الجمهورية العربية السورية

وزارة التعليم العالي

الجامعة الافتراضية السورية

Syrian Arab Republic

Ministry of Higher Education

Syrian Virtual University

2017

دراسة أعدت لنيل درجة الماجستير في تقانات الويب

Master in Web Technologies

التشفير الهجين لرسائل شبكات التواصل الاجتماعي

Merging Cryptography for Broadcast letters by Social media

إعداد الطالب

محمود اللحام

إشراف

أ.د. محمد نور شمه

الفهرس	رقم الصفحة
اهداء	**1**
كلمة شكر	**2**
ملخص باللغة العربية	**3**
ملخص باللغة الإنكليزية	**5**
مشكلة البحث	**7**
مقدمة	**11**
الفصل الأول: التشفير التقليدي	**18-39**
التشفير التقليدي	18
قياس قوة التشفير وأمن الخوارزميات	27
تعاريف وخواص	28
ملخص الفصل الأول	39
الفصل الثاني: التشفير باستخدام متجهات تولد مصفوفات مثلثية ذات طبيعة خاصة	**40-55**
تعاريف ومبرهنات هامة	43
خوارزمية التشفير	50
خوارزمية فك التشفير	52
ملخص الفصل الثاني	55
الفصل الثالث: التشفير باستخدام المصفوفات المنتهية المجزأة	**56-72**
المصفوفة الجزئية المنتهية	56
خوارزمية التشفير بواسطة المصفوفة الجزئية المنتهية	59
ملخص الفصل الثالث	72
الفصل الرابع: مسائل وتطبيقات بلغة C#	**73-127**
مدخل إلى لغة C#	73
مدخل إلى برمجة تطبيقات الويب	97
بيئة ASP.NET	99
مكتبة ASP.NET SignalR	113
أمثلة عامة	116
المراجع	125

اهداء

اهداء

إلى عائلتي والدي **عبد الرزاق اللحام** ووالدتي **قمر الترك** وأختي **سعاد اللحام** الذين قدموا لي كافة وسائل الدعم المادي والمعنوي في كل مراحل إعداد هذا العمل....

اهداء

إلى جميع محبي العلم والتطور والتكنولوجيا

اهداء

إلى جميع الأصدقاء والأقارب المخلصين الذين يتمنون لي النجاح.

كلمة شكر

أتوجه بالشكر

إلى كل من ساهم بإنجاز هذا العمل، وأخص بالذكر المشرف العلمي الأستاذ الدكتور **محمد نور شمه** لما قدمه لي من إرشادات ومعلومات مفيدة وقيمة أفادت هذه الرسالة والارتقاء بها الى مستوى علمي عالي.

كما أتوجه بالشكر

إلى خالي الدكتور **منير الترك** لمساعدته ومتابعته لي في هذا العمل.

ملخص

التشفير الهجين لرسائل شبكات التواصل الاجتماعي

قمنا بدراسة التشفير الهجين وفق الفصول التالية:

1- **الفصل الأول:** تمت دراسة تعاريف ومبرهنة تتعلق بالتشفير بشكل عام.

2- **الفصل الثاني:** قدمنا في هذا الفصل تشفير جديد يعنمد على أشعة خاصة تولد مصفوفات مربعة من خلال المبرهنتين الجديدتين الاساسيتين (1) (2) فحصلنا على ما يلي:

- نتائج سريعة وصعبة الفك حيث تعتمد على جداء مصفوفات كبيرة الحجم وبالتالي فإن تحليل الشيفرة بهدف كسرها يتطلب زمناً طويلاً وتكلفة كبيرة.
- قابلية تطبيق هذه الطريقة التشفيرية في أهداف محدودة وصغيرة مثل الشركات المحلية، المدارس الجامعات، المعامل ذات الإنتاجية المحدودة.

3- **الفصل الثالث:** لقد قمنا في هذا الفصل بالتشفير باستخدام المصفوفات المجزأة معتمدين على المبرهنات الجديدة (1)،(2)،(3) أي تقسيم النص الأصلي إلى مجموعة نصوص واستخدمنا نمطين من التشفير الهجين .

- نمط تشفير هيل الذي يتغير بتغير مصفوفة هيل ومصفوفة الانسحاب مثال(1)،(2) أي بتغير:

$$pi, Bi : i = 1,2, \dots$$

- نمط تشفير المصفوفة الشعاعية للشعاع الخاص V الذي يتغير بتغير المصفوفة الشعاعية A_V أو B_V كما في المثاليين (3)،(4).
- يمكن سرد أمثلة كثيرة لتشفير الهجين حيث تتطبق على المصفوفات المجزأة ، فتارةً نستخدم المصفوفة الشعاعية A_V وأخرى تستخدم مصفوفة هيل ومصفوفة الانسحاب pi, Bi كما سنراه بالفصل الرابع.

4- **الفصل الرابع:** تطبيقات برمجية بلغة C# مع أمثلة توضيحية.

5- تم تطوير تطبيق ويب باستخدام(C#,ASP.NET) لشركة صغيرة مؤلفة من عدد من الموظفين حيث يتم التخاطب بينهم على شكل مجموعات أو بشكل فردي عن طريق التشفير الهجين ويتم فك التشفير وفق طريقة (hover) اللمس المباشر للرسالة.

6- الأطروحة مرفقة بـ CD يحتوي على التطبيق من اعداد الطالب محمود اللحام وعلى ملف الأطروحة (PDF+ Word).

Abstract

Merging Cryptography for Broadcast letters by Social media

We have studied the merging cryptography in the following chapters:

1- **First chapter:** we generally introduce the basic definitions and a theorem in the cryptography field.

2- **Second chapter:** we set in this chapter a new type of cryptography depending on special vectors to generate squared matrices through two new essential theorems (2) (1) and we conclude two results:

- Quick results and difficult to decrypt depends on multiplying huge matrices so analyzing the cipher to break it will take more time and high cost
- This method can be applied in small and limited targets like local companies, schools, universities and factories that have low productivity.

3- **Third chapter:** In this chapter, we have used the split matrices depending on the new theories (1), (2), (3) -divide the plain text into several texts using two models of merging cryptography.

- Hill model: where the model changes as long as the Hill matrix and the retrieve matrix change –see examples (1), (2): $pi, Bi : i = 1,2, \dots$
- Vector matrix model for special V, which changes as the vector matrix change, A_V or B_V- see examples (3), (4).
- We can give you many examples of merging cryptography for split matrices some use Vector matrix model and other use Hill model as we will see in fourth chapter .

4- **Fourth chapter:** sufficient examples by using C#.

5- We develop web application using (C#, ASP.NET) for a small organization consists of a number of employees communicating with each other in groups or by individual using merging cryptography and decrypt message by using (hover) property.

6- The dissertation has CD including the application that is developed by Mahmoud Allahham and (PDF + Word) copy of the research.

مشكلة البحث:

لقد تطورت التقنيات الحاسوبية ودخلت في مختلفة مجالات الحياة سواء العلمية منها أو الإدارية أو المالية أو العسكرية إلى ما هنالك إضافة إلى ذلك انتشرت رقعة عمل المنظومات الحاسوبية بحيث أصبحت لا تقتصر على منطقة جغرافية بل تشمل الكرة الأرضية بأسرها أدى كل ذلك إلى نشوء مخاطر حقيقية ناجمة عن محاولة الدخول غير المشروع إلى البيانات المعالجة والمخزنة في الحواسيب والمنقولة فيما بينهما وذلك بغية الحصول على هذه المعلومات لأغراض مختلفة أو محاولة تغييرها أو تدميرها.

انطلاقا من ذلك نشأت أفكار مختلفة في البداية تسعى إلى حماية هذه المعلومات وقد تطورت هذه الأفكار بشكل متسارع نتيجة لتعاظم وتسارع المخاطر لتشكل علما قائما بحد ذاته هو علم التشفير وحماية المعلومات والأمثلة التالية يمكن أن تعطينا فكرة عن عمليات انتهاك أو خرق المعلومات:

1) يقوم المستخدم A بإرسال ملف إلى المستخدم B يحوي هذا الملف معلومات هامة وحساسة لا يجوز نشرها (مثل بيان الأجور، كشف مصرفي. الخ) يستطيع المستخدم C وهو مستخدم غير مرخص له بقراءة الملف، مراقبة عملية نقل، وأخذ نسخة من الملف أثناء إرساله.
2) يقوم D مدير الشبكة بإرسال رسالة إلى الحاسب E الواقع في نطاق إدارته. تعلم هذه الرسالة الحاسب Eبضرورة تحديث ملف التحويل وذلك من أجل إضافة ملفات المستثمرين الجدد الذين يملكون حق الدخول إلى هذا الحاسب يقوم المستخدم Fباعتراض هذه الرسالة ويغير من محتواها بإضافة أو حذف بعض المداخل، ومن ثم يوجه الرسالة إلى E والذي يتلقاها كما لو أنها قادمة من D مباشرة ويقوم بناء عليها بتعديل ملف التحويل كما هو وراد في هذه الرسالة.
3) بدلاً من أن يقوم المستخدم F باعتراض الرسالة، يقوم بتشكيل رسالة تحوي المداخل التي يرغب بها ويوجهها إلى E كما لو أنها كانت قادمة من المدير Dيستقبل الحاسب Eهذه الرسالة وينفذ مضمونها بتعديل ملف التحويل كما هو وراد فيها.
4) يتم طرد موظف ما بدون إنذار. يرسل المدير الإداري رسالة إلى المستخدم لإبطال حساب هذا الموظف. لإتمام الإبطال يقوم المستخدم بإرسال رسالة إلى ملف الموظف، يمكن للموظف اعتراض هذه الرسالة وتأخيرها لوقت ما، بحيث يتاح له في هذا الوقت الحصول على معلومات هامة من المستخدم. يقوم الموظف بعد ذلك بإرسال هذه الرسالة وذلك من أجل التعتيم على العمل الذي قام به.

ويمكن تقسيم وسائل حماية المعلومات إلى فئات عامة أهمها:

1- وسائل للتحكم في الاطلاع على المعلومات مثل كلمات السر

2- وسائل للتحكم في انتقال المعلومات عبر شبكات الاتصال

3- وسائل حماية قواعد المعلومات

4- وسائل تعموية (تشفيريه) يتم بوساطتها تعمية أو تشفير المعلومات بحيث لا يستطيع أن يقرأها الا المصرح له بذلك والتي تعد من أفضل الوسائل لحماية المعلومات إذ أنها تحمي المعلومات من الكشف أو العبث حتى ولو وقعت في أيد غير مرخص لها بذلك .

وقد أدت عملية حماية المعلومات عن طريق تعميتها أو تشفيرها إلى بروز علم التشفير(التعمية) بفرعيه الأساسيين وهما:

- **الفرع الأول:** علم التعمية او التشفير وهو ما يختص بدراسة وتصميم طرائق آمنة لتغيير المعلومات من صيغتها الأصلية المقروءة إلى صيغة غير مقروءة لا يمكن إعادتها إلى أصلها إلا بوجود معلومة سرية تعرف بالمفتاح.
- **الفرع الثاني:** علم كسر التعمية (التشفير) أو استخراج المعمى وهو ما يختص بدراسة وتصميم طرائق لاستعادة النص المقروء من النص غير المقروء وبدون المفتاح.

ويمكن أن يعرف التشفير بأنه عبارة عن دراسة التقنيات الرياضية المتعلقة بعدد من مظاهر أمنية المعلومات مثل الوثوقية (Confidentiality)، تكامل البيانات (Data Integrity)، إثبات شخصية الكينونة (Entity Authentication) وإثبات شخصية مصدر البيانات (Data Origin Authentication). إن التشفير ليس عبارة عن وسيلة لتزويد أمنية المعلومات فقط وإنما عبارة عن مجموعة من التقنيات.

إن فكرة نظام التشفير (Cipher System) هي إخفاء المعلومات الموثوقة بطريقة معينة بحيث يكون معناها غير مفهوما للشخص غير المخول. وان أي شخص يعترض عبارة معينة (أي يحصل عليها) مرسلة من المشفر (Cryptographer) إلى مستقبل العبارة يسمى المعترض (Interceptor).

وقد عُرف علم التشفير أو التعمية منذ القدم، حيث استخدم في المجال الحربي والعسكري. فقد ذكر أن أول من قام بعملية التشفير للتراسل بين قطاعات الجيش هم الفراعنة. واستخدم الصينيون طرائق عديدة في علم التشفير والتعمية لنقل الرسائل أثناء الحروب. فقد كان قصدهم من استخدام التشفير هو إخفاء الشكل الحقيقي للرسائل حتى لو سقطت في يد العدو فإنه تصعب عليه فهمها. وكذلك ذكر أن العرب لهم محاولات جدية في مجال التشفير سنذكرها لاحقاً.

وأفضل طريقة استخدمت في القدم هي طريقة القيصر يوليوس وهو أحد قياصرة الروم.

أهمية البحث:

بناء على مسح إحصائي عن حوادث امن المعلومات في النظم الآلية اعد من قبل الحكومة الأمريكية وجد إن العوامل المؤثرة وتأثير كل عامل هي على النحو التالي:

أ: الأفعال غير المقصودة وتمثل 50% من نسبة الحوادث في امن المعلومات.

ب: عدم أمانة العاملين في الحاسب الآلي وتمثل 15-20% من نسبة الحوادث.

ج: الابتزاز والضغط المطلبي وتمثل 10% من الحوادث.

د: المياه والسوائل وعدم الالتزام بالمواصفات البيئية وتمثل 10%.

هـ: الاعتداء الخارجي لا يتجاوز 5%.

و: الكوارث الطبيعية والحريق ويمثل 10%.

من هذا يتضح إن العوامل الثلاثة الأولى أي تلك التي مصدرها الأفراد المتعاملين مع الحاسب الآلي قد تسببت في 80% من جملة الحوادث المهددة لأمن المعلومات في حين إن العوامل التي لا يدخل فيها العامل البشري لا تتعدى 20% وان تأثير العامل البشري الخارجي لا يتجاوز الـ 5% مما يشير إلى تركيز المشكلة في الأفراد المتعاملين مع الحاسب الآلي في عصرنا الحالي باتت الحاجة ملحة أكثر لاستخدام علم "التشفير" وذلك لارتباط العالم ببعضه عبر شبكات مفتوحة حيث يتم استخدام هذه الشبكات في نقل المعلومات إلكترونياً سواءً بين الأشخاص العاديين أو بين المنظمات الخاصة والعامة عسكرية كانت أم مدنية فلابد من طرائق تحفظ سرية المعلومات لذلك فقد بذلت الجهود الكبيرة من جميع أنحاء العالم لإيجاد الطرائق المثلى التي يمكن من خلالها تبادل البيانات مع عدم إمكانية كشف هذه البيانات.

