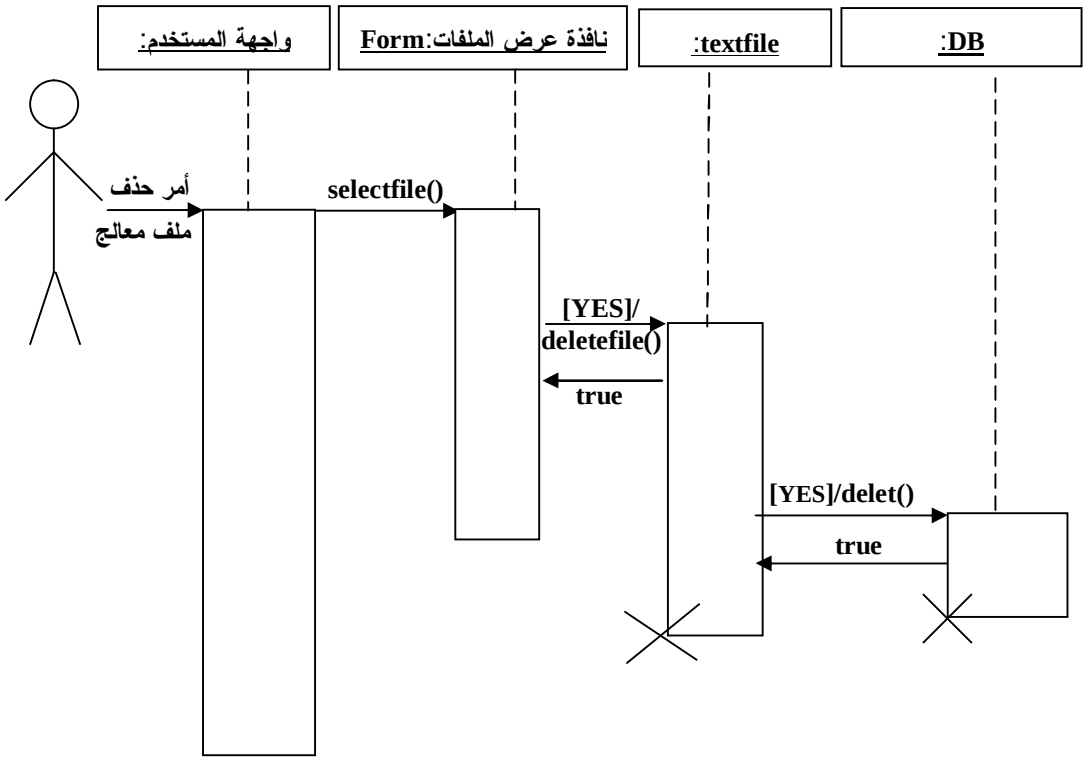
شكل(33): مخطط التتابع في حالة عرض ملف.

- **في حالة اختيار ملف معالج من قبل**



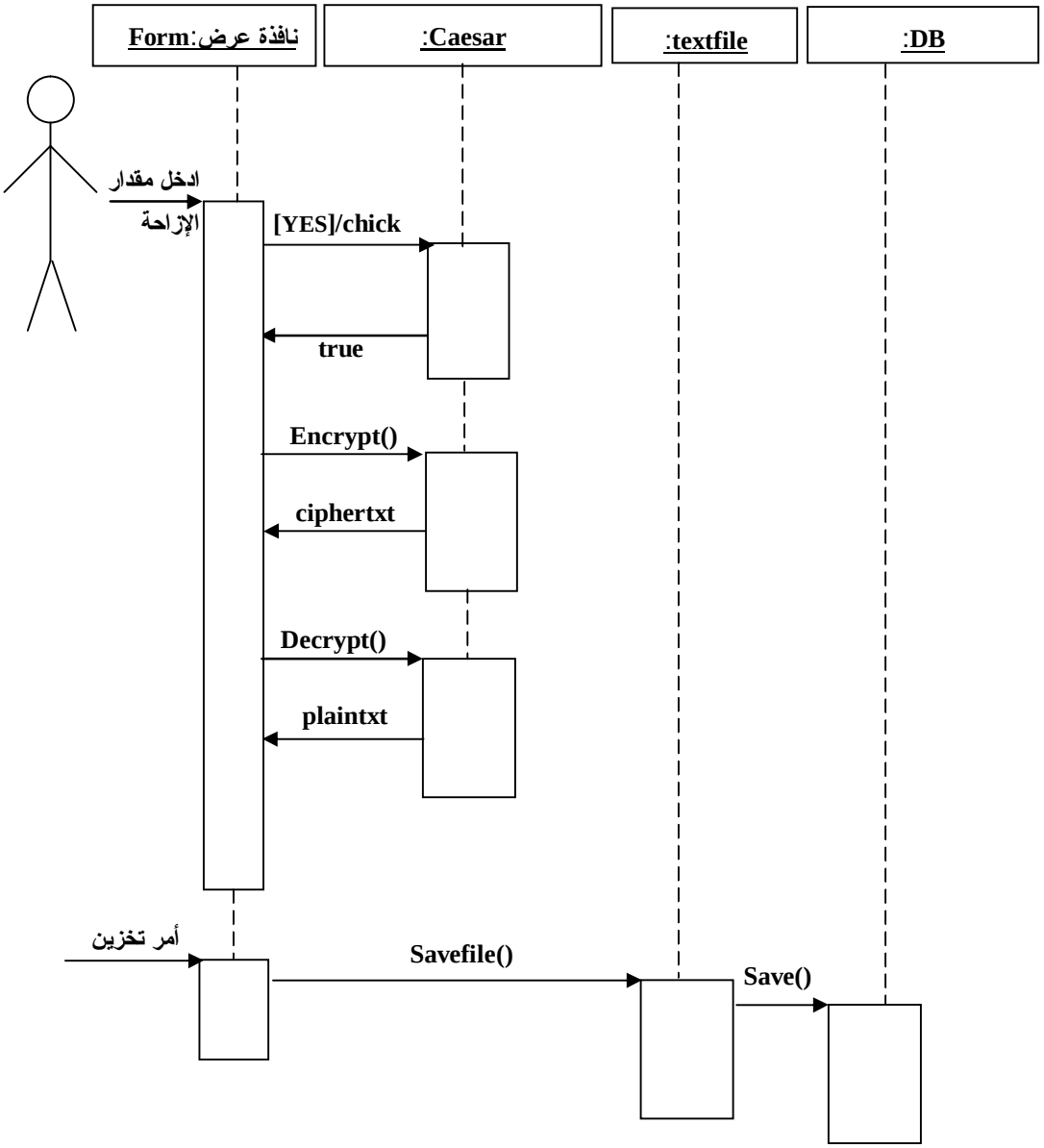
شكل(34): مخطط التتابع في حالة اختيار ملف معالج من قبل.