يمكننا تلخيص أهمية هذا البحث للمجتمع بالنقاط التالية:

1- الوثوقية (Confidentiality): هي عبارة عن خدمة معينة تمنع من خلالها معرفة محتويات المعلومات عن جميع المشتركين عدا الأشخاص المخولين بامتلاك هذه المعلومات. يعتبر مفهوم الأمنية (Secrecy) مرادفا لكل من الوثوقية والخصوصية (Privacy).

2- تكامل البيانات (Data Integrity): عبارة عن خدمة موجهة لأغراض احتواء التغييرات غير المسموح بها (Unauthorized) للبيانات ولتحقيق هذا الهدف يجب أن تمتلك الإمكانية لكشف معالجة البيانات من قبل الأطراف غير المخولة. تشمل معالجة البيانات عمليات مثل الحشر (Insertion)، الحذف (Deletion) والإحلال (Substitution). يجب أن يكون مستقبل الرسالة قادرا على إثبات أن العبارة لم يتم تحويرها أثناء الإرسال وأن العدو يجب ألا يكون قادراً على إحلال عبارة كاذبة بدلاً من عبارة شرعية.

3- إثبات الشخصية (Authentication): عبارة عن خدمة أو وظيفة تتعلق بتحقيق التعريف (Identification)، هذه الوظيفة تطبق على كل من المشتركين في الاتصال (Two Parties) وعلى المعلومات أيضا حيث أن الأطراف المشتركة عند الاتصال عليها أن تعرف بعضها إلى البعض الأخر. أما ما يخص المعلومات المستلمة فيجب أن تطابق شخصياً المعلومات الأصلية التي أرسلت وكذلك تاريخ إرسال المعلومات ومحتويات المعلومات ووقت الإرسال

ومازال العمل والبحث في مجال علم التشفير مستمراً وذلك بسبب التطور السريع للحاسوب والنمو الكبير للشبكات وبخاصة الشبكة العنكبوتية العالمية (الإنترنت). في الحرب العالمية الثانية أصبح للرياضيات اهتماما كبيرا في هذا المجال، كمثال على ذلك

(Hans Rohrbach 1903-1993) في ألمانيا و (Alan Mathison Turing 1912-1954) في إنكلترا، (Adrin Albert 1905-1972.A) و (Marshall Hall b.1910) كان لهما الاهتمام المتزايد في هذا الحقل في الولايات المتحدة

هدف البحث:

إن هدف البحث هو الزيادة (maximize) إلى الحد الأقصى لعدم الترتيب في عرض المعلومات وذلك لغرض إخفاءها او وصولها إلى الشخص الصحيح معتمدين في ذلك على الرياضيات التي اصبح لها دور كبير في هذا المجال خاصة فيما يتعلق بالخوارزميات و الدوال الرياضية المصفوفية حيث إننا سنعتمد على دالة النظائر الجمعية في تطوير الطريقة الألمانية في التشفير وتعميمها لتشمل كامل نظام ASCII بالإضافة إلى استخدام البرامج الحاسوبية وذلك للوصول إلى نتائج سريعة ودقيقة مع مراعاة الأهداف العملية و الطبيعية والواجبات المناطة بنا حيث إنه يتطلب دائما تحقيق ما يلي في أي نظام تشفيري :

1. الموثوقية أو التعويل (Reliability)
2. الأمنية (Secrecy)
3. السرعة (Rapidity)
4. المرونة (Flexibility)
5. الاقتصاد (Economy)

وقبل الشروع في ثنايا هذا البحث لا بد من التوقف في محطات وتاريخ هذا العلم.

مقدمة:

الدراسة المرجعية:

علم التشفير (التعمية) عند العرب (Cryptography At The Arabs):

علم التعمية واحدٌ من علوم كثيرة تدين للعرب ولادةً ونشأةً وتطوّراً، وهو ليس كغيره من العلوم التي تَرجم العربُ بعضَ أصولها، ثم أغنَوْها وطوّروها كالرياضيات والفيزياء والفلسفة، وإنما هو علمٌ عربيُّ المولد، يعود الفضلُ إلى العرب في ابتكاره، ووضْع أسسه ومصطلحاته، وإرساء قواعده ومنهاجياته، وتطويره إلى أن بلغ مرحلةً ناضجة. وغدا ما وضعوه فيه مرجعاً قَبَسَ منه المشتغلون بالتعمية من بَعْدُ. فالعرب أوّلُ من كتب في طرائق التعمية الرئيسية التي ما انفكّ العالَم يَستخدم بعضَها حتى يومنا هذا، وهم أوّلُ مَن وضع المنهجياتِ الأساسيةَ في استخراج المعمّى، ودوّنوا فيهما مصنفاتٍ مستقلةً على غاية من الأهمية منذ القرن الثالث الهجري، فسبقوا بذلك الغربيين نحواً من سبعة قرون.

لقد اطَّلع ديفيد كَهْن الذي يُعَدّ كبيرَ مؤرخي علم التعمية على بعض ما كتبه العرب في هذا العلم، من خلال ما نقله القلقشندي في (صبح الأعشى) نقلاً عن ابن الدُّرَيْهِم، فقال في كتابه (Kahn on Codes) (ص 284): ((لقد طوّر المسلمون معرفةً نظريةً في استخراج المعمّى تنِمّ عن ممارستهم العملية لاعتراض المراسلات واستخراج تعميتها، وذلك على الرغم من تشكيك بعض الباحثين في ذلك)).

وقال في موضع آخر من الكتاب نفسِه (ص 21): ((اطّلعتُ على مقالٍ نُشر في مجلة الدراسات الساميّة... بيّن المقال أن العربَ مارسوا استخراج المعمّى قبل الغرب بزمن طويل. ووفّر لي المقال ما أعُدُّه أكبرَ فتح تاريخي في كتابي كله)). وأضاف في (ص 41) من الكتاب نفسِه: ((كانت طريقةُ التعمية التي استعملها قيصرُ كافيةً لعصره، لأن أوائلَ مستخرجي التعمية لم يَظهروا إلا بعد عدة قرون منه. فالعربُ هم الذين اكتشفوا مبادئَ استخراج المعمّى، إلا أن معرفتهم تقلّصت مع أُفول حضارتهم، ولم يكتشف الغربُ استخراجَ المعمّى من جديد إلا في عصر النهضة)). وقال أيضاً في كتابه (The Code Breakers) (ص 93): ((وُلِدَ علم التعمية بشِقَّيْه بين العرب، فقد كانوا أوّل من اكتشفَ طرائق استخراج المعمّى ودوّنها)). هذا ويمكن أن نوجز السَّبْقَ الذي حقّقه أسلافُنا العربُ في هذا العلم بما يلي:

1-يعقوب بن إسحاق الكندي (260 هـ / 873 م) هو أول مَن:

- كَتَبَ مخطوطةً في استخراج المعمّى، وذلك في القرن الثالث الهجري / التاسع الميلادي، أي قبل نحو ستة قرون من الإيطالي ألبيرتي (Alberti) الذي يَصف مؤرّخو الغرب كتابَه بأنه أول مخطوطة في هذا العلم .
- فرّق بوضوح بين طريقتَي التعمية الأساسيتَين: الإبدال والقلب، أي قبل بورتا (Porta) بنحو سبعة قرون.
- استخدم فكرة الكلمة المحتمَله التي تُنسَب خطأً إلى بورتا (Porta).

2-علي بنُ عَدْلان (666هـ / 1268م): هو أوّل مَن استخدم النصوص بدون فاصل، وأسماها المُدْمَج، وذلك قبل بورتا (Porta) بثلاثة قرون.

3-علي بنُ الدُّرَيْهِم (762 هـ / 1359م) هو أول مَنْ:

- اخترع فكرة الجدول المنسوب إلى فيجنيير (Vigenere) المشهور، وذلك قبله بقرنين.
- عرض طريقة التعمية باستعمال شبكة بسيطة، وذلك قبل كاردانو (Cardano) بقرنين أيضاً

ملاحظة: أصبح مصطلح التشفير هو المصطلح السائد حديثاً بدلا من التعمية، الأمر الذي يدفعنا لاستخدامه في أطروحتنا من الآن فصاعداً.

التشفير (التعمية)في العصر الحديث (Cryptography In The Modern Era):[2]

في العصر الحديث كانت للسرية دور مركزي ,لكن منذ الحرب العالمية الأولى , حصل في علم التشفير تطورات هامة وظهرت مطبوعات عنها بانتظام تقريبا , وتقدمت بالطريقة نفسها التي تقدمت بها التوجهات الاختصاصية الاخرى , وفي عام 1918 ظهرت واحدة من أشد مقالات تحليل وفك الشيفرة أثرا في القرن العشرين , وهي مقالة William F.Friedman وعنوانها

The Index of Coincidence and its Applications in Cryptography, والتي مثلت تقريرا عن بحث أجري في المختبرات الخاصة .Riverbank Laboratories، وقد نشر المقالة مع أن البحث كان قد أجري بوصفه جزءا من المجهود الحربي. وفي العام نفسه , سجل Edward H.Hebern من كاليفورنيا , أول اختراع لآلة قلب دوار , وهي الالة التي بقيت أنجح الآلات في التعمية العسكرية مدة 50 سنة تقريبا .

إلا أن الأمور بدأت تتغير بعد الحرب العالمية الاولى فهيئات الجيش والأسطول الامريكية , التي كانت تعمل بسرية تامة , بدأت بإحراز تقدم جوهري في التعمية ,وأثناء ثلاثينيات وأربعينيات القرن العشرين ظهرت بضع مقالات أساسية في بعض المنشورات العلنية , ونشرت عدة مؤلفات عن التعمية , لكنها كانت متخلفة جدا عن اخر ما توصل اليه في الموضوع , وبحلول نهاية الحرب العالمية الثانية تلاشت المؤلفات التعموية العلنية باستثناء عمل مرموق واحد هو مقالة CLAUDE SHANNON بعنوان :

"The Communication Theory of Secrecy Systems" أي "نظرية الاتصالات لنظم السرية" التي ظهرت عام 1994, وكانت هذه المقالة مشابهة لمقالة Friedman التي ظهرت عام 1918, وقد نزعت عنها صفة السرية بعد انتهاء الحرب العالمية الثانية.

وبقيت المؤلفات التعموية من عام 1949 وحتى عام 1967 شحيحة، الى أن ظهر في عام 1967 نوع مختلف من الاسهام في الموضوع , هو كتاب [6]David Kahnالتاريخي .The Code Breakers لم

يتضمن الكتاب أي أفكار تقنية جديدة، لكنه احتوى تاريخا كاملا إلى حد بعيد لما حصل في الماضي، ومنه بعض المقالات التي كانت لا تزال الحكومة الأمريكية تعتبرها سرية. وتكمن أهمية الكتاب في منظوره الواسع وفي الرواج الذي لاقاه وجعل الناس ممن لم يعر التعمية أي اهتمام من قبل يدركون أهميتها. ومن ثم بدأت شذرات من مقالات التشفير الجديدة بالظهور.

وفي الوقت نفسه تقريبا، قام Horst Feistel –الذي عمل بنظم تمييز العدو من الصديق في القوى الجوية الامريكية – بنقل علم التعمية الى مختبر IBM Watson وبدأ هناك بتطوير ما سمي لاحقا مقياس تشفير المعطيات الامريكي , وفي بداية سبعينيات القرن العشرين , نشرت شركة IBM عدة تقارير علمية عن الموضوع أعدها Feistel و زملاؤه .

وقد بات ملحاً موضوع ربط التشفير بالمصفوفات المجزأة وهذا ماقام به الباحثان د.شمه ود.الخطيب [4] عام 2015. والذي وضعنا على خطا استخدام المصفوفات المجزأة وتطويرها باستخدام التشفير الشعاعي.

كما أن أهمية تشفير هيل ومصفوفة هل دفعنا للبحث عن نمط جديد من مصفوفات هل بحيث يتم توليده بعيداً عن التوليد العشوائي (Random Hill Matrix) عن طريق المصفوفة الشعاعية [5] عام 2017 ثم استثماره في اطروحتنا هذه.

هناك العددي من الأعمال لتوليد مصفوفة هل نذكر منها [2] , [3] 2016 المصفوفة الفيثاغورثية

التطورات المتلاحقة لعلم التشفير:

اقتصر استخدام علم التشفير على عدد قليل من الدول وخصوصا الغربية منها وتميزت الفترة ما قبل 1976م بالسرية التامة حول التطورات في هذا العلم وتركزت معظم أساليب في هذه الفترة على ما يسمى بنظم المفتاح الواحد أي عملية التعمية وحلها تستخدم نفس المفتاح ويعتبر عام 1976م عام ميلاد علم التعمية الحديث , فقد اقترح BRUCE SCHNEIER و Martin Hellman التعمية بالمفتاح العلني .حيث اكتشفت نظم جديدة للتشفير عرفت باسم نظم التشفير المشاعة حيث تتم عملية التشفير بواسطة مفتاح معلن غير سري أما عملية فك التشفير فتتم بواسطة مفتاح مختلف وسري .

<u>الترميز (Coding):</u>

مع سرعة تطور الإنترنت وسرعة انتشاره ووصوله تقريباً إلى كل بلد وإلى كل منزل، بدأت الحاجة إلى تنظيمه وعولمته ليصل إلى هؤلاء المستخدمين بلغاتهم، وهنا ظهرت هذه المنظمات التي ترعى تطور هذه الشبكة العالمية وضمان وصولها بالشكل السليم إلى هؤلاء المستخدمين باللغة التي يفهمونها، وهنا ظهرت أنظمة خاصة بالترميز فمن هذه الأنظمة:

[11] Unicode: هي مجموعة رموز عالمية تستخدم لتعريف جميع الرموز والحروف المستخدمة في أغلب لغات العالم وتجميعها في ترميز واحد لتسهيل عرض وإرسال المعلومات بغض النظر عن اللغة المستخدمة.
هذا الترميز العالمي يستخدم من 1 إلى 4 بايت (البايت=8 بت) لترميز الحروف، ولم يستخدم حتى هذه اللحظة سوى ثلث العدد المتاح في Unicode لترميز حروف هذه اللغات. هناك ثلاثة أنواع رئيسية تستخدم حاليا لترميز Unicode:

UTF-8

وهو المفضل لدى مبرمجي الويب، حيث يستخدم 1 بايت إذا كانت الرموز موجودة في ترميز ASCII، وتستخدم 2 إلى 4 بايت للرموز المعقدة.