- **في حالة حذف ملف معالج**



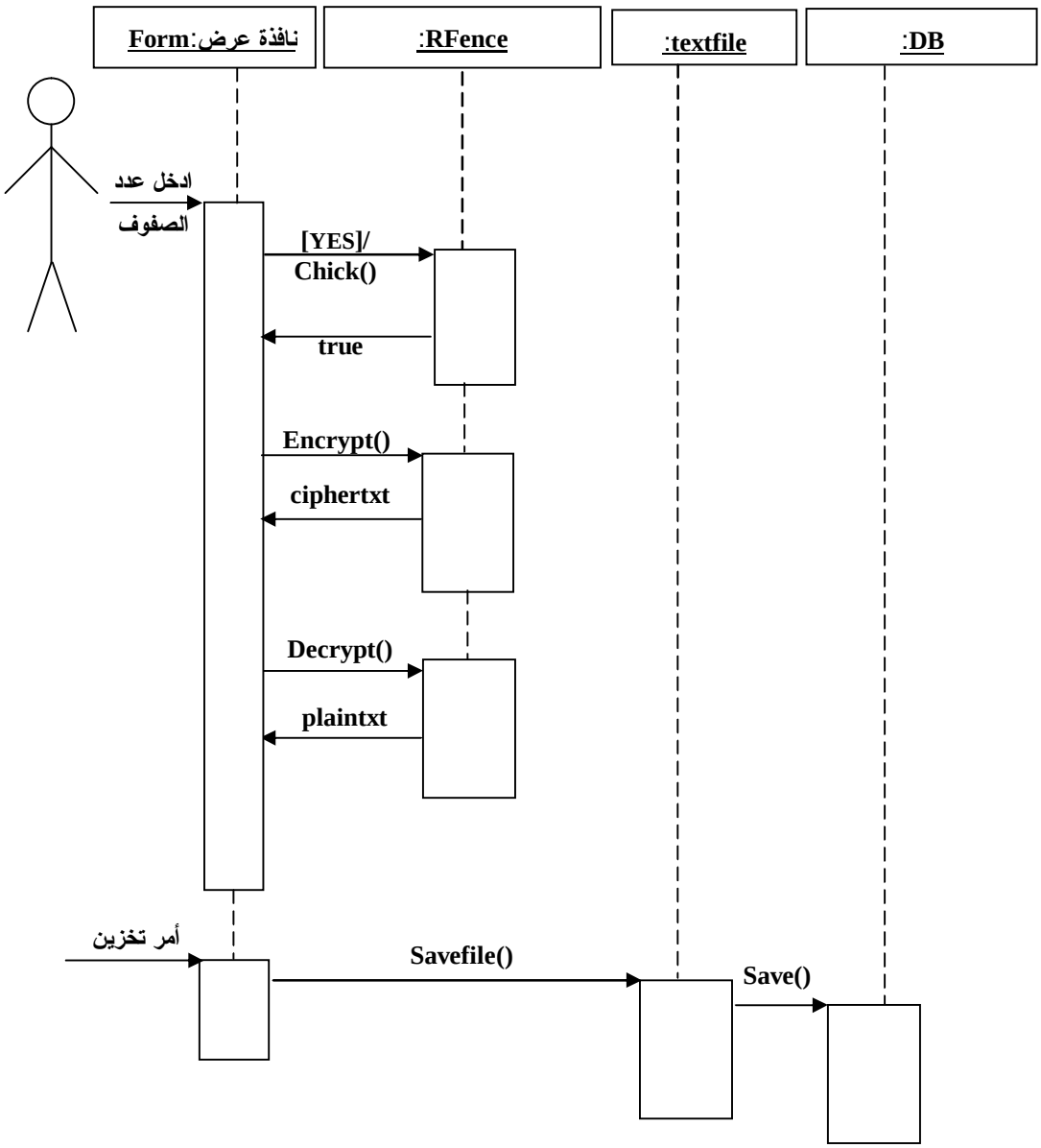
شكل(35): مخطط التتابع في حالة حذف ملف.

- **في حالة التشفير وفك التشفير بخوارزمية قيصر**



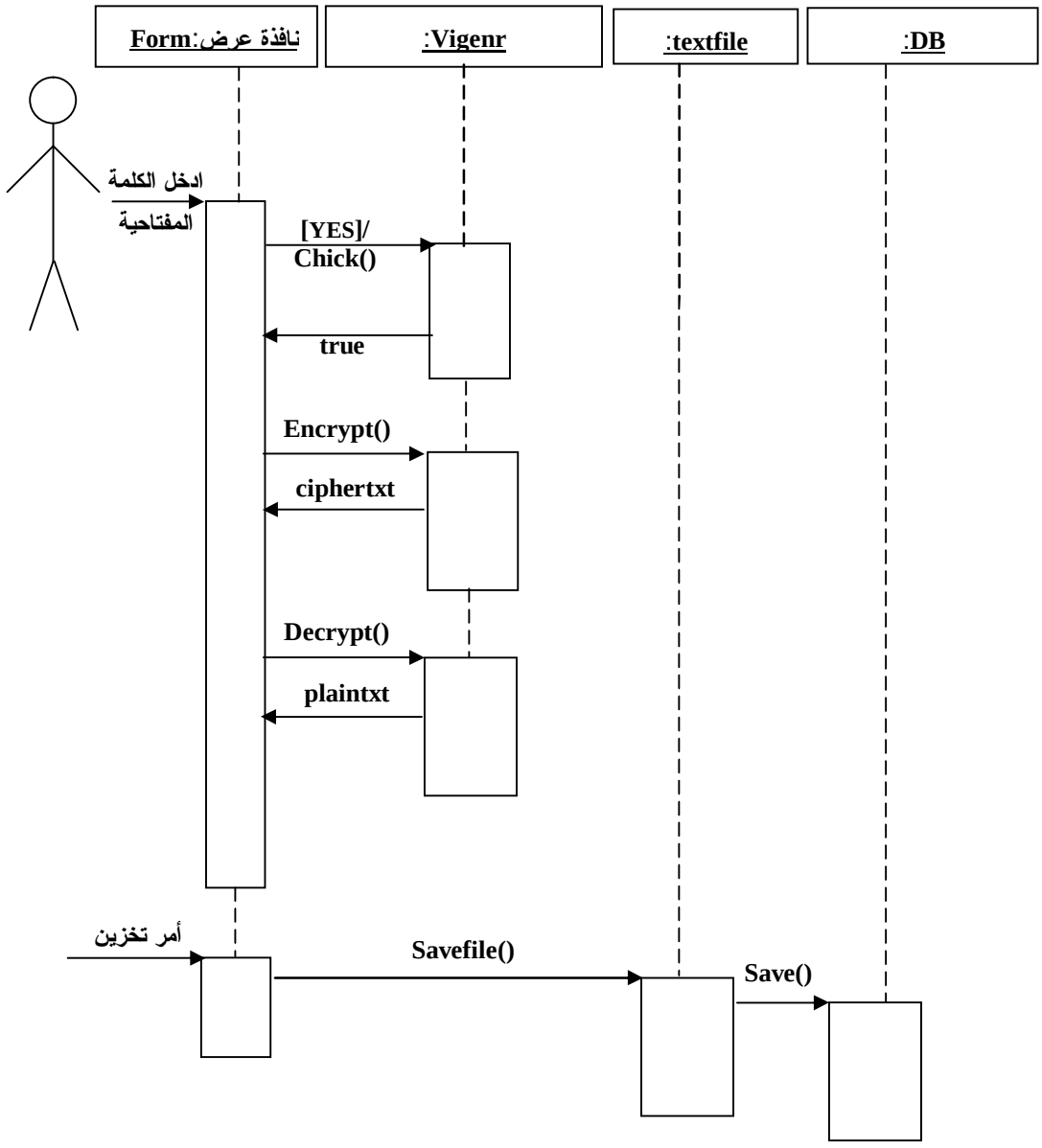
شكل (36): مخطط التتابع لخوارزمية قيصر.

- **في حالة التشفير وفك التشفير بخوارزمية Rail Fence**



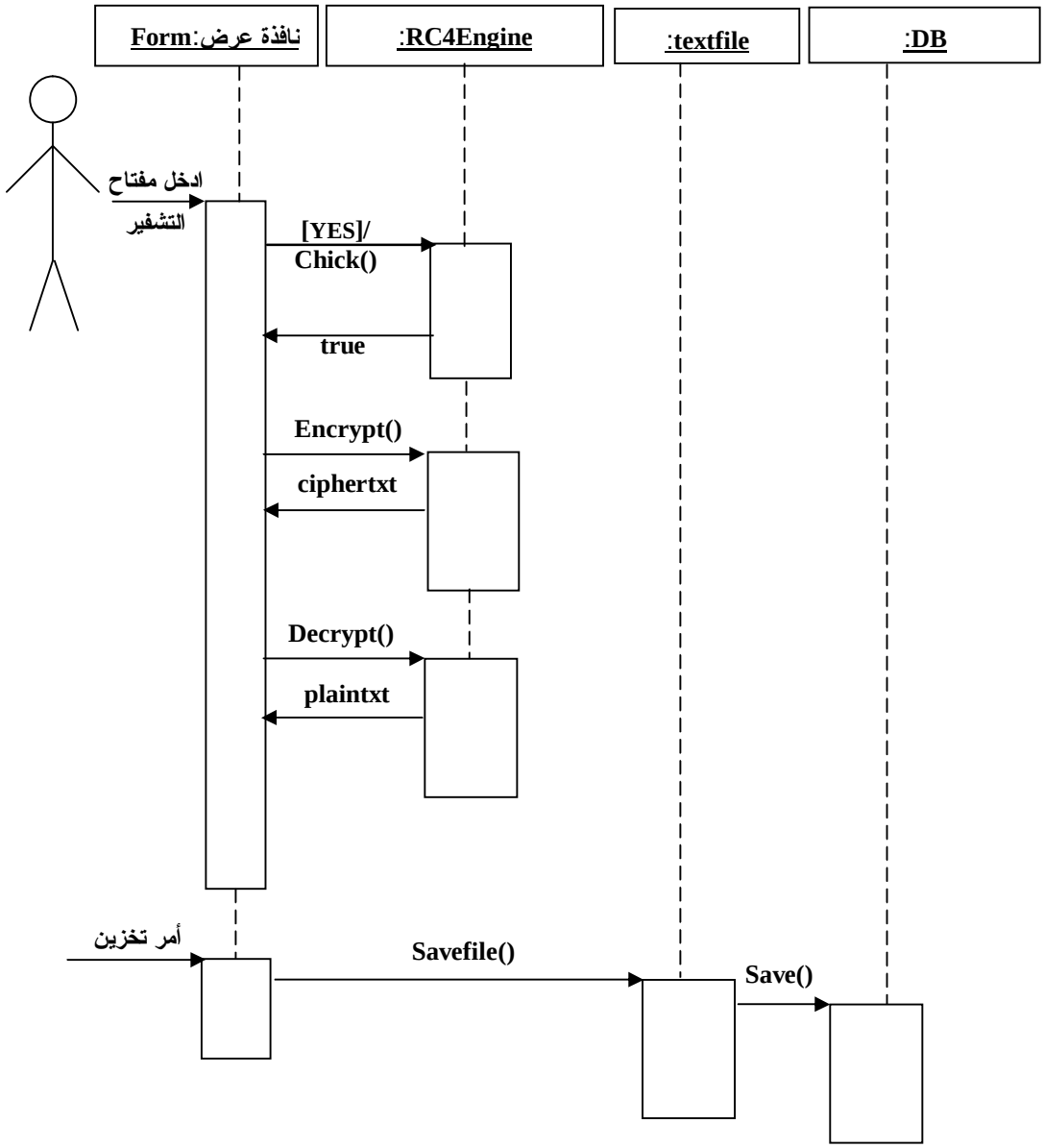
شكل(37): مخطط التتابع لخوارزمية Rail Fence.

- **في حالة التشفير وفك التشفير بخوارزمية Vigenere**



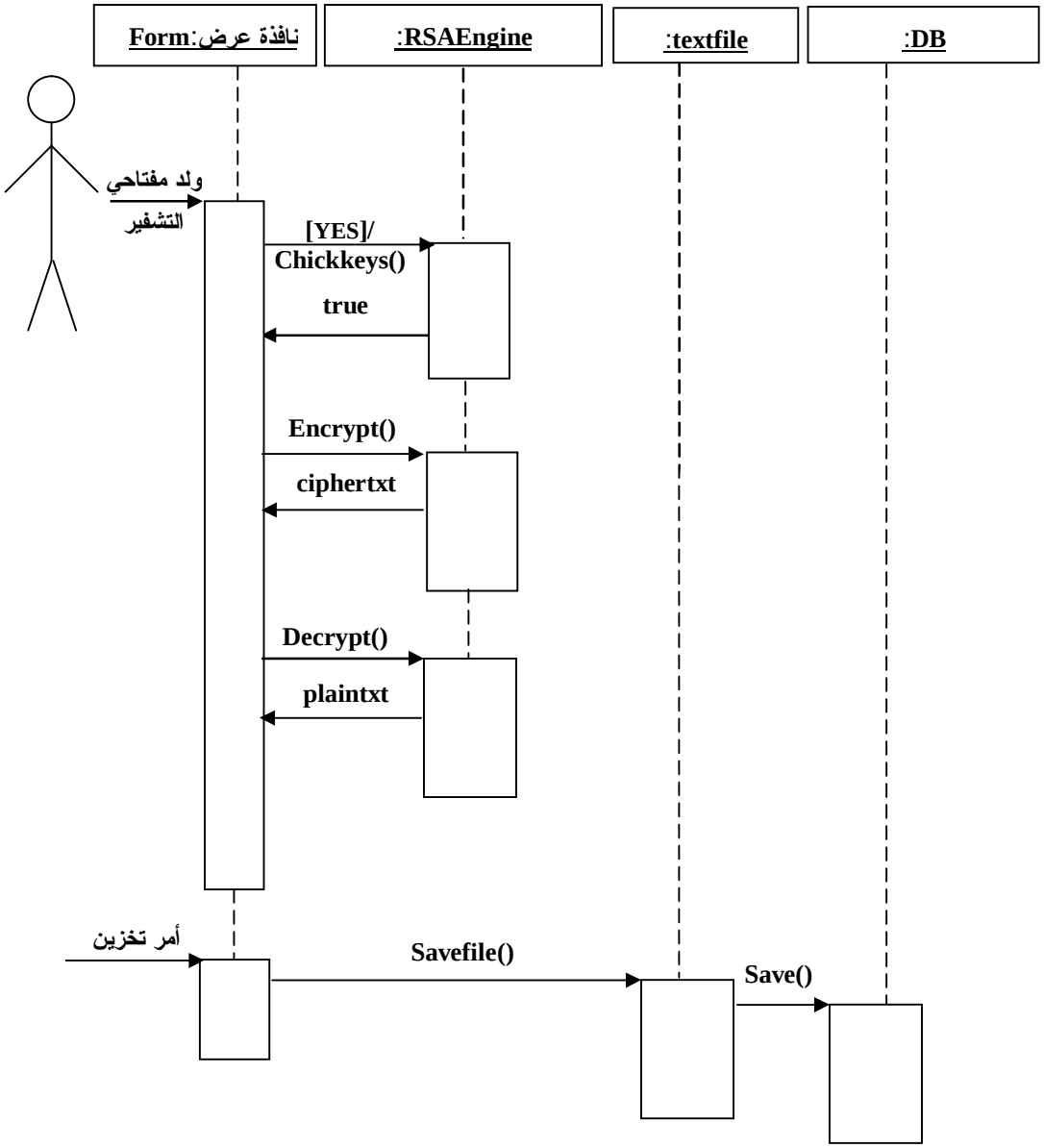
شكل(38): مخطط التتابع لخوارزمية Vigenere.