UTF-16

هذا الترميز يستخدم إما 2 بايت للترميز إذا كانت الرموز موجودة في (Basic Multilingual Plane BMP) و 4 بايت للرموز غير الموجودة.

UTF-32

هذا الترميز يستخدم 4-بايت على الدوام.

ASCII: هي اختصار لجملة American Standard Code for Information Interchange, وهي مختصرة في الحروف ASCII ، تلفظ عادة آسكي ، هي مجموعة رموز ونظام ترميز مبني على الألف باء اللاتينية بالشكل الذي تستخدم به في الإنجليزية الحديثة ولغات غرب أوروبية أخرى. من أكثر الاستخدامات شيوعا للنصوص المكتوبة باللآسكي تشتمل على استخدامها في أنظمة الحاسوب، كما تستخدم في أجهزة الاتصالات وأنظمة التحكم التي تتعامل مع نصوص.

DBCS :اختصار للجملة Double Byte Character Set و هو (مثل Unicode)، جدول محارف عالمي.
اعتمد فيه على 2 بايت لبعض المحارف وبايت واحد فقط لبعضها الآخر (مما سبب بعض المشاكل).

BCD : نظام التشفير الثنائي العشري وهي اختصار لجملة Binary Coded Decimal وهو يمثل فقط ب 0 و 1 , وكل 1 بت في النظام العشري يقابله 4 بت في النظام الثنائي , وهكذا

EBCDIC: وهو نوع من أنواع تشفير BCD والذي يعني: Extended Binary Coded Decimal " Interchange Code"

"النظام الثنائي العشري الموسع". وهو مجموعة أحرف مرمزة وضعت لتمثيل البيانات النصية من ترميز شركة "أي بي ام " " IBM " يستخدم 8 بت للمحرف وذلك لتمثيل 256محرف ممكن في النظام الثنائي .

وهو يستخدم أساسا في حاسبات " أي بي ام " الكبرى" IBM mainframes "

الحروف ومقابلاتها العددية:

فيما يلي جدول حصلنا عليه باستخدام برنامج *MATHEMATICA* يوضح المقابل العددي لبعض حروف *UNICODE* (UTF-8) المتوافقة مع ASCII.

Decimals	Unicode	Decimals	Unicode	Decimals	Unicode	Decimals	Unicode
34	"	63	?	92	\	121	Y
35	#	64	@	93	]	122	Z
36	$	65	A	94	^	123	{
37	%	66	B	95	_	124	|
38	&	67	C	96	`	125	}
39	'	68	D	97	a	126	~
40	(	69	E	98	B	127	•
41	)	70	F	99	C	128	€
42	*	71	G	100	D	129	پ
43	+	72	H	101	E	130	‚
44	,	73	I	102	F	131	ƒ
45	-	74	J	103	G	132	„
46	.	75	K	104	H	133	…
47	/	76	L	105	I	134	†
49	1	78	N	107	K	136	ˆ
50	2	79	O	108	L	137	‰
51	3	80	P	109	m	138	ٹ
52	4	81	Q	110	N	139	‹

53	5	82	R	111	O	140	Œ
54	6	83	S	112	P	141	چ
55	7	84	T	113	Q	142	ژ
56	8	85	U	114	R	143	ڈ
57	9	86	V	115	S	144	گ
58	:	87	W	116	T	145	‘
59	;	88	X	117	U	146	’
60	<	89	Y	118	V	147	“
61	=	90	Z	119	w	148	”
62	>	91	[	120	X	149	•
150	–	177	±	204	ج	231	Ç
151	—	178	²	205	ح	232	È
152	ک	179	³	206	خ	233	É
153	™	180	´	207	د	234	Ê
154	ڑ	181	µ	208	ذ	235	Ë
155	›	182	¶	209	ر	236	ى
156	œ	183	·	210	ز	237	ي
157		184	¸	211	س	238	Î
158		185	¹	212	ش	239	Ï
159	ں	186	؛	213	ص	240	ً
160		187	»	214	ض	241	ٌ
161	،	188	¼	215	×	242	ٍ
162	¢	189	½	216	ط	243	َ
163	£	190	¾	217	ظ	244	Ô
164	¤	191	؟	218	ع	245	ُ
165	¥	192	٥	219	غ	246	ِ
166	¦	193	ء	220	.	247	÷

ّ	248	ف	221	آ	194	§	167
	249	ق	222	أ	195	¨	168
ْ	250	ك	223	ؤ	196	©	169
Û	251	à	224	إ	197	ª	170
Ü	252	ل	225	ئ	198	«	171

إن وجود رموز شاغرة في نظام UNICODE سببها أحد أمرين:

- أوامر تخص الشركة المصنعة وهي أوامر داخلية
- إتاحة الفرصة لإدخال محارف وحروف جديدة تخص البلدان المستوردة.

فالأحرف العربية بتنوعها العريض والمائل ... الخ، تختلف عن الايطالية مثلا، وهكذا ...

الفصل الأول

التشفير التقليدي

الرسالة والتشفير (cryptography and letter):

إن كلمة Cryptography مأخوذة من اللغة اليونانية وهي عبارة عن دمج لكلمتين kryptus (hidden) وتعني السري أو المخفي وكلمة graphein(to write) وتعني الكتابة ، أي تعني الكتابة السرية أو المخفية ، استخدم الإنسان الكتابة المخفية منذ القدم حيث تذكر المصادر التاريخية أن استخدام الكتابة المخفية يرجع إلى 1900 قبل الميلاد في مصر.

تسمى عملية تحويل المعلومة من الشكل المقروء أو الطبيعي (PlainText) إلى الشكل المشفر أو المخفي (Cipher Text) بالتعمية أو التشفير (Encryption)كما تسمى العملية المعاكسة بفك التعمية أو فك التشفير Decryption

صورة1: طريقة عمل التشفير

أنواع التشفير:

حالياً يوجد نوعان من التشفير وهما كالتالي: .

1. التشفير التقليدي. (Conventional Cryptography).
2. تشفير المفتاح العام. (Public Key Cryptography).

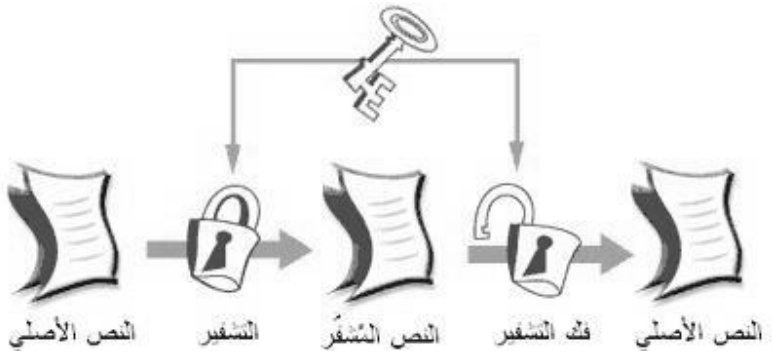
1. **التشفير التقليدي أو التقليدي** :(Conventional Cryptography):

الطرق التقليدية هي الطرق التي استخدمت فيما مضى قبل اختراع الحواسيب وبقيت الاساس لكثير من الخوارزميات الحديثة التي تستخدم اليوم حيث إنها كانت تعتمد على ابدال او احلال حرف مكان أخر والخوارزميات الجيدة كانت تقوم بالاثنين لكن جميع هذه الخوارزميات تعتمد على الحروف فقط بعكس الخوارزميات الحديثة التي تعتمد على التعامل مع البت (0 او 1) وتقسم هذه الطرق الى قسمين رئيسين:

1) **شفرات الاحلال (substitution cipher)** :في هذا النوع من التشفير يكون عن طريق احلال حرف من النص الاصلي بحرف اخر ليكون هو الحرف المشفر عملية الاحلال هذه تكون عن طريق جمع مفتاح ما الى الحرف من النص الاصلي ليكون الناتج هو الحرف المشفر مثل (تشفير قيصر) .

2) **شفرات الابدال(Transposition cipher)** : في هذا النوع التشفير يكون عن طريق تغيير أماكن حروف النص الاصلي أي مجرد تبديل في المواقع

رسم توضيحي صورة (2).



سنتحدث في البداية عن النظام التشفيري (cryptosystem) و هو الخماسية (P,C,K,E,D) حيث تعرف هذه الخماسية كما يلي:

P هي مجموعة منتهية من النصوص الواضحة المحتملة.

C هي مجموعة منتهية من النصوص المشفرة المحتملة.

K هي الفضاء الرئيسي, هي مجموعة منتهية من المفاتيح الممكنة.

E وهي قاعدة التشفير.

D و هي قاعدة فك التشفير.

$\forall k \in K, \exists e_k \in E\left(\text{قاعدة تشفير}\right), \exists d_k \in D\left(\text{قاعدة فك تشفير}\right)$

حيث $e_k: P \longrightarrow C$, $d_k: C \longrightarrow P$.

و $\forall x \in P, d_k(e_k(x)) = x$

من المعلومات المنطقية أو البديهية (Premise) في التشفير أنه المجموعات:

M,C,K,$\{E_e$:eEK$\}$, $\{D_d$:dEk$\}$ تعتبر معلنة للجميع . إن الوسيلة الوحيدة لتأمين الأمنية هو بأن نحافظ على سرية كل من (e,d). وقد اثبتت التجارب أن المحافظة على سرية خوارزميات التشفير هو في الواقع عملية صعبة جداً.

سنبدأ الأن بشفرات الاحلال وتقسم إلى اربعة اقسام رئيسية:

النوع الأول: Mono alphabetic substitution cipher

النوع الثاني: poly alphabetic substitution cipher

النوع الثالث: Polygram substitution cipher

النوع الرابع: Homophonic substitution cipher

لنبدأ **بالنوع الأول شيفرات Mono alphabetic substitution cipher:** وهي من أضعف أنواع التشفيرات ويمكن كسرها بسهولة عن طريق التحليل الإحصائي

وفي هذا النوع من الشيفرات نقوم بإحلال حرف من النص الاصلي بحرف أخر ومن أشهر شيفرات هذا النوع:

Rot13 cipher − Atbash cipher − Affine cipher − Caesar cipher

شيفرة قيصر (Caesar Cipher):

وهي طريقة قديمة ابتكرها القيصر يوليوس لعمل الرسائل المشفرة بين قطاعات الجيش وقد أثبتت فاعليتها في عصره. ولكن في عصرنا الحديث ومع تطور الكمبيوتر لا يمكن استخدام هذه الطريقة وذلك لسرعة كشف محتوى الرسائل المشفرة بها.

مثال (1):

المثال التالي يوضح طريقة عمل شيفرة قيصر: إذا شفرنا كلمة "SECRET" واستخدمنا قيمة المفتاح 3، فإننا نقوم بتغيير مواضع الحروف ابتداءً من الحرف الثالث وهو الحرف "D"، وعليه بما أن ترتيب الحروف سوف يكون على الشكل التالي:

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

فسوف تصبح بعد استخدام القيمة الجديدة لها من المفتاح "3" على الشكل الحالي:

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

الآن قيمة الـ A → D، B → E ، C → F، وهكذا...

بهذا الشكل فان كلمة "SECRET" سوف تكون "VHFUHW". لتعطي أي شخص آخر إمكانية قراءة رسالتك المشفرة؛ يجب أن ترسل له قيمة المفتاح "3".

شيفرة أفاين: (AFFINE Cipher):

ليكن $P = C = Z_{26}$

$K = \{(a, b) \in Z_{26} \times Z_{26} \backslash gcd\,(a, 26) = 1\}$

$\forall x \in P\ , \forall y \in C\ , \forall k \in K$

فان:

$$e_k(x) = mx + k(mod26)$$

و :

$$d_k(y) = m^{-1}(y - k)(mod26)$$

حيث الأحرف ومقابلاتها العددية كما يلي

A	B	C	D	E	F	G	H	I	G	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

مثال (2):

نريد تشفير العبارة : *War lost*

والمفتاح *k=10* ومعامل الضرب *m=7*

الحل : قبل أن نبدأ في التشفير يجب ان نتأكد من أن هناك معكوس ل *m=7* حتى يمكن فك التشفير

نأخذ القاسم المشترك الأعظم ل *n=26,m* فنجد أن ***gcd(7,26)=1***

الأن بعد تحويل النص الأصلي إلى أرقام يكون بهذا الشكل:

19 18 14 11 17 0 22

وبتطبيق القانون $e_k(x) = mx + k(mod26)$ نحصل على:

C1 = 7 * 22 + 10 MOD 26= 8

C2 = 7 * 0 + 10 MOD 26 = 10

C3 = 7 * 17 + 10 MOD 26 = 25

C4 = 7 * 11 + 10 MOD 26= 9

C5 = 7 * 14 + 10 MOD 26= 4

C6 = 7 * 18 + 10 MOD 26= 6

C7 = 7 * 19+ 10 MOD 26= 13

والنتيجة بعد التشفير هي : 8 10 25 9 4 6 13

وبتحويلها إلى حروف نحصل على النص المشفر وهو *IKZJE GN*

فك التشفير:

الآن لفك التشفير نطبق القانون : $d_k(y) = m^{-1}(y-k)(mod26)$

حيث أننا نحتاج : $m^{-1} = 7^{-1} = ?$ لذلك نتبع خوارزمية اقليدس كما يلي :

26=3*7+5

7=1*5+2

5=2*2+1

2=1*2+0

والأن نعود بالعكس :

1=5−2*2

=5−2(7−5)

=3*5−2*7

=3(26−3*7)−2*7

=3*26−11*7

1=3*26+(−11)*7

وبأخذ مقاس الطرفين :

1=3*26+(−11)*7=0+(−11)*7= (−11)*7=1 mod26

$$m^{-1} = -11 = 15 \bmod 26 \rightarrow m^{-1} = 15$$

وبالتالي :

P1 = 15 * (8 – 10) MOD 26 = 22

P2 = 15 * (10 – 10) MOD 26=0

P3= 15 * (25 – 10) MOD 26 = 17

P4 = 15 * (9 – 10) MOD 26 = 11

P5 = 15 * (4 – 10) MOD 26 = 14

P6 = 15 * (6 – 10) MOD 26 = 18

P7= 15 * (13 – 10) MOD 26 = 19

ومنه النص الأصلي الرقمي هو:

19 18 14 11 17 0 22

وبتحويله إلى حروف نحصل على النص War lost وهو النص الأصلي المطلوب

شيفرة أتباش (Atbash cipher) :

وهي من ابسط أنواع الشيفرات وكانت في الأصل للغة العبرية وطريقتها بأن نجعل الحرف الأول في اللغة مكان الحرف الأخير والحرف الثاني ما قبل الأخير وهكذا ...