- **في حالة التشفير وفك التشفير باستخدام خوارزمية RC4**



شكل(39): مخطط التتابع لخوارزمية RC4.

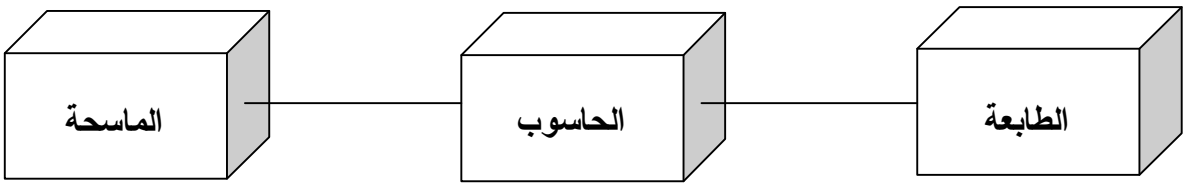
- **في حالة التشفير وفك التشفير بخوارزمية RSA**



شكل (40): مخطط التتابع للخوارزمية RSA.

4.4.7 مخطط التجهيز Deployment Diagram

مخطط التجهيز يبين المكونات المادية التي يحتاجها النظام، الشكل(41) يوضح هذا المخطط.



شكل (41): مخطط التجهيز.

الفصل الخامس

التنفيذ والتطبيق والاختبار

5.1 التنفيذ والتطبيق

وهي المرحلة النهائية حيث يتم تحويل النظام المصمم إلى نظام واقعي (نظام يعمل على الحاسوب)، وذلك بإعداد البرامج اللازمة للنظام المصمم وتحديد الملفات حسب العلاقات التي تم تصميمها مسبقا وأيضا تجهيز الشاشات المطلوبة للتعامل مع هذه الملفات وبعد ذلك يتم الشروع في كتابة البرامج.

عند بدء تشغيل المنظومة تظهر للمستخدم الواجهة الرئيسية كما هي موضحة في الشكل(39). الواجهة الرئيسية والواجهات اللاحقة بسيطة جدا لأي مستخدم. و جميع الواجهات (انظر الملحق الثاني) بها زر للمساعدة و شرح لجميع العمليات التي تقوم بها.

وسوف نتعرف في هذا الفصل على بيئة نظام التشغيل المستخدمة وكذلك لغة البرمجة التي استخدمت لعمل النظام كما سنذكر نتائج النظام وهى المرحلة النهائية.

5.1.1 متطلبات نظام التشغيل والكيان المادي للمنظومة

يمكن تشغيل هذه المنظومة على جهاز حاسوب لا تقل سرعته عن 500MHz مجهز باسطوانة صلبة لا تقل سعتها التخزينية عن 10GB مع ذاكرة بسعة لا تقل عن 128 MB مع نظام تشغيل نوافذ Windows XP و كذلك رزمة برمجيات Access 2003 أو أحدث.

5.1.2 نبذة عن بيئة التشغيل Windows XP

هو نظام تشغيل من إنتاج ميكروسوفت، ويعتبر النظام الأكثر رواجاً واستخداماً، وأهم ما يتميز به تعددية المهام حيث من الممكن تنفيذ أكثر من تطبيق بسرعة وسهولة في نفس الوقت، وهذه البيئة تعطي مميزات كثيرة للبرمجة، كما أنها تمتاز بصغر حجم الملفات التنفيذية فيها ويرجع ذلك إلى وجود مكتبات الربط الديناميكي، كذلك تتيح الاستفادة الكاملة من نظم إدارة قواعد البيانات.

5.1.3 نبذة عن اللغة المستخدمة

لقد قمنا باستخدام لغة (Visual Basic.net) في برمجة هذا النظام لما لها من مميزات عديدة يصعب حصرها، وسوف نتحدث عن بعض هذه المميزات.

من أهم مميزات لغة vb.net أن بها مكتبة خاصة للتشفير تحتوى على العديد من خوارزميات التشفير منها خوارزمية RSA.

كذلك فإن اللغة تحتوي على حوالي 3500 صنف جاهزة وقابلة للوراثة بما فيها من أدوات controls مما يمنحنا القدرة على تطويرهن بشكل مناسب، كما أصبح بالإمكان تعريف المتغيرات داخل المقاطع مثل مقاطع الجمل الشرطية أو الجمل التكرارية بحيث تكون معزولة عن المتغيرات الموجودة خارج المقطع ويتم التخلص منها بمجرد الخروج من المقطع، بالإضافة إلى أن اللغة تقوم بكتابة نهاية المقاطع تلقائياً بمجرد كتابتك لبدايته وضغط زر Enter، فمثلاً لو كتبت الجملة التالية:

```
For i as integer =0 to 100
```

فإنه بمجرد الضغط على Enter سوف يتم كتابة الجملة التالية:

```
Next
```

وكذلك تقوم اللغة بتنسيق المسافات تلقائياً بحيث يبدو البرنامج منسقاً ومنظماً وواضحاً عند قراءته، كما أن هناك تحسينات كثيرة في تلميحات الشاشة التي تعرض قيم المتغيرات وأنواعها ومعاملات الإجراءات والدوال وقيمها المعادة مع نبذة عن وظيفة كل دالة وكل معامل.

وكذلك أصبح هناك تحسينات كثيرة في مظهر النموذج والأدوات وأصبح هناك العديد من الخصائص والوسائل الجديدة التي تمت إضافتها لهذه العناصر بحيث تمنحك تحكماً أكبر فيها فمثلاً أصبح بإمكانك تحديد درجة شفافية النموذج والتحكم في تفعيل الأزرار وإلغاء تفعيلها حسب الحاجة.