Plain: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Cipher: ZYXWVUTSRQPONMLKJIHGFEDCBA

ويمكن تمثيلها رياضيا بالدالة التالية :

$$f_1(x) = (n - x) mod(n)$$

مثلا لتشفير الكلمة money يصبح لدينا الناتج : nlmvb

Poly alphabet:

شيفرة هل:

سميت بهذا الاسم نسبة الى مخترعها Laster S Hill وهي تعتمد في عملها على الجبر الخطي خاصة فيما يتعلق بالتعامل مع المصفوفات

تعريف (مصفوفة هل):

هي مصفوفة عددية مربعة من المرتبة k ($k = 1,2,3..$) نرمز لها A_k وتحقق ما يلي : $|A_k| \in M_n$ أي أن : $\gcd(|A_k| , n) = 1$

تعريف دالة هيل: وهي دالة من الشكل: $f(X) = (AX) mod(n)$ حيث ان المصفوفة A هي مصفوفة هيل بينما المصفوفة X هي مصفوفة النص الأصلي .

مثال (3):

ليكن النص الأصلي هو ACT ولتكن مصفوفة هيل هي:

$$A = \begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}$$

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix} * \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} = \begin{bmatrix} 67 \\ 222 \\ 319 \end{bmatrix} mod(26) = \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix}$$

ومنه فإن النص المشفر هو POH.

مثال (4): أوجد مصفوفة هل (المفتاح) إذا علمت ان النص الاصلي الرقمية هي:

$$X = \begin{bmatrix} 8 & 0 & 26 \\ 5 & 8 & 5 \\ 25 & 1 & 13 \end{bmatrix}$$ وان النص المشفر رقمياً هو: $$\begin{bmatrix} 18 & 16 & 36 \\ 13 & 8 & 31 \\ 25 & 1 & 13 \end{bmatrix}$$

الحل:

لدينا: $$X = \begin{bmatrix} 8 & 0 & 26 \\ 5 & 8 & 5 \\ 25 & 1 & 13 \end{bmatrix}$$ ولدينا أيضاً: $$Y = AX = \begin{bmatrix} 18 & 16 & 36 \\ 13 & 8 & 31 \\ 25 & 1 & 13 \end{bmatrix}$$

ومنه سوف نقوم بإيجاد مصفوفة هل اي مصفوفة المفتاح بالاعتماد على فكرة إيجاد مصفوفة الانتقال وذلك كما يلي:

$$\forall(x, y, z) \Rightarrow (x, y, z) = \alpha a_1 + \beta a_2 + \gamma a_3$$
$$\Rightarrow (x, y, z) = \alpha(8,0,26) + \beta(5,8,5) + \gamma(25,1,13)$$
$$\Rightarrow \begin{cases} x = 8\alpha + 5\beta + 25\gamma \\ y = 8\beta + \gamma \\ z = 26\alpha + 5\beta + 13\gamma \end{cases}$$

ومنه بحل جملة هذه المعادلات حلا مشتركا نجد أن:

$$\alpha = \frac{-33x - 20y + 65z}{1426}$$

$$\beta = \frac{-13x + 273y + 4z}{2139}$$

$$\gamma = \frac{104x - 45y - 32z}{2139}$$

ومنه ينتج أن:

$$(x,y,z) = \frac{-33x-20y+65z}{1426}\,a_1 + \frac{-13x+273y+4z}{2139}\,a_2 + \frac{104x-45y-32z}{2139}\,a_3$$

ومنه بتعويض أسطر مصفوفة النص المشفر في العلاقة السابقية نحصل على العلاقات التالية:

$$(18,16,36) = a_1 + 2a_2 + 0a_3$$
$$(13,8,31) = a_1 + a_2 + 0a_3$$
$$(25,1,13) = 0a_1 + 0a_2 + a_3$$

ومنه فإن مصفوفة هيل او المفتاح ماهي إلا عبارة عن مصفوفة الامثال في العلاقات الثلاث السابقة اي أن:

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ومن هنا نستنتج سهولة كسر دالة تشفير هيل لذلك يمكننا أن نلاحظ اهمية اضافة خاصية الانسحاب والدوران بالإضافة إلى النظائر الجمعية على دالة هيل اذ يستحيل على أحدهم معرفة مصفوفة هيل لانعدام امكانية الحل السابقة وهذا ما نسعى إليه وهو إضافة الأمان إلى الرسائل والنصوص المبثوثة عبر الوسائط المختلفة فمثلا في مثالنا هذه اضافة خاصية الدوران لمصفوفة هيل سوف يكون امام اي شخص يحاول كسر المفتاح

$3! = 6$ احتمالات للمصفوفة A

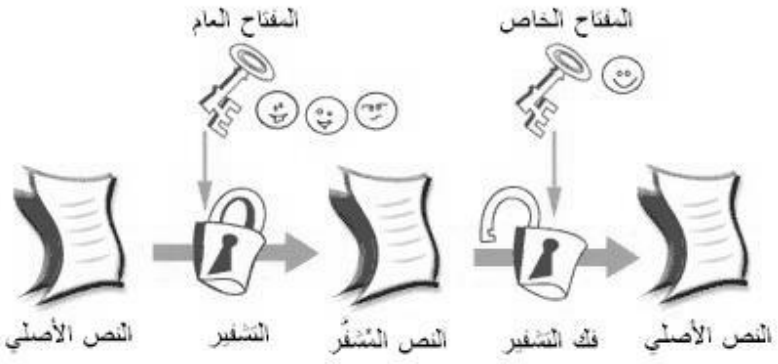
2. تشفير المفتاح العام:(Public Key Cryptography).

أو ما يعرف بالتشفير اللا متماثل (Cryptography Asymmetric). تم تطوير هذا النظام في السبعينات في بريطانيا وكان استخدامه حكراً على قطاعات معينة من الحكومة. ويعتمد في مبدأه على وجود مفتاحين

وهما المفتاح العام Public keyوالمفتاح الخاص Privet key، حيث إن المفتاح العام هو لتشفير الرسائل والمفتاح الخاص لفك تشفير الرسائل.

المفتاح العام يرسل لجميع الناس أما المفتاح الخاص فيحتفظ به صاحبه ولا يرسله لأحد. فمن يحتاج أن يرسل لك رسالة مشفرة فإنه يستخدم المفتاح العام لتشفيرها ومن ثم تقوم باستقبالها وفك تشفيرها بمفتاحك الخاص. فيما يلي رسم توضيحي صورة (3) .

صورة3: توضح عمل التشفير باستخدام المفتاح العام والمفتاح الخاص.



بعض الأمثلة على أنظمة تشفير المفتاح العام:

EAS, DAS, PGP, DSA, Deffie-Hellman, Elgamal, RSA

جميع هذه الأنظمة تعتمد على مبدأ التشفير اللا تماثلي أو التشفير باستخدام المفتاح العام والمفتاح الخاص.

مزايا وعيوب التشفير التقليدي والتشفير باستخدام المفتاح العام:

التشفير التقليدي أسرع بكثير باستخدام أنظمة الكمبيوتر الحديثة، ولكنه يستخدم مفتاح واحد فقط. إلا أنه أكثر عرضة للاختراقات . أما تشفير المفتاح العام فيستخدم مفتاحين في عملية التشفير وفك التشفير، وهو أقوى وأقل عرضة للاختراقات، ولكنه أبطأ من التشفير التقليدي. ونتيجة لهذه المزايا والعيوب أصبحت الأنظمة الحديثة تستخدم كلا الطريقتين حيث إنها تستخدم الطريقة التقليدية للتشفير , أما تبادل المفتاح السري الواحد بين الأطراف المتراسلة تتم من خلال استخدام طريقة تشفير المفتاح العام.

قياس قوة التشفير[3]:

التشفير قد يكون قوياً أو ضعيفاً، حيث إن مقياس القوة للتشفير هو الوقت (التعقيد) والمصادر المتطلبة لعملية كشف النصوص غير مشفرة من النصوص المشفرة فهي تعتمد على عدد الخانات المكونة لكل رقم و تقاس بالبت فمثلا إذا كان الرقم مكون من 40 خانة فإن القوة ستكون 40 بت إذا كان الرقم عبارة عن 56 خانة تكون قوة التشفير 56 بت وهكذا. علما بأن التكنولوجيا المتوفرة في هذا المجال يمكن أن توفر قوة تشفير تصل إلى أكثر من 3000 بت ولكن لم تسمح الحكومة الأمريكية حتى الان بتداول قوة تشفير أكثر من 128 بت لأنه كاف جدا لحماية التجارة الإلكترونية و جدير بالذكر أن الوقت اللازم ليتمكن أحد لصوص الإنترنت لفك شفرة بقوة 56 بت هو 22 ساعة وخمسة عشر دقيقة ، أما الوقت اللازم لفك شفرة بقوة 128 بت باستخدام التكنولوجيا الحالية لفك الشفرات فهو 2 ترليون سنة!! لأن اللص في حالة 56 بت بحاجة لتجربة 72 كوادرليون من الاحتمالات (يعني رقم و أمامه 15 صفر) أما في قوة 128 فإن الاحتمالات المطلوبة للتجربة تصل إلى عدد فلكي وهو 340 انديسليون (يعني رقم وأمامه 36 صفر)ولذلك لم نسمع أبدا بأن معلومة تم تشفيرها بهذه القوة قد تم فكها من قبل هؤلاء اللصوص المحترفين و نحن لا نعتقد بأن أحد يمكنه فعل ذلك على

الأقل في المستقبل القريب أو المنظور ولذلك تسوق على شبكة الإنترنت وأنت مطمئن البال بشرط التأكد من قوة التشفير المستخدمة من قبل الموقع الذي تود الشراء منه و كذلك التأكد من قوة التشفير في متصفحك...

أمن الخوارزميات (Security of Algorithms):

توفر الخوارزميات المختلفة درجات مختلفة في الأمان إذ إنها تعتمد على مقدار الصعوبة المطلوبة لغرض كسر هذه الخوارزميات , إذا كانت الكلفة المطلوبة لكسر خوارزمية معينة اكبر من قيمة البيانات المشفرة عندئذ فانه من المحتمل أن تكون هذه الخوارزمية أمنة . إذا كان الوقت المطلوب لكسر خوارزمية معينة اكبر من وقت البيانات المشفرة لبقائها أمينة , عند ذلك فإنها قد تكون خوارزمية أمنة . إذا كانت البيانات المشفرة بمفتاح مفرد اقل من كمية البيانات الضرورية لكسر الخوارزمية , فعند ذلك من المحتمل أن تكون أمنة . يقال من " المحتمل " بسبب انه يوجد دائما هجومات جديدة في تحليل الشفرة . من ناحية أخرى , فان قيمة معظم البيانات تتناقص مع الزمن . انه من المهم جدا أن تكون قيمة البيانات دائما اقل من الكلفة المطلوبة لكسر الأمنية المطلوبة لحمايتها .

صنف العالم كيندسن (Lars Knudsen) الأنواع التالية من الكسر لأي خوارزمية :

1) الكسر الكلي (Total Break) : محلل الشفرة يجد المفتاح , k , بحيث إن

$D_k (C) = P$

2) الاستنتاج العام (Global Deduction): محلل الشفرة يجد خوارزمية بديلة , A , مكافئة إلى D_k (C), بدون معرفة المفتاح K.

3) الاستنتاج المحلي (Local) Deduction Instance): محلل الشفرة يجد النص الواضح لنص مشفر مفترض .

4) استنتاج المعلومات (Information Deduction) : محلل الشفرة يحصل على بعض المعلومات حول المفتاح أو النص الواضح . هذه المعلومات يمكن أن تكون بتات قليلة من المفتاح , بعض المعلومات حول صيغة النص الواضح , والخ .

يقال عن الخوارزمية أنها أمنة غير مشروطة (Unconditional Secure) في حالة مهما تكن كمية النص المشفر الذي يملكه العدو , كأن لا يوجد معلومات كافية لغرض استرجاع النص الواضح . في الواقع , فان فقط شفرة الوسادة (One –Time Pad) هي غير قابلة للكسر معطية موارد غير محددة . كل أنظمة التشفير الأخرى هي قابلة للكسر في هجوم النص المشفر فقط , وذلك ببساطة بمحاولة البحث عن كل المفاتيح الممكنة واحدا بعد الآخر وتدقيق فيما إذا كان النص الواضح الناتج ذو معنى . هذا يطلق عليه هجوم القوة الوحشية (Brute Force Attack) .

يبدي علم التشفير اهتماماً متميزاً بأنظمة التشفير التي هي حسابيا متعذرة الكسر. إن الخوارزمية تعتبر أمنة حسابيا (Computationally Secure) (أحيانا يطلق عليها قوية) إذا لم يكن بالإمكان كسرها بالموارد المتوفرة , أما حاليا أو في المستقبل . إن ما تحتويه الموارد المتوفرة يكون مفتوحا للاعتراض .

يمكن قياس التعقيد لأي هجوم بعدة وسائل مختلفة :

1) تعقيد البيانات (Data Complexity) : كمية البيانات المطلوبة كمدخل إلى الهجوم .

2) تعقيد المعالجة (Processing Complexity) : الوقت المطلوب لتنفيذ الهجوم , هذا غالبا ما يطلق عليه عامل الشغل (Work Factor) .

3) متطلبات الخزن (Storage Requirements) : كمية الذاكرة المطلوبة لتنفيذ الهجوم

تعاريف وخواص لا بد منها :

تعريف قابلية القسمة :

إذا كان لدينا عددان صحيحان a,b حيث $a \neq 0$.نقول إن a يقسم b إذا وجد عدد صحيح ثالث c بحيث $b = a.c$.ونشير إلى ذلك بالرمز $a|b$.ونقول إن b يقبل القسمة على a.

مثال (4):

$a = 3, b = 27$ نلاحظ أن $a|b$ لأنه يوجد $c = 9$ بحيث يتحقق $3.9 = 27$.

تعريف خوارزمية القسمة :

ليكن y, b عددان صحيحان بحيث $b > 0$. يوجد عددان صحيحان q, r بحيث $y = b.q + r$.

نسمي y المقسوم ، b القاسم ، q حاصل القسمة ، r باقي القسمة.

مثال (5):

$$65 = 3.(21) + 2$$

وكذلك $-21 = 5 \times (-5) + 4$

تعريف العدد الأولي :

هو عدد أكبر من الواحد لا يقبل القسمة إلا قاسمين مختلفين هما الواحد والعدد نفسه.العدد غير الأولي يسمى عدداً مركباً.

مثال (6):

الأعداد 2، 3، 11 ، 29 ، 53 هي أعداد أولية.

وأما الأعداد 4، 9، 10 ، 39 ، 51 فهي أعداد غير أولية.

نتائج :

- جميع الأعداد الصحيحة التي هي أكبر من الواحد لها قاسم أولي.
- الأعداد الأولية غير منتهية.
- إذا كان n عدداً مركبا (مؤلف من جداء عددين أوليين على الأقل) فإن له قاسم أولي لا يتعدى $\sqrt{n}$.

مثال (7):

$n = 39$ (عدد مركب) عندئذٍ $\sqrt{n} \approx 6$ وبالتالي قواسم n الأولية التي هي أصغر من $\sqrt{n}$ هي $2,3,5$ من بينها 5 قاسم أولي للعدد n.

مثال (8):

$n = 56$(عدد مركب) عندئذٍ $\sqrt{n} \approx 7$ وبالتالي قواسم n الأولية التي هي أصغر من $\sqrt{n}$ هي $2, 3, 5, 7$ من بينها 7 قاسم أولي للعدد n .

مثال (9):

$n = 101$ (عدد أولي) عندئذٍ $\sqrt{n} \approx 10$ وبالتالي قواسم n الأولية التي هي أصغر من $\sqrt{n}$ هي $2, 3, 5, 7$ لا يوجد بينها قاسم أولي للعدد n.

تعريف القاسم المشترك الأعظم :

القاسم المشترك الأعظم لعددين x, y هو أكبر عدد صحيح يقبل القسمة على العددين x, y معاً. ونرمز له بالرمز $\gcd(x, y)$ أو اختصاراً (x, y).

مثال (10): أوجد $gcd(30,18)$

الحل :

قواسم 30 هي : $1, 2, 3, 5, 6, 10, 15, 30$

قواسم 18 هي : $1, 2, 3, 6, 9, 18$

القواسم المشتركة للعددين $1, 2, 3, 6$

أكبر هذه القواسم هو 6 أي أن $(30,18) = 6$.

مثال (11): أوجد $gcd(21,20)$

الحل :

قواسم 21 هي : $1,3,7,21$

قواسم 20 هي : $1,2,4,5,10,20$

القواسم المشتركة للعددين $(21,20)$ هي : 1 فقط، إذن $gcd(21,20) = 1$.

تعريف العددان الأوليان فيما بينهما :

نقول إن العددين x, y أوليان فيما بينهما إذا كان $gcd(x, y) = 1$.

مثال (12):

العددان $12,35$ أوليان فيما بينهما لأن $(12,35) = 1$.

مثال (13):

العددان $110,55$ ليسا أوليين فيما بينهما لأن $(110,55) = 55 \neq 1$.

نتيجة :

إذا كان x, y غير أوليين فيما بينهما فإن $\frac{x}{(x,y)}, \frac{y}{(x,y)}$ أوليان فيما بينهما.

مثال (14):

$(30,18) = 6 \neq 1$ لكن $\frac{18}{6} = 3$, $\frac{30}{6} = 5$ نلاحظ أن $(3,5) = 1$.

تعريف 1 (زمرة أولر)[2]:

نسمي M_n مجموعة جميع الأعداد الأولية نسبياً مع العدد n أي إن :

$$M_n = \{a \in N ; 1 \leq a < n; \gcd(a, n) = 1\} \qquad (3)$$

ولنعرف عليها عملية ضرب بالمقاس n . إن M_n مع عملية الضرب تشكل زمرة تدعى زمرة أولر , وبما إن M_n زمرة بالنسبة للضرب فالنظير لكل عنصر منها موجود .

نتيجة:

من التعريف 1 ينتج إنه عندما $n = 256$ فإن $M_n = \{1,3,5 \dots 255\}$

تعريف خوارزمية إقليدس :

إذا كان $c = qd + r$ فإن $(c,q) = (q,r)$.

مثال (15):

لدينا $22 = 3.6 + 4$ نلاحظ $(22,6) = 2 = (6,4)$.

مثال (16):

لدينا $51 = 2.20 + 11$ نلاحظ $(51,20) = 1 = (20,11)$.

تطبيق خوارزمية إقليدس في إيجاد القاسم المشترك الأعظم.

بتطبيق خوارزمية إقليدس بشكل متتالي نحصل على ما يلي :

$$c = qd_0 + r_0$$
$$q = r_0 d_1 + r_1$$
$$r_0 = r_1 d_2 + r_2$$
$$r_1 = r_2 d_3 + r_3$$
$$\vdots$$
$$r_{j-2} = r_{j-1} d_j + r_j \quad ; \; r_j = 0 \Rightarrow r_{j-2} = r_{j-1} d_j$$

فيكون :

$$(c,q) = (q,r_0) = (r_0,r_1) = (r_1,r_2) = \dots = (r_{j-2},r_{j-1}) = r_{j-1}$$

مثال (17):

أوجد $gcd(252,198)$ باستخدام خوارزمية إقليدس.

الحل:

$$252 = 198(1) + 54$$
$$\Rightarrow 198 = 54(3) + 36$$
$$\Rightarrow 54 = 36(1) + 18$$
$$\Rightarrow 36 = 18(2) + 0$$

ومنه

$$(252,198) = (198,54) = (54,36) = (36,18) = 18$$

مثال (18):

أوجد $gcd(53,51)$ باستخدام خوارزمية إقليدس.

الحل:

$$53 = 51(1) + 2$$
$$51 = 2(25) + 1$$
$$2 = 1(1) + 1$$
$$1 = 1(1) + 0$$

ومنه $gcd(53,51) = 1$ إذاً العددان أوليان فيما بينهما.

تعريف النظير الضربي :

نقول إن العدد $1 \le b < n$ هو النظير الضربي للعدد $1 \le a < n$ بالمقاس n إذا كإن باقي قسمة $a.b$ على العدد n هو الواحد أي : $a.b \equiv 1 \bmod(n)$

كيف يمكننا إيجاد النظير الضربي؟

الطريقة الأولى : بالاعتماد على خوارزمية إقليدس الممتدة.

الطريقة الثانية : بالاعتماد على نظرية أولر.

تعريف خوارزمية إقليدس الممتدة :

يمكن تمثيل القاسم المشترك الأعظم لعددين على الشكل $(x,y) = mx + ny$.

مثال (19):

اكتب $(26,21)$ على الشكل $m.26+n.21$.

الحل:

نبدأ كما في خوارزمية إقليدس

$$26 = 21(1) + 5$$

$$21 = 5(4) + 1$$

$$5 = 5(1) + 0 \quad \Rightarrow \quad (26,21) = 1$$

$$1 = 21 - 4.5 \qquad (1)$$

$$5 = 26 - 1.21 \qquad (2)$$

نعوض (2) في (1) نجد

$$1 = 21 - 4(26 - 1.21)$$

$$1 = 21 - 4.26 + 4.21$$

$$1 = \underbrace{(5)}_{m}21 + \underbrace{(-4)}_{n}26$$

نتيجة هامة :

إذا كان x,y أوليين فيما بينهما، ومثلنا القاسم المشترك الأعظم لهما على الشكل $1 = mx + ny$ فإن $n\ y = 0\ mod\ n$ و $m\ x = 1\ mod n$ أي أن y هو النظير الضربي للعدد x بالمقاس n ، و m هو النظير الضربي x للعدد بالمقاس y

مثال (20):

بالعودة إلى المثال السابق نجد أن $21^{-1} = 5\ mod\ 26$

وكذلك $26^{-1} = -4\ mod\ 21$.

مثال (21):

أوجد $59^{-1}\ mod\ 13$ بالاعتماد على خوارزمية إقليدس الممتدة.

الحل:

$$59 = 4(13) + 7$$
$$13 = 1(7) + 6$$
$$7 = 1(6) + 1$$
$$6 = 6(1) + 0 \qquad \Rightarrow (59,13) = 1$$

كما نلاحظ أن :

$$1 = 7 - 1.6 = 7 - 1(13 - 1.7) =$$
$$= 7 - 13 + 7 = 2(7) + (-1)(13) =$$
$$= 2(59 - 4.13) + (-1)(13) = (2)(59) + (-9)(13)$$

نستنتج أن : $59^{-1} \bmod 13 = 2$

وكذلك $(-9^{-1})mod2 = 13$

$(-9^{-1})mod59 = 13$

مثال (22):

أوجد $217^{-1} \bmod 257$ بالاعتماد على خوارزمية إقليدس الممتدة.

الحل:

$$257 = (1)(217) + 40$$
$$217 = (5)(40) + 17$$
$$40 = (2)(17) + 6$$
$$17 = (2)(6) + 5$$
$$6 = (1)(5) + 1$$
$$5 = (5)(1) + 0 \qquad \Rightarrow (257,217) = 1$$

باتباع نفس الخطوات في التمرين السابق نجد

$$217^{-1} = -45 \bmod 257 = 212 \bmod 257$$

مثال (23):

أوجد $3^{-1} mod 256$ بالاعتماد على خوارزمية إقليدس الممتدة.

الحل:

$$256 = 3.85 + 1$$
$$\Rightarrow 1 = 256 - 3.85$$
$$\Rightarrow 1 = 256 + 3(-85)$$

كما نلاحظ أن :

$$-85 = 256 - 85 \; mod 256 = 171$$

نستنتج أن :

$$3^{-1} mod 256 = 171$$

للتأكد :

$$171.3 = 513 = 1 mod 256$$

مثال (24):

أوجد $5^{-1} mod 256$ بالاعتماد على خوارزمية إقليدس الممتدة.

الحل:

$$256 = 5.51 + 1$$
$$\Rightarrow 1 = 256 - 5.51$$
$$\Rightarrow 1 = 256 + 5(-51)$$

كما نلاحظ أن :

$$-51 = 256 - 51 \; mod 256 = 205$$

نستنتج أن :

$$5^{-1} mod 256 = 205$$

للتأكد :

$$205.5 = 1025 = 1 mod 256$$

مثال (25):

أوجد 2551^{-1} **mod** 5051 بالاعتماد على خوارزمية إقليدس الممتدة.

الحل:

$$5051 = (1)(2551) + 2500$$
$$2551 = (1)(2500) + 51$$
$$2500 = (49)(51) + 1$$
$$49 = (49)(1) + 0 \qquad \Rightarrow (5051, 2551) = 1$$

كما نلاحظ أن :

$$1 = (50)(5051) + (-99)(2551)$$

نستنتج أن :

$$2551^{-1} = -99 \bmod 5051 = 4952 \bmod 5051$$

للتأكد :

$$4952.2551 = 12632552 = 1 \bmod 5051$$

تعريف مجموعة الأعداد الأولية نسبياً :

ليكن n عدداً صحيحاً. نرمز لمجموعة الأعداد الأولية نسبياً مع n بالرمز M_n أي أن :

$$M_n = \{x \in N \; ; \; (x,n) = 1\}$$

ونرمز لعدد عناصر M_n بالرمز $\varphi(n)$. فمثلاً $M_{257} = \{1,2,3,\ldots,256\}$ فيكون $\varphi_{257} = 256$.

مبرهنة :

إذا كان $n = p_1^{k_1} p_2^{k_2} \ldots p_t^{k_t}$ فإن : $\boldsymbol{\varphi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)..\left(1 - \frac{1}{p_t}\right)}$

نتائج :

- إذا كان n أولياً فإن $\varphi(n) = n - 1$.
- إذا كان nعدداً مركباً(غير أولي) ، وكان $n = p_1^{k_1} p_2^{k_2} \ldots p_t^{k_t}$ فإن :

$$\varphi(n) = (p_1 - 1)p_1^{k_1 - 1}(p_2 - 1)p_2^{k_2 - 1}..(p_t - 1)p_t^{k_t - 1}$$

مثال (26):

-إذا كان $n = 257$ فإن $\varphi(n) = 257 - 1 = 256$.

-إذا كان $n = 256 = 2^8$ فإن $\varphi(n) = (2-1)2^{8-1} = 2^7 = 128$.

مبرهنة (نظرية أولر) في إيجاد المقلوب :

إذا كان $(x,n) = 1$ فإن $x^{\varphi(n)} = 1 \bmod n$.

نتيجة :

إذا كان $(x,n) = 1$ فإن $x^{-1} = x^{\varphi(n)-1} \bmod n$.

نتيجة :

إذا كان n عدداً أولياً وكان $(x,n) = 1$ فإن $x^{-1} = x^{n-2} \bmod n$.

مثال (27):

إذا كان $n = 28$ ، $x = 9$ فإن $(28,9) = 1$ كما أن

$$n = 2^2.7 \Rightarrow \varphi(n) = 28\left(1-\frac{1}{2}\right)\left(1-\frac{1}{7}\right) = 12$$

أو بطريقة أخرى :

$$n = 2^2.7 \Rightarrow \varphi(n) = (2-1)2^{2-1}(7-1)7^{1-1} = 12$$

إذن

$$9^{-1} = 9^{11} = \left(9^2\right)^5.9 = 81^5.9 = 25^5.9 = (-3)^5.9 =$$

$$= 9^2(-27) = 81.1 = 81 = 25 \bmod 28$$

أي أن نظير (9) هو (25) بالمقاس (28).