5.1.4 مستوى الأمن والدخول إلى المنظومة

أثناء برمجة المنظومة وضعنا في اعتبارنا توفير مستوى من الحماية والخصوصية للمستخدم، وذلك بوضع كلمة مرور للدخول إلى المنظومة، حيث يقوم كل مستخدم بالتسجيل وذلك بإدخال البيانات التالية:

- اسم المستخدم.
- كلمة المرور.

وبعد أن يتم تخزين هذه البيانات يستطيع المستخدم الذي سبق تسجيله الدخول إلى المنظومة.

استخدام المنظومة ليس مقتصراً على شخص معين وإنما تستخدم من قبل أي شخص مستفيد منها، هذا الشخص قد يكون طالب أو مدرس أو محلل نظام... الخ، وعليه لا تحتوى هذه المنظومة على صلاحيات، أي أن الهدف من كلمة المرور ليس إعطاء صلاحيات وإنما توفير الخصوصية للمستخدم، حيث لا يمكن لأي شخص الإطلاع على الملفات التي قام المستخدم بتشفيرها أو فك تشفيرها، كذلك لن يستطيع أي شخص معرفة المفاتيح التي استعملها المستخدم.

5.2 الاختبار

5.2.1 أهداف الاختبار

في عملية الاختبار يتم تأمين المنظومة من خلال إجراء, عدة اختبارات الغرض منها:

- التأكد من صحة البيانات المدخلة.
- اكتشاف الأخطاء وتصحيحها بعد مراجعتها.
- محاولة الحصول على مخرجات النظام بصورة صحيحة ودقيقة أي منع حدوث أي خطأ يؤثر على مخرجات النظام.

5.2.2 حماية النظام

لقد تم فتح ملف خاص بالمستخدمين وهذا الملف يحتوي على اسم المستخدم وكلمة المرور بالإضافة إلى الملفات الخاصة به، حيث لا يمكن لأي مستخدم أن يستخدم المنظومة إلا بعد إدخال كلمة المرور الخاصة به، كذلك لا يمكن لأي مستخدم الإطلاع على ملفات المستخدمين الآخرين.

5.2.3 الاختبار الفعلي للنظام

يتم في هذه المرحلة إجراء اختبارات فعلية على البيانات المدخلة للنظام لمعرفة مدى فعالية العمل في النظام، مع الأخذ في الاعتبار أن تظهر المخرجات بصورة صحيحة لكي يتم التأكد من نجاح عملية الاختبار.

الخلاصة

لقد وضحنا في هذا الكتاب مفهوم التشفير وتحدثنا عن بعض خوارزميات التشفير المستخدمة حالياً والتي كانت تستخدم في السابق، مع إعطاء أمثلة عن بعض هذه الخوارزميات.

كما قمنا ببرمجة منظومة لتشفير الملفات باستخدام بعض خوارزميات التشفير، بالإضافة إلى أننا استخدمنا لغة النمذجة الموحدة UML لتحليل وتصميم هذه المنظومة، مع إعطاء نبذة عن هذه اللغة لكي يتسنى للقارئ فهمها من الناحية النظرية والتطبيقية.

ونسأل الله –عز وجل- أن يوفقنا في تقديم كتاب جيد يكون سنداً وعوناً لكل شخص يريد دراسة وفهم خوارزميات التشفير.

والسلام عليكم ورحمة الله وبركاته

المراجع

أولاً- التشفير

بعض مواقع الشبكة العالمية للمعلومات (الانترنت):

- www.Wikipedia.org
- www.bramjnet.com
- www.kremlinencrypt.com
- www.roro44.com
- www.mawsoah.net
- www.ma3refah.net
- www.alhasebat.com

ثانياً- لغة النمذجة الموحدة

1. مشروع تخرج بعنوان محرر لغة النمذجة الموحدة UML Editor للطالبين أكرم عبد الله المدني وسرمد محمد العاني.
2. مشروع تخرج بعنوان شرح لغة النمذجة الموحدة UML للطالبين ابريك عوض ابريك وصالح عقوب صالح.
3. موقع الدكتور خالد الشقروني.

الملاحق

الملحق الأول

لغة النمذجة الموحدة

UML

1.1 نشأة لغة النمذجة الموحدة

تم إنشائها من قبل شخص يدعى قرادي بوش Grady Booch مؤسس OOD وهي اختصار للعبارة Object Oriented Design، ثم في عام 1994، قام جيم رامبخ Jim Rumbaugh مؤسس OMT وهي اختصار للعبارة Object Management Group، بالانضمام إلى قرادي بوش للعمل في شركة راشيونال Rational Corp، كان الغرض من المشاركة هو دمج أفكارهما و صبها في منهجية موحدة و كان عنوان العمل لهذه المنهجية هي "المنهجية الموحدة Unified Method".

في عام 1995، انضم أيضا مبدع OOSE ايفار جاكوبسون Ivar Jacobson، إلى راشيونال، و قام بضم أفكاره خاصة مفهوم" حالات الاستخدام Use Cases" في المنهجية الموحدة التي أصبحت تدعى لغة النمذجة الموحدة، لقت هذه اللغة قبولا واسعا لدى المهتمين ببناء البرمجيات على اختلاف منهجياتهم، وقد أصبحت اللغة المعتمدة لترميز العمليات البرمجية لدى الوسط الصناعي.

لغة النمذجة الموحدة تقدم وسيلة رمزية مبسطة للتعبير عن مختلف نماذج العمل البر مجي، يسهل بواسطتها على ذوي العلاقة من محللين و مصممين و مبرمجين بل وحتى المستفيدين التخاطب فيما بينهم وتبادل المعلومات بطريقة موحدة و موجزة تغنيهم عن الوصف اللغوي المعتاد.

هنا يجب التنويه إلى نقطتين شكلتا سوء فهم ارتبط بلغة UML لدى الكثيرين:

- لغة النمذجة الموحدة ليست منهجية لبناء البرمجيات، بمعنى أنها لن ترشدك إلى أفضل الطرق لتصميم البرمجيات و تطويرها.
- لغة النمذجة الموحدة لا ترتبط بمنهجية محددة لتنشئة البرمجيات، بالرغم من أنها استلهمت رموزها من منهجيات سابقة.

1.2 تعريف لغة النمذجة الموحدة

لغة النمذجة الموحدة معناها إن هناك منهجيات توحدت داخل هذه اللغة وأنها لغة تختص بالنمذجة. وهي لغة نمذجة رسومية تقدم لنا صيغة لوصف العناصر الرئيسية للنظم البرمجية.

في السابق كان المطورون يفكرون بطريقة تسلسلية دالية Functional Orientation، أي أنهم كانوا يقسمون النظام إلى عدة إجراءات أو دوال بحيث يعمل البرنامج بطريقة تسلسلية، لذا ظهر ما يسمى بنموذج سير المعلومات DFD (Data Flow Diagram) و نموذج الكائنات والعلاقات ERD (Entity Relationship Diagram).