تعريف النظير الجمعي:

ندعو العدد $a' = n - a$ بالنظير الجمعي للعدد a بالمقاس n.

مبرهنة: إذا كان $(a,n) = 1$ فإن $(a',n) = 1$.

مثال (28):

$$(27,52) = 1 = (25,52)$$

مبرهنة shamma: [9]

إذا كان $a.b = 1 \bmod n$ فإن $a'.b' = 1 \bmod n$.

مثال (29):

أوجد $17^{-1} mod 20$ بالطرق التالية:

1) طريقة shamma
2) طريقة اقليدس
3) طريقة أولر

الحل

1) طريقة shamma نعلم أن:

$$3.7 = 21 = 1mod20$$

وحسب مبرهنة shamma السابقة أي بأخذ المتمم الجمعي ينتج أن:

$$(20-3)(20-7) = 1mod20$$
$$(17)(13) = 1mod\ 20 \Rightarrow 17^{-1}mod20$$

2) طريقة اقليدس

$$20 = 1.17 + 3$$
$$\Rightarrow 17 = 3.5 + 5$$
$$\Rightarrow 3 = 2.1 + 1$$
$$\Rightarrow 1 = 3 - 2.1 =$$
$$3 - 1(17 - 3.5) =$$
$$20 - 17 - 1\big(17 - 5(20 - 17)\big)$$
$$\Rightarrow 1 = 6.20 - 7.17$$
$$\Rightarrow 17^{-1} = -7 = 20 - 7 = 13mod20$$

3) طريقة أولر نعلم أن: $\varphi(20) = 8$

وبالتالي:

$$17^{-1}mod20 = 17^{\varphi(20)-1} =$$
$$17^7 = (-3)^6.(-3) =$$
$$81(-3).(9)mod20 = 1.(-27) =$$
$$-7mod20 = 13mod20$$

نتيجة وجدنا أن $17^{-1}mod20 = 13$ بالطرق الثلاثة السابقة.

خلاصة الفصل الأول:

في هذا الفصل تم ذكر التعاريف والمبرهنات الأساسية المتعلقة بالتشفير بشكل عام ذكر أهم أنواع التشفير التقليدي وذكر قوة التشفير وأمن الخوارزميات.

الفصل الثاني

التشفير باستخدام متجهات تولد مصفوفات مثلثية ذات طبيعة خاصة

مقدمة:

نقدم في هذا الفصل نمطاً جديداً من التشفير باستخدام مفتاح التشفير وهو عبارة عن مصفوفة عددية خاصة (مثلثية عليا أو سفلى) محققة لشرطي مصفوفة Hill (وهي مصفوفة مربعة لها نفس مقاس مصفوفة النص الأصلي ومحددها غير معدوم وأولي نسبياً مع المقاس n)، وهذه المصفوفات متولدة من متجهات مركباتها أعداد أولية نسبياً مع المقاس $n = 256$ و علاقة التشفير هي ناتج ضرب مصفوفة النص الأصلي بمفاتح التشفير ، ويتم ذلك من خلال الاعتماد على ما يلي:

1. على نظام ASCII ذي المقاس $n = 256$.
2. على خوارزمية تشفير تنتج لنا تشفير نصوص كبيرة الحجم
3. إمكانية توسيع الفكرة بالاعتماد على مصفوفات لأعداد أولية نسبياً مع لمقاس $n = 256$ غير متبعة في الحالات التقليدية

الهدف من التشفير :

هدف التشفير الحفاظ على سرية الرسائل المبثوثة عبر قنوات الاتصال المختلفة مثل الجوال والبريد الالكتروني والفيس بوك والوتس آب والحكومات الالكترونية .. الخ [3] . الآن ومع التطور السريع لتشفير البيانات في الوتس والفيس بوك وغيره , بات من الضروري اعتماد أنماطاً جديدة من التشفير سهلة الاستخدام ويصعب كسرها , بحيث تغطي هذا النوع من العمل .

ولهذا اعتمدنا في هذا العمل مفتاح التشفير وهو المتجه ذي m بعد أي

$$V = (a_1, a_2, \cdots, a_m) \qquad (1)$$

حيث إن:

$$\gcd(a_i, n) = 1 , \qquad i = 1, \cdots, m \qquad (2)$$

و إن $n = 256$ عدد محارف نظام ASCII.

فيما يلي نذكر ببعض الأعداد الأولية نسبياً بالمقاس $n = 256$ (مرتبة وفق العدد و نظيره الضربي) مرفقة في الجدول التالي (1) : [1] و[7]

Decimals	Symmetric product	Decimals	Symmetric product
1	1	129	129
3	171	131	43
5	205	133	77
7	183	135	55
9	57	137	185
11	163	139	35
13	197	141	69
15	239	143	111
17	241	145	113
19	27	147	155
21	61	149	189
23	167	151	39
25	41	153	169
27	19	155	147
29	53	157	181
31	223	159	95
33	225	161	97
35	139	163	11
37	173	165	45
39	151	167	23
41	25	169	153
43	131	171	3

45	165	173	37
47	207	175	79
49	209	177	81
51	251	179	123
53	29	181	157
55	135	183	7
57	9	185	137
59	243	187	115
61	21	189	149
63	191	191	63
65	193	193	65
67	107	195	235
69	141	197	13
71	119	199	247
73	249	201	121
75	99	203	227
77	133	205	5
79	175	207	47
81	177	209	49
83	219	211	91
85	253	213	125
87	103	215	231
89	233	217	105
91	211	219	83
93	245	221	117
95	159	223	31
97	161	225	33

203	227	75	99
237	229	109	101
215	231	87	103
89	233	217	105
195	235	67	107
229	237	101	109
15	239	143	111
17	241	145	113
59	243	187	115
93	245	221	117
199	247	71	119
73	249	201	121
51	251	179	123
85	253	213	125
255	255	127	127

تعاريف ومبرهنات هامة:

نحتاج لبعض التعاريف الخاصة اللازمة لإظهار النتائج الهامة في هذا الفصل

1. مفتاح التشفير:

هو المتجه (4) $V = (a_1, a_2, \cdots, a_m)$ **حيث إن:**

$$\gcd(a_i, n) = 1 \, , \; i = 1, \cdots, m$$

2. مفتاح فك التشفير:

هو المتجه (5) $B = (a_1^{-1}, a_2^{-1}, \cdots, a_m^{-1})$ **حيث إن:**

$$\gcd(a_i^{-1}, n) = 1 \, , \qquad i = 1, \cdots, m$$

3. مصفوفة المتجه A_V :

هي مصفوفة مثلثية سفلى متولدة من المتجه V في (4) كما يلي:

$$\boldsymbol{A_V} = \begin{bmatrix} a_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_1 & a_2 & 0 & 0 & 0 & 0 & 0 \\ a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 \\ a_1 & a_2 & a_3 & a_4 & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ a_1 & a_2 & a_3 & . & . & a_{m-1} & a_m \end{bmatrix} \qquad (6)$$

4. مبرهنة (1)

المصفوفة العكسية السفلى لمصفوفة المتجه A_V :

هي مصفوفة ${A_V}^{-1}$ مثلثية سفلى ناتجة من المصفوفة $\boldsymbol{A_V}$ كما يلي :

$$\boldsymbol{A_V}^{-1} = \begin{bmatrix} {a_1}^{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ -{a_2}^{-1} & {a_2}^{-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & -{a_3}^{-1} & {a_3}^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & -{a_4}^{-1} & {a_4}^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & -{a_5}^{-1} & {a_5}^{-1} & 0 & 0 \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & -{a_{m-1}}^{-1} & {a_m}^{-1} \end{bmatrix} \qquad (7)$$

الاثبات:

بما أن المصفوفتان ${A_V}^{-1}, A_V$ من نفس المرتبة m فإن ناتج ضرب المصفوفتين هو :

$$\boldsymbol{A_V}.\boldsymbol{A_V}^{-1} = \begin{bmatrix} a_1.{a_1}^{-1} & 0 & 0 & 0 & 0 & 0 & .. & 0 \\ -a_1.{a_1}^{-1} + a_2.{a_2}^{-1} & a_2.{a_2}^{-1} & 0 & 0 & 0 & 0 & .. & 0 \\ 0 & -a_2.{a_2}^{-1} + a_3.{a_3}^{-1} & a_3.{a_3}^{-1} & 0 & 0 & 0 & .. & 0 \\ 0 & 0 & -a_3.{a_3}^{-1}. + a_4.{a_4}^{-1} & a_4.{a_4}^{-1} & 0 & 0 & .. & 0 \\ . & . & . & . & . & ... & . & . \\ 0 & 0 & 0 & 0 & 0 & 0 & -{a_{m-1}}^{-1} & {a_m}^{-1} \end{bmatrix}$$

أي أن:

$$\boldsymbol{A_V}.\boldsymbol{A_V}^{-1} = I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \dots & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ & \cdot & \dots & \cdot & & \cdot & & \\ \cdot & & \cdot & & \cdot & & \cdot & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

5. نتيجة (1):

المصفوفة $A_V \; mod256. {A_V}^{-1} \; mod256 = I$

الاثبات:

بما أن المتجه $V = (\, a_1, a_2, \cdots, a_m\,)$ **نظير للمتجه** $B = (a_1^{-1}, a_2^{-1}, \cdots, a_m^{-1})$ حيث إن:

$$a_i.a_i^{-1} = 1 \text{mod} 256$$

فإن:

$$\boldsymbol{A_V}.\boldsymbol{A_V}^{-1} \; mod\; 256 =$$

$$= \begin{bmatrix} a_1.{a_1}^{-1} & 0 & 0 & 0 & 0 & 0 \dots & 0 \\ -a_1.{a_1}^{-1} + a_2.{a_2}^{-1} & a_2.{a_2}^{-1} & 0 & 0 & 0 & 0 \dots & 0 \\ 0 & -a_2.{a_2}^{-1} + a_3.{a_3}^{-1} & a_3.{a_3}^{-1} & 0 & 0 & 0 \dots & 0 \\ 0 & 0 & -a_3.{a_3}^{-1}. + a_4.{a_4}^{-1} & a_4.{a_4}^{-1} & 0 & 0 \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \dots & \cdot & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -{a_{m-1}}^{-1} & {a_m}^{-1} \end{bmatrix} mod\; 256$$

$$= \begin{bmatrix} a_1.{a_1}^{-1}\, mod\, 256 & 0 & 0 & 0 & 0 & 0 \dots & 0 \\ 0 & a_2.{a_2}^{-1} mod\, 256 & 0 & 0 & 0 & 0 \dots & 0 \\ 0 & 0 & a_3.{a_3}^{-1} mod\, 256 & 0 & 0 & 0 \dots & 0 \\ 0 & 0 & 0 & a_4.{a_4}^{-1} mod\, 256 & 0 & 0 \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \dots & \cdot & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -{a_{m-1}}^{-1} & {a_m}^{-1} mod\, 256 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & .. & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & .. & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & .. & 0 \\ 0 & 0 & 0 & 1 & 0 & .. & 0 & \\ & . & ... & . & & . & & \\ . & & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \boldsymbol{I}$$

6. مصفوفة المتجه B_V :

هي مصفوفة مثلثية عليا متولدة من المتجه V في (4) كما يلي:

$$\boldsymbol{B_V} = \begin{bmatrix} a_1 & a_2 & a_3 & .. & .. & a_{m-1} & a_m \\ 0 & a_2. & a_3 & .. & .. & a_{m-1} & a_m. \\ 0 & 0 & a_3 & .. & .. & a_{m-1} & a_m \\ 0 & 0 & 0 & .. & a_{m-2} & a_{m-1} & a_m \\ 0 & 0 & 0 & .. & 0 & a_{m-1} & a_m \\ 0 & 0 & 0 & 0 & 0 & 0 & a_m \end{bmatrix}$$

7. مبرهنة (2)

المصفوفة العكسية العليا للمصفوفة B_V :

هي مصفوفة ${B_V}^{-1}$ مثلثية عليا ناتجة من المصفوفة $\boldsymbol{B_V}$ كما يلي:

$$\boldsymbol{B_V}^{-1} = \begin{bmatrix} {a_1}^{-1} & -{a_1}^{-1} & 0 & .. & .. & 0 & 0 \\ 0 & {a_2}^{-1} & -{a_2}^{-1} & .. & .. & 0 & 0 \\ 0 & 0 & {a_3}^{-1} & -{a_3}^{-1} & .. & 0 & 0 \\ .. & & .. & .. & .. & .. & .. \\ 0 & 0 & 0 & 0 & a_{m-1} & a_m & \\ 0 & 0 & 0 & 0 & 0 & a_m & \end{bmatrix}$$

الاثبات :

بما أن المصفوفتان $B_V{}^{-1}, B_V$ من نفس المرتبة m فإن ناتج ضرب المصفوفتين هو :

$$B_V . B_V{}^{-1} = I$$

أي أن:

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 & .. & 0 \\ 0 & 1 & 0 & 0 & .. & 0 \\ 0 & 0 & 1 & 0 & .. & 0 \\ 0 & 0 & 0 & 1 & .. & 0 \\ .. & .. & .. & .. & 1 & 0 \\ 0 & 0 & 0 & .. & 0 & 1 \end{bmatrix}$$

8. نتيجة (2):

المصفوفة $B_V \; mod256 . {B_V}^{-1} \; mod256 = I$

الاثبات:

بما أن المتجه $V = (a_1, a_2, \cdots, a_m)$ **نظير للمتجه** $B = (a_1^{-1}, a_2^{-1}, \cdots, a_m^{-1})$

حيث إن: $a_i . a_i^{-1} = 1 \text{mod} 256$

فإن:

$$\boldsymbol{B_V . {B_V}^{-1}} \; mod \; 256 =$$

$$\begin{bmatrix} a_1 . {a_1}^{-1} & 0 & 0 & 0 & 0 & 0 .. & 0 \\ -a_1 . {a_1}^{-1} + a_2 . {a_2}^{-1} & a_2 . {a_2}^{-1} & 0 & 0 & 0 & 0.. & 0 \\ 0 & -a_2 . {a_2}^{-1} + a_3 . {a_3}^{-1} & a_3 . {a_3}^{-1} & 0 & 0 & 0 .. & 0 \\ 0 & 0 & -a_3 . {a_3}^{-1} . + a_4 . {a_4}^{-1} & a_4 . {a_4}^{-1} & 0 & 0 .. & 0 \\ . & . . . & . & . & ... & . & . . . & . \\ & & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & {a_{m-1}}^{-1} . {a_m}^{-1} \end{bmatrix} mod \; 256$$

$$= \begin{bmatrix} a_1 . {a_1}^{-1} \; mod \; 256 & 0 & 0 & 0 & 0 & 0 .. & 0 \\ 0 & a_2 . {a_2}^{-1} mod \; 256 & 0 & 0 & 0 & 0.. & 0 \\ 0 & 0 & a_3 . {a_3}^{-1} mod \; 256 & 0 & 0 & 0 .. & 0 \\ 0 & 0 & 0 & a_4 . {a_4}^{-1} mod \; 256 & 0 & 0 .. & 0 \\ . & . . . & . & . & ... & . & . . . & . \\ & & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & {a_{m-1}}^{-1} . {a_m}^{-1} mod \; 256 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & .. & 0 \\ 0 & 1 & 0 & 0 & .. & 0 \\ 0 & 0 & 1 & 0 & .. & 0 \\ 0 & 0 & 0 & 1 & .. & 0 \\ .. & .. & .. & .. & 1 & 0 \\ 0 & 0 & 0 & .. & 0 & 1 \end{bmatrix} = I$$

مثال (1):

بما أن نظير المتجه $V = (1,3,\ 5,\ 7,\ 9,11,13)$ هو الشعاع

$V = (1,171,\ 205,\ 183,\ 57,163,197)$ من جدول النظير الضربي (1) حيث إن:

$$\boldsymbol{A_V} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 3 & 3 & 3 & 3 & 3 & 3 \\ 0 & 0 & 5 & 5 & 5 & 5 & 5 \\ 0 & 0 & 0 & 7 & 7 & 7 & 7 \\ 0 & 0 & 0 & 0 & 9 & 9 & 9 \\ 0 & 0 & 0 & 0 & 0 & 11 & 11 \\ 0 & 0 & 0 & 0 & 0 & 0 & 13 \end{bmatrix}$$

وهي مصفوفة مثاثية عليا

$$\boldsymbol{A_V}.\boldsymbol{A_V}^{-1} = I \text{ وأن}$$

$$\boldsymbol{A_V}^{-1} = \begin{bmatrix} 1 & -1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & -1/5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/5 & -1/7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/7 & -1/9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/9 & -1/11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/11 & -1/13 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/13 \end{bmatrix}$$

واعتماداً على جدول النظائر الضربية (1) نجد:

$$\boldsymbol{A_V}^{-1}\ mod256 = \begin{bmatrix} 1 & -171 & 0 & 0 & 0 & 0 & 0 \\ 0 & 171 & -205 & 0 & 0 & 0 & 0 \\ 0 & 0 & 205 & -183 & 0 & 0 & 0 \\ 0 & 0 & 0 & 183 & -57 & 0 & 0 \\ 0 & 0 & 0 & 0 & 57 & -163 & 0 \\ 0 & 0 & 0 & 0 & 0 & 163 & -197 \\ 0 & 0 & 0 & 0 & 0 & 0 & 197 \end{bmatrix} mod256$$

وهي مصفوفة مثلثية عليا أيضاً.

مثال(2):

نشكل المصفوفة المثلثية السفلى من الشعاع V السابق فتكون المصفوفة:

$$\boldsymbol{B_V} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 5 & 0 & 0 & 0 & 0 \\ 1 & 3 & 5 & 7 & 0 & 0 & 0 \\ 1 & 3 & 5 & 7 & 9 & 0 & 0 \\ 1 & 3 & 5 & 7 & 9 & 11 & 0 \\ 1 & 3 & 5 & 7 & 9 & 11 & 13 \end{bmatrix}$$

وهي منقول المصفوفة A_V

مبرهنة (3):

إن $\boldsymbol{B_V}^{-1}$ هو منقول $\boldsymbol{A_V}^{-1}$ لأن

الإثبات:

$$\boldsymbol{A_V}.\boldsymbol{A_V}^{-1} = I \Rightarrow Trans\left(\boldsymbol{A_V}, \boldsymbol{A_V}^{-1}\right) = Trans\left(\boldsymbol{B_V}^{-1}\right).T(\boldsymbol{B_V}) = I$$

وذلك حسب خواص منقول جداء مصفوفتين:

وبما أن $\boldsymbol{B_V}^{-1}.\boldsymbol{B_V} = I \Leftrightarrow \boldsymbol{B_V}.\boldsymbol{B_V}^{-1} = I$

حيث إن: $Trans(\boldsymbol{B_V}) = \boldsymbol{A_V}$

$$Trans\left(\boldsymbol{B_V}^{-1}\right) = \boldsymbol{A_V}^{-1}$$

9. علاقة التشفير:

هي العلاقة $Y = A_V.X\ mod256$ (8)

أو هي العلاقة (9) $Y = B_V.X\ mod256$

10. علاقة فك التشفير

هي العلاقة (10) $X = {A_V}^{-1}.Y\ mod256$

أو هي العلاقة (11) $X = {B_V}^{-1}.Y\ mod256$

11. خوارزمية التشفير وفق (8):

1- نضع النص الأصلي P في مصفوفة A عدد أعمدتها m بعدد عناصر المفتاح.

2- نبدل حروف P بأرقام ASCII من برنامج Excel2010 فنحصل على X

3- نطبق العلاقة التالية :

$$Y = A_V.X$$

4- نأخذ جميع العناصر بالمقاس $n = 256$ فنحصل على (8) أي

$$Y = A_V.X\ mod256$$

5- نبدل الأرقام بالمقابلات من جدول ASCII فنحصل على الرسالة المشفرة رقمياً.

مثال (3):

لنشفر الرسالة: 'PROPERTIES OF MATRIX MULTIPLICATION ' وفق المفتاح

$A = (a_1, a_2, a_3, a_4, a_5, a_6, a_7) = (1,3,5,7,9,11,13)$ وفق المصفوفة (6) .

الحل:

1. نشكل مصفوفة النص الأصلي :

P	I	M	M	I
R	E	A	U	C
O	S	T	L	A
P	_	R	T	T
E	O	I	I	I
R	F	X	P	O
T	_	_	L	N

2. نحولها إلى أرقام موافقة لـ ASCII، فنجد:

$$X = \begin{bmatrix} 80 & 73 & 77 & 77 & 73 \\ 82 & 69 & 65 & 85 & 67 \\ 79 & 83 & 84 & 76 & 65 \\ 80 & 95 & 82 & 84 & 84 \\ 69 & 79 & 73 & 73 & 73 \\ 82 & 70 & 88 & 80 & 79 \\ 84 & 95 & 95 & 76 & 78 \end{bmatrix}$$

3. نولد المصفوفة A_V متولدة من المتجه V كما يلي :

$$A_V = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 5 & 0 & 0 & 0 & 0 \\ 1 & 3 & 5 & 7 & 0 & 0 & 0 \\ 1 & 3 & 5 & 7 & 9 & 0 & 0 \\ 1 & 3 & 5 & 7 & 9 & 11 & 0 \\ 1 & 3 & 5 & 7 & 9 & 11 & 13 \end{bmatrix}$$

4. نطبق علاقة التشفير $$Y = A_V.X \ mod\ 256 \quad (8)$$

فنحصل على المصفوفة:

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 5 & 0 & 0 & 0 & 0 \\ 1 & 3 & 5 & 7 & 0 & 0 & 0 \\ 1 & 3 & 5 & 7 & 9 & 0 & 0 \\ 1 & 3 & 5 & 7 & 9 & 11 & 0 \\ 1 & 3 & 5 & 7 & 9 & 11 & 13 \end{bmatrix} . \begin{bmatrix} 80 & 73 & 77 & 77 & 73 \\ 82 & 69 & 65 & 85 & 67 \\ 79 & 83 & 84 & 76 & 65 \\ 80 & 95 & 82 & 84 & 84 \\ 69 & 79 & 73 & 73 & 73 \\ 82 & 70 & 88 & 80 & 79 \\ 84 & 95 & 95 & 76 & 78 \end{bmatrix} mod 256$$

5. بإصلاح الناتج نجد:

$$Y = \begin{bmatrix} 80 & 73 & 77 & 77 & 73 \\ 70 & 24 & 16 & 76 & 18 \\ 209 & 183 & 180 & 200 & 87 \\ 1 & 80 & 242 & 20 & 163 \\ 110 & 23 & 131 & 165 & 52 \\ 244 & 25 & 75 & 21 & 153 \\ 56 & 236 & 30 & 241 & 143 \end{bmatrix}$$

6- نستبدل الأرقام بالمحارف فنحصل على النص المشفر

P	**I**	**M**	**M**	**I**
F	"**24**"	"**16**"	**L**	"**18**"
ر	٠	ٔ	ب	**W**
"**1**"	**P**	ٍ	"**20**"	**£**
n	"**23**"	***f***	**¥**	**4**
ô	"**25**"	**K**	"**21**"	**™**
8	ى	–	ٌ	ڈ

وهو النص المشفر.

ملاحظة:

قد لا يظهر مقابل محرفي لبعض الأرقام لكن هذا لا يعني عدم وجوده وإنما هو محرف مخفي فقط لذلك نبقي على الرقم المقابل ضمن اشارتي تنصيص " " مثل العدد "24" و "18" في السطر الثاني.

12. خوارزمية فك التشفير وفق (10):

يتم بشكل معاكس لعملية التشفير

1- نضع النص المشفر في مصفوفة عدد أعمدتها يساوي m بعدد عناصر المفتاح العكسي.

$$A^{-1} = (a_1^{-1}, a_2^{-1}, \cdots, a_m^{-1})$$

2- نعوض المحارف بالمقابلات العددية في ASCII من الجدول (2).

3- نطبق العلاقة $X = {A_V}^{-1}.Y$

4- نأخذ جميع العناصر بالمقاس $n = 256$ فنحصل على (9) أي

$$X = A_V{}^{-1}.Y \; mod 256$$

نبدل الأرقام بالمحارف في نظام ASCII فنحصل على الرسالة الأصلية.

مثال (4):

لنفك التشفير عن الرسالة المشفرة في المثال (1) وفق المفتاح العكسي للمفتاح

$A = (a_1, a_2, a_3, a_4, a_5, a_6, a_7) = (1, 3, 5, 7, 9, 11, 13)$ وفق المصفوفة (7) .

الحل:

1. نستبدل مصفوفة النص المشفر بالأرقام من جدول ASCII فنجد :

$$Y = \begin{bmatrix} 80 & 73 & 77 & 77 & 73 \\ 70 & 24 & 16 & 76 & 18 \\ 209 & 183 & 180 & 200 & 87 \\ 1 & 80 & 242 & 20 & 163 \\ 110 & 23 & 131 & 165 & 52 \\ 244 & 25 & 75 & 21 & 153 \\ 56 & 236 & 30 & 241 & 143 \end{bmatrix}$$

2. نشكل مصفوفة المفتاح العكسي B كما في المصفوفة (7) وهي مصفوفة مثلثية عليا:

$$B = A_1{}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -171 & 171 & 0 & 0 & 0 & 0 & 0 \\ 0 & -205 & 205 & 0 & 0 & 0 & 0 \\ 0 & 0 & -183 & 183 & 0 & 0 & 0 \\ 0 & 0 & 0 & -57 & 57 & 0 & 0 \\ 0 & 0 & 0 & 0 & -163 & 163 & 0 \\ 0 & 0 & 0 & 0 & 0 & -197 & 197 \end{bmatrix}$$

3. نطبق العلاقة العكسية(10):

$$X = {A_1}^{-1}.Y \ \ mod\ 256 =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -171 & 171 & 0 & 0 & 0 & 0 & 0 \\ 0 & -205 & 205 & 0 & 0 & 0 & 0 \\ 0 & 0 & -183 & 183 & 0 & 0 & 0 \\ 0 & 0 & 0 & -57 & 57 & 0 & 0 \\ 0 & 0 & 0 & 0 & -163 & 163 & 0 \\ 0 & 0 & 0 & 0 & 0 & -197 & 197 \end{bmatrix} . \begin{bmatrix} 80 & 73 & 77 & 77 & 73 \\ 70 & 24 & 16 & 76 & 18 \\ 209 & 183 & 180 & 200 & 87 \\ 1 & 80 & 242 & 20 & 163 \\ 110 & 23 & 131 & 165 & 52 \\ 244 & 25 & 75 & 21 & 153 \\ 56 & 236 & 30 & 241 & 143 \end{bmatrix} mod\ 256$$

4. فنحصل على مصفوفة النص الأصلي رقمياً:

$$X = \begin{bmatrix} 80 & 73 & 77 & 77 & 73 \\ 82 & 69 & 65 & 85 & 67 \\ 79 & 83 & 84 & 76 & 65 \\ 80 & 95 & 82 & 84 & 84 \\ 69 & 79 & 73 & 73 & 73 \\ 82 & 70 & 88 & 80 & 79 \\ 84 & 95 & 95 & 76 & 78 \end{bmatrix}$$

5. نبدل الأرقام المحارف في جدول ASCII فنحصل على مصفوفة النص الأصلي:

P	**I**	**M**	**M**	**I**
R	**E**	**A**	**U**	**C**
O	**S**	**T**	**L**	**A**
P	**_**	**R**	**T**	**T**
E	**O**	**I**	**I**	**I**
R	**F**	**X**	**P**	**O**
T	**_**	**_**	**L**	**N**

فالرسالة الأصلية هي:

PROPERTIES OF MATRIX MULTIPLICATION′

13. ملاحظات:

1) خوارزمية التشفير وفق العلاقة (9) تتم بطريقة مشابهة تماماً لخوارزمية التشفير وفق العلاقة (8).
2) خوارزمية فك التشفير وفق العلاقة (11) تتم بطريقة مشابهة تماماً لخوارزمية التشفير وفق العلاقة (10).
3) خوارزمية التشفير وفك التشفير السابقتين قابلتين للتطبيق برمجياً بلغة C# (سي شارب) وهذا ما سنراه لاحقاً في قسم الملحقات حيث نورد أمثلة تطبيقية على ذلك.