وبعد ظهور البرمجيات كائنية المنحنى Object Oriented(OO) اكتشف المطورين أن الطرق القديمة لا تنفع لنمذجة الأنظمة وهذا ما أدى إلى ظهور لغة النمذجة الموحدة.

1.3 لماذا ظهرت لغة النمذجة الموحدة؟

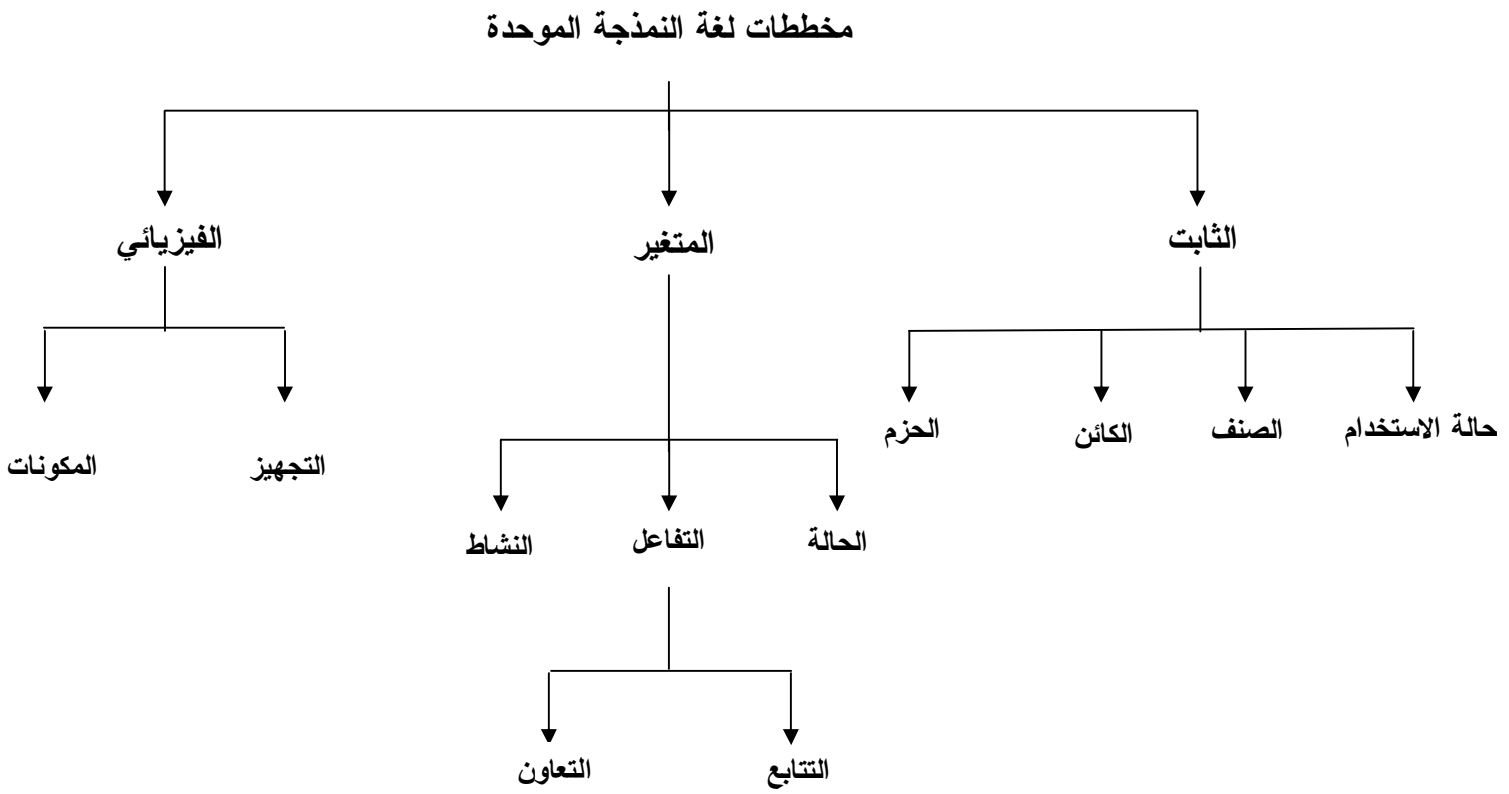
دائما هناك فراغ أو فجوة بين المستخدم والمبرمج، فالمبرمج بسبب علمه يعتقد أن الناس من حوله قريبين في مستوى معرفته، وحتى لو لم يعتقد ذلك فهو لا يقدر حجم المسافة بين الاثنين.

وهذا يعنى أن لغة النمذجة الموحدة وجدت لسببين:

- لتبين مدى فهمنا للمشروع أو النظام.
- لتمثيل النظام بلغة دقيقة تفسر تفسيرا واحدا.

1.4 تصنيف مخططات لغة النمذجة الموحدة

مخططات لغة النمذجة الموحدة موضحة في الشكل (42).



شكل(42): مخططات لغة النمذجة الموحدة.

1.4.1 المخططات الثابتة

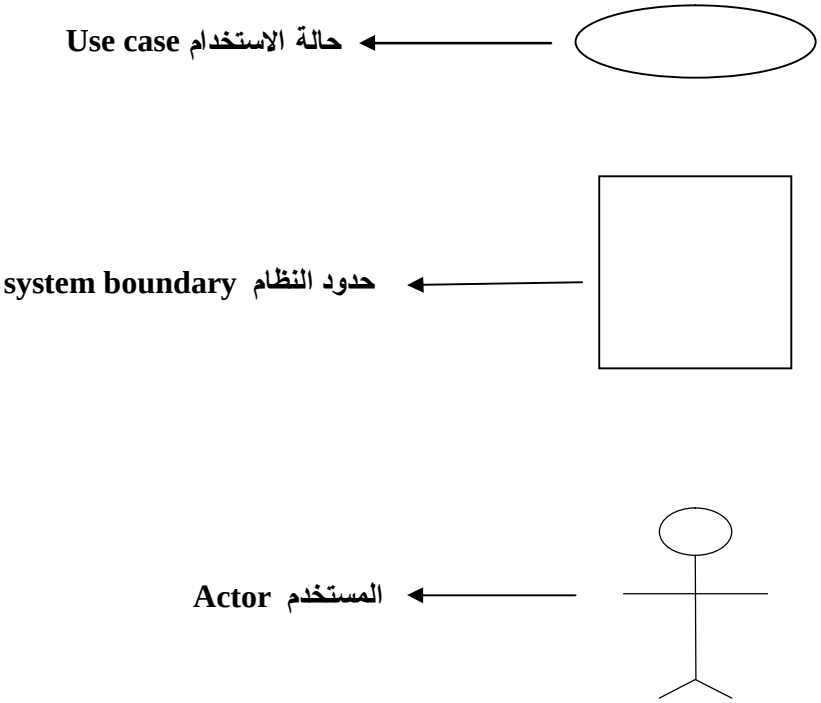
في هذه المخططات يكون عرض البيانات بصورة ثابتة، وهي تنقسم إلى أربعة مخططات تسمى مخطط حالة الاستخدام، مخطط الصنف، مخطط الكائن، مخطط الحزم.

1.4.1.1 مخطط حالة الاستخدام

حالة الاستخدام هي وصف لسلوك النظام من وجهة نظر المستخدم بحيث لا تتدخل في تفاصيل النظام فهي ذات فائدة خلال مراحل التحليل والتصميم، وتساعد في فهم المتطلبات.

المخطط يكون سهل الاستيعاب مما يمكن كلاً من المطورين والمحللين والمصممين والمبرمجين والمختبرين والمستفيدين من الاشتغال عليه.

الأدوات الخاصة بمخطط حالة الاستخدام Tools of Use Case



ACTOR هو الشخص الذي يستخدم النظام أو الشيء الذي يتعامل مع النظام.

1.4.1.2 مخطط الصنف

مخطط الصنف يستخدم في مرحلتي التحليل، التصميم، وكذلك يستخدم لرسم خريطة للمفاهيم العامة التي تمكن المستفيد من استيعابها، وتسمى النموذج المفاهيمي Conceptual Model.

النموذج المفاهيمي يعتبر أداة قوية لتحليل المتطلبات مثل مخطط حالة الاستخدام.

1.4.1.3 مخطط الكائن

الكائن هو عبارة عن متغير من نوع صنف، وكل صنف عبارة عن وصف عام لمجموعة من الكائنات، أي أن هذا النموذج هو عبارة عن مثال حقيقي لمخطط الأصناف أو هو عبارة عن صورة لحظية للنظام.

1.4.1.4 مخطط الحزم

مخطط الحزم يقوم بجمع الأشياء التي لها علاقة مع بعضها البعض في مكان واحد، كما أنه يعرفنا بمحتويات النظام حيث يتم تقسيم النظام إلى عدة أجزاء كل جزء يستخدم في شيء معين، وهذا المخطط عبارة عن نظرة علوية للنظام، وكذلك يفيدنا في عملية التطوير.

1.4.2 المخططات المتغيرة

في هذه المخططات سوف يتم توضيح تصرفات النظام، والتفاعلات التي تحدث بين الكائنات، وكذلك التفاعلات التي تحدث بين الأصناف، أي أنه في هذه المخططات يكون عرض البيانات بصورة متحركة.

وهو ينقسم إلى ثلاثة مخططات تسمى مخطط الحالة State Diagram، مخطط التفاعل Interaction Diagram (وهو ينقسم إلى مخطط التتابع Sequence Diagram ومخطط التعاون Collaboration Diagram)، مخطط النشاط Activity Diagram.

1.4.2.1 مخطط الحالة

مخطط الحالة يتعامل مع الكائنات حيث كل كائن له حالة state وكل حالة لها بداية وليس بالضرورة أن يكون لها نهاية.

1.4.2.2 مخطط التفاعل

مخطط التفاعل يوضح كيفية تفاعل الكائنات مع بعضها البعض وينقسم إلى قسمين:

أ- مخطط التتابع

مخطط التتابع يوضح كيفية تفاعل الكائنات مع بعضها البعض، وذلك بإيضاح الرسائل المتبادلة، مع التركيز على الوقت ويقصد بالوقت هنا هو وقت حدوث الحدث.

ب - مخطط التعاون

نقوم برسم مخططات التعاون لوصف الكيفية التي تتعاون بها الكائنات مع بعضها البعض. مخطط التعـــاون يشبه مخطط التتابع ولكن الفرق بينهما أن مخطط التعاون لا يركز على الوقت عكس مخطط التتابع.

1.4.2.3 مخطط النشاط

هو نموذج يركز على النشاط ونتائجه، وهو شبيه بمخطط الحالة والاختلاف بينهم هو أن مخطـــط الحالـــة يركز على الأصناف بينما مخطط النشاط يركز على النظام بأكمله.

1.4.3 المخططات الفيزيائية

في هذه المخططات يتم توضيح تفاصيل النظام، وهي تنقسم إلـــى مخططـــين تســـمى مخطـــط المكونـــات Component Diagram، مخطط التجهيز Deployment Diagram.

1.4.3.1 مخطط المكونات

يتشابه مخطط المكونات مع مخطط الحزم في أنه يركز على محتويات النظام، ولكن الاخـــتلاف بينهمـــا أن مخطط المكونات يقوم بتقسيم لنظام إلى عدة مقاطع ويوضح كيفية تفاعل هذه المقاطع مع بعضها البعض.

1.4.3.2 مخطط التجهيز

هذا المخطط يوضح ما هي المكونات المادية المطلوبة للنظام أو التي يجب توفيرها للنظام.

الملحق الثاني

واجهات النظام

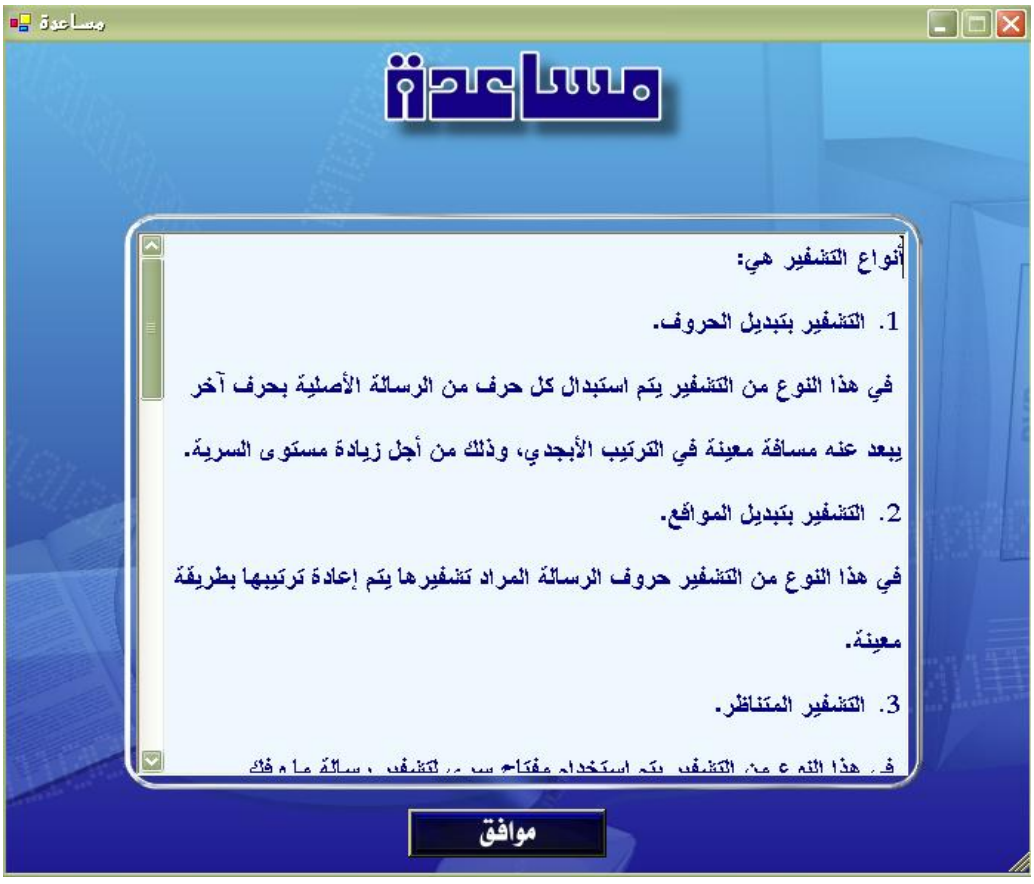
شكل(43): الواجهة الرئيسية للمنظومة.

شكل(44): الواجهة الرئيسية للمنظومة عند الضغط على زر خروج.

شكل(45): واجهة إدخال اسم المستخدم وكلمة المرور.

شكل(46): واجهة تسجيل لمستخدم جديد.

شكل(47): واجهة تغيير كلمة المرور.

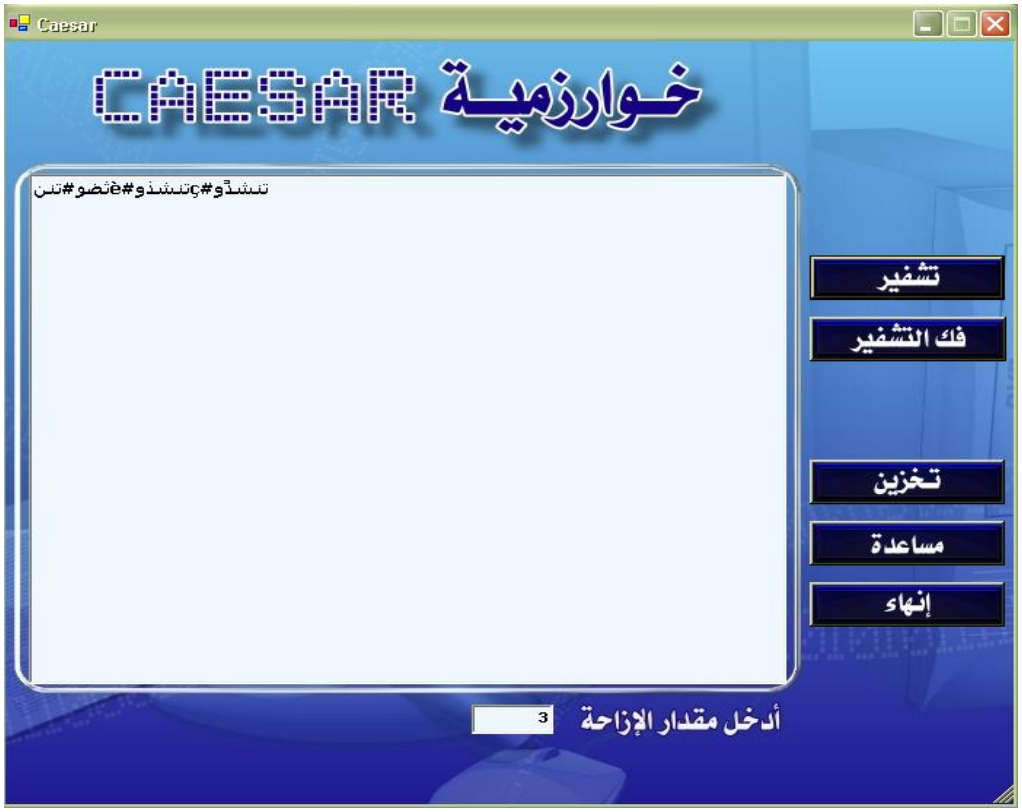


شكل(48): واجهة المساعدة.

شكل(49): واجهة كتابة نص.

شكل(50): واجهة فتح ملف.

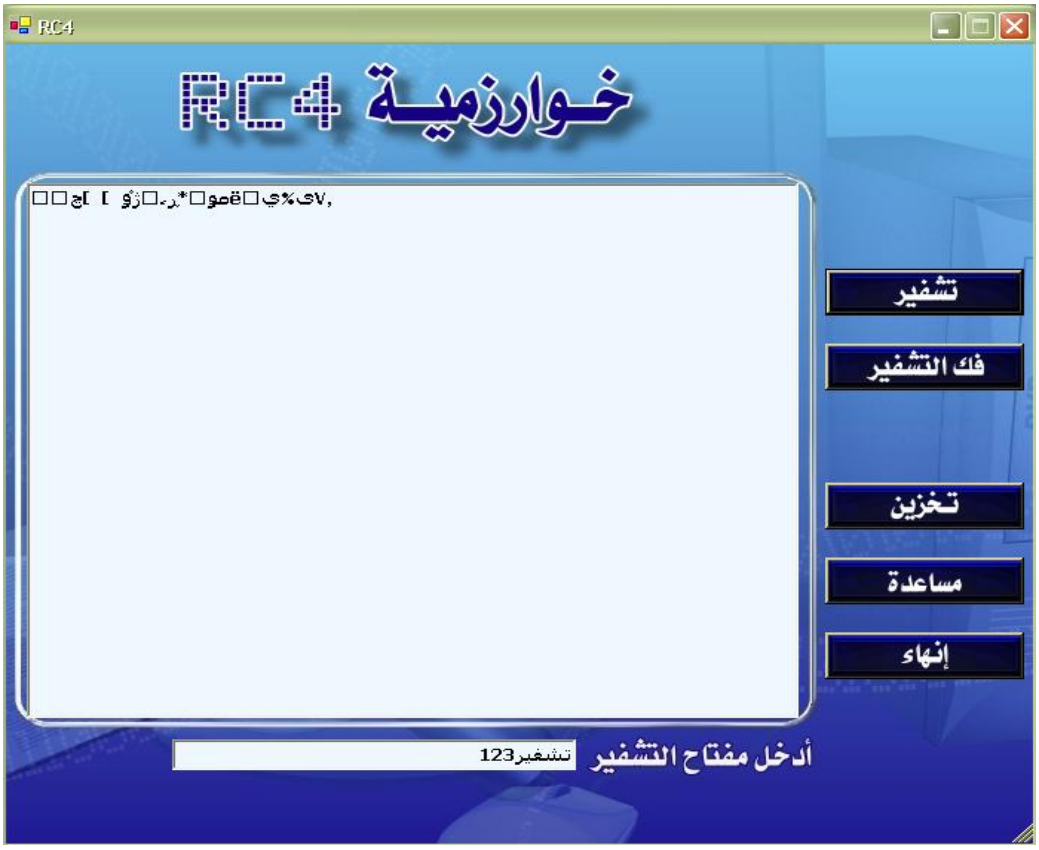
شكل(51): واجهة اختيار ملف معالج مسبقاً.



شكل(52): واجهة خوارزمية Caesar.

شكل(53): واجهة خوارزمية **Vigenere**.

شكل(54): واجهة خوارزمية **Rail Fence**.



شكل(55): واجهة خوارزمية RC4.

شكل(56): واجهة توليد المفتاحين العام والخاص لخوارزمية RSA.

شكل(57): واجهة خوارزمية RSA.