ملخص الفصل الثاني:

قدمنا في هذا الفصل تشفير جديد يعتمد على أشعة خاصة تولد مصفوفات مربعة من خلال المبرهنتين الجديدتين الاساسيتين (1) (2) فحصلنا على ما يلي:

1) نتائج سريعة وصعبة الفك حيث تعتمد على جداء مصفوفات كبيرة الحجم وبالتالي فإن تحليل الشيفرة بهدف كسرها يتطلب زمناً طويلاً وتكلفة كبيرة.
2) قابلية تطبيق هذه الطريقة التشفيرية في أهداف محدودة وصغيرة مثل الشركات المحلية المدارس الجامعات المعامل ذات الإنتاجية المحدودة.

الفصل الثالث

التشفير باستخدام المصفوفات المنتهية المجزأة

مقدمة :

نقدم في هذا الفصل نمطا جديدا من التشفير باستخدام المصفوفات الجزئية المنتهية لتشفير الرسائل المرمزة بنظام ASCII وذلك بتجزئة مصفوفة النص الأصلي إلى عدد محدود من المصفوفات الجزئية المنتهية والعمل على ترميز هذه المصفوفات باستخدام خوارزميات وشروط خاصة لتشفير نص موضوع داخل كل مصفوفة جزئية من مصفوفة النص الأصلي واستخدام عدة مفاتيح للتشفير داخل النص الواضح، للحصول على رسائل مشفرة بأكثر من مفتاح وبأكثر من خوارزمية.

يعتبر علم التشفير من أهم العلوم لحماية المعلومات فهو يهدف إلى الحفاظ على سرية الرسائل المبثوثة عبر قنوات الاتصال المختلفة مثل الجوال والبريد الالكتروني و الفيس بوك... وغيرها ، لذلك يتجه الباحثين إلى دراسة طرق وآليات جديدة للتشفير مع تقدم العلوم. عرضنا في هذا الفصل نمطا جديداً من التشفير للرسائل النصية بتجزئة مصفوفة النص الأصلي إلى عدد محدود من المصفوفات الجزئية المنتهية والعمل على ترميز هذه المصفوفات باستخدام خوارزميات واستخدام شروط خاصة لتشفير نص موضوع داخل كل مصفوفة جزئية من مصفوفة النص الأصلي ، وبالتالي زيادة عدد مفاتيح التشفير داخل الرسالة بحيث نحصل على رسائل مشفرة بأكثر من مفتاح وبأكثر من خوارزمية مما يجعل الشفرة صعبة الكسر و يحافظ على أمن المعلومات داخل النصوص. تم تقسيم الفصل إلى عدة فقرات.

تناولنا في الفقرة الأولى بعض المفاهيم وبعض التعاريف، وفي الفقرة الثانية تم وضع النتائج والبراهين والنظريات (أهم نتائج الفصل) التي تم التوصل إليها وتم وضع خطوات خوارزمية التشفير المقترحة كما تم عرض مثال تطبيقي يشرح أحد التوجهات لاستخدام الخوارزمية بشكل عام وأخيراً تم عرض بعض التوصيات والمقترحات من أجل الأعمال المستقبلية.

تعاريف أساسية: [6] و [9]

1. تعريف 1: المصفوفة الجزئية المنتهية (FMM)

هي مصفوفة مربعة X تقسم إلى عدة مصفوفات جزئية X_i حيث:

$$\bigcap_i X_i = \phi \quad \text{and} \quad \bigcup_i X_i = X$$ [12].

تعريف 2: النظير الضربي لعدد a بالنسبة لمقاس n [7] .

ليكن $a \in Z, n \in N$ حيث $\gcd(a,n) = (a,n) = 1$ عندئذٍ يسمى العدد $b \in Z$ بالنظير الضربي للعدد a إذا حقق الشرط

التالي:

$$a.b \equiv 1 (\mathrm{mod}\, n)$$

أي أن العدد $a.b-1$ يقبل القسمة على العدد n، ويرمز للنظير الضربي للعدد a بالرمز a^{-1}

تعريف 3: النظير الضربي لمصفوفة مربعة P:

هي مصفوفة مربعة P^{-1} تحقق الشرط $P.P^{-1} = I \ mod 256$

تعريف4: علاقة التشفير (Enciphering Key)

نعرف علاقة التشفير بشكل عام بالشكل: $$F(x) = (PX + B) \bmod(256) \quad (4)$$

حيث P مصفوفة قابلة للقلب بالنسبة للمقاس 256 (مثلا المصفوفة الشعاعية أو مصفوفة هل) و المصفوفة B هي مصفوفة انسحاب) ، والمصفوفة X هي المصفوفة العددية الناتجة عن النص الأصلي بعد استبدال المحارف بالأرقام من جدول ASCII

تعريف 5: علاقة فك التشفير (Deciphering Key)

علاقة فك التشفير وهي الدالة العكسية لدالة التشفير ولها الشكل:

$$X = P^{-1}(F - B) \bmod(256) \quad (5)$$

حيث إن: P^{-1} مقلوب المصفوفة P بالنسبة للمقاس 256.

2. أهم نتائج الفصل:

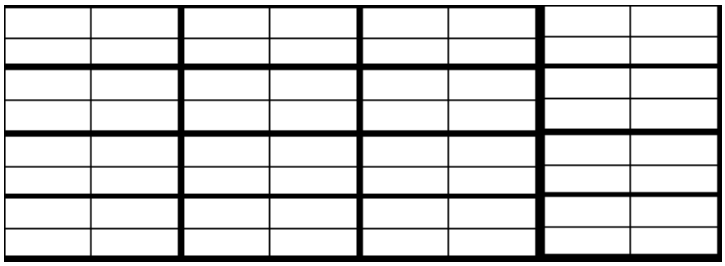
قدم الفصل [12] عدة مبرهنات تبين عدة إمكانات لتجزئة المصفوفات المنتهية إلى عدد محدود من المصفوفات الجزئية و من هذه المبرهنات المبرهنة التالية:

مبرهنة(1) :

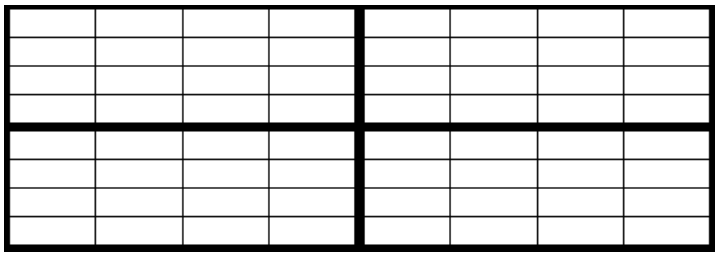
ليكن لدينا مصفوفة مربعة من المرتبة $m*m$ حيث $m = m_1 * m_2$ عندها تقسم المصفوفة المربعة إلى m_1^2 من المصفوفات المربعة الجزئية، وكل منها يحوي m_2^2 عنصر[12].

مثال(1): بفرض لدينا مصفوفة مربعة من المرتبة 8*8 حيث 2*4=8 عندئذٍ:

- يمكن أن نجزأ المصفوفة إلى 16 مربع وكل مربع يحوي 4 عناصر على الشكل التالي:



- أو يمكن أن نجزأ المصفوفة إلى 4 مربعات وكل مربع يحوي 16 عنصر، على الشكل التالي:



وهناك عدة إمكانات للتقسيم يختلف وفقا للخوارزميات التي نريد استخدامها.

عرضنا ضمن هذا الفصل تعميم استخدام المصفوفات المجزاة في التشفير وذلك للوصول إلى شفرة صعبة الكسر ذات درجة أمان عالية بحيث جعلنا آلية التشفير يعتمد على عدة خوارزميات وعدة مفاتيح تطبق على أجزاء مختلفة من النص حيث تشفير كل مصفوفة مجزاة بخوارزمية تشفير معينة بواسطة مفتاح معين بشكل وحيد وفق **المبرهنتين التاليتين:**

مبرهنة(2): إن كل نص واضح T (من الحروف ASCII) ضمن مصفوفة جزئية X_{ij} يشفر وفق الدالة المصفوفية:

$$F_k(x) = (P_k X_k + B_k) \bmod 256, k = 1,2..m \quad (4)$$

بشكلٍ وحيد(P_k مصفوفة قابلة للقلب بالنسبة للمقاس 256 و $X = [X_{ij}]$).

البرهان:

لتكن X_1 , X_2 مصفوفتين من الأعداد المقابلة لنصين مختلفين من الحروف ASCII أي: $X_1 \neq X_2$ ولتكن المصفوفتين B , P الثابتتين، ولنبرهن أن: $F(X_1) \neq F(X_2)$.

$$X_1 \neq X_2 (\bmod 256) \Rightarrow PX_1 \neq PX_2 (\bmod 256) \Rightarrow$$
$$P_1X_1 + B_1 \neq P_2X_2 + B_2 (\bmod 256) \Rightarrow F(X_1) \neq F(X_2)$$

إذا العلاقة (4) يتم التشفير بواسطتها بشكل وحيد.

مبرهنة(3):

إن فك التشفير عن كل نص معمى C (من الحروف ASCII) والموافق لمصفوفة جزئية يتم بشكلٍ وحيد من خلال الدالة المصفوفية:

$$X_k = P_k^{-1}(F_k - B_k) \bmod(256)\ , k = 1,2..m \quad (5)$$

(P_k مصفوفة قابلة للقلب بالنسبة للمقاس 256).

البرهان:

لتكن F_1 , F_2 المصفوفتين العدديتين المقابلتين لشيفرتين مختلفتين من حروف ASCII ولتكن X_1, X_2 المصفوفتين المقابلتين للرسالتين الأصليتين الموافقتين للشيفرتين F_1 , F_2 على الترتيب بواسطة الدالة:

$$X_k = P_k^{-1}(F_k - B_k) \bmod(256)\ , k = 1,2 \quad (5)$$

ولنثبت أن : $X_1 \neq X_2$

لدينا $F_1 \neq F_2$ وبالتالي بسهولة نجد:

$$F_1 - B_1 \neq (F_2 - B_1) \bmod(256) \Rightarrow$$

$$P_1^{-1}(F_1 - B\ ) \neq P_2^{-1}(F_2 - B_2) \bmod(256) \Rightarrow$$

$$X_1 \neq X_2$$

وبالتالي ، فإن فك التشفير عن كل نص مشفر من خلال الدالة (5) يتم بشكلٍ وحيد .

خوارزمية التشفير المقترحة بواسطة المصفوفات المجزاة :

ليكن النص الواضح T ولتشفيره تطبق الخطوات التالية:

1. يتم ترتيب حروف النص في مصفوفة T من المرتبة (m,m) ، عدد أسطرها وعدد أعمدتها m، ومن ثم يتم كتابة m على الشكل: $m = m_1 * m_2$ ويتم تجزيء المصفوفة T إلى مصفوفات جزئية.
2. يتم تبديل المحارف بالأرقام ضمن المصفوفة T وفق جدول ASCII (1). ويتم الحصول على المصفوفة X المجزأة إلى مصفوفات $X_k = [X_{ij}]$.
3. يتم تطبيق علاقة التشفير

$$F(x) = (PX + B) \bmod(256) \qquad (4)$$

بواسطة مفاتيح مختلفة على المصفوفات المجزأة ونحصل على المصفوفة C (مصفوفة النص المشفر) للنص الأصلي.

4. يتم تبديل الأرقام ضمن المصفوفة C بالمحارف من الجدول (1) الفصل الأول ، فيتم الحصول على النص المشفر.

أولاً التشفير المجزأ اعتماداً على مصفوفة هيل ومصفوفة الانسحاب B:

مثال (2) التشفير:

لنشفر الرسالة التالية:

{This paper presents a new approach of Cryptography using FFM }

الحل:

1- عدد المحارف 60 وبالتالي لنأخذ m=16*4=64، ولنشكل مصفوفة المحارف على الشكل التالي:

$$T=\left(\begin{array}{cccc|cccc} T & e & n & w & c & y & h & - \\ h & r & t & - & h & p & y & F \\ i & - & s & a & - & t & - & F \\ s & p & - & p & o & o & u & M \\ \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ \\ - & r & a & p & f & g & s & - \\ p & e & - & r & - & r & i & - \\ a & s & n & o & C & a & n & - \\ p & e & e & a & r & p & g & - \end{array}\right)$$

المصفوفة T قسمت إلى أربع مصفوفات مربعة وكل مصفوفة فيها 16 عنصر، وهذه أحد التقسيمات، وسنقوم بتجزيء المصفوفة السابقة إلى شكل آخر يساعدنا في تطبيق أنواع خاصة من الخوارزميات[11] وفق التالي:

$$T=\left(\begin{array}{cccc|cccc} T & e & n & w & c & y & h & - \\ h & r & t & - & h & p & y & F \\ \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ \\ i & - & s & a & - & t & - & F \\ s & p & - & p & o & o & u & M \\ \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ \\ - & r & a & p & f & g & s & - \\ p & e & - & r & - & r & i & - \\ a & s & n & o & C & a & n & - \\ p & e & e & a & r & p & g & - \end{array}\right)$$

2- يتم تبديل المحارف بالأرقام ضمن المصفوفة T وفق جدول ASCII (1). ويتم الحصول على المصفوفة X:

$$\begin{bmatrix} X1 \\ X2 \\ X3 \quad X4 \end{bmatrix} = \left(\begin{array}{cccc|cccc} 84 & 101 & 110 & 119 & 99 & 121 & 104 & 32 \\ 104 & 114 & 116 & 32 & 104 & 112 & 121 & 70 \\ \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ \\ 105 & 32 & 115 & 97 & 32 & 116 & 32 & 70 \\ 115 & 112 & 32 & 112 & 111 & 111 & 117 & 77 \\ \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ \\ 32 & 114 & 97 & 112 & 102 & 103 & 115 & 32 \\ 112 & 101 & 32 & 114 & 32 & 114 & 105 & 32 \\ 97 & 115 & 110 & 111 & 67 & 97 & 110 & 32 \\ 112 & 101 & 101 & 97 & 114 & 112 & 103 & 32 \end{array}\right)$$

الآن سنعرض أحد إمكانات الحل:

a- تشفير المصفوفة الأولى في الصف الأول بطريقة المصفوفات الشعاعية وهي من نموذج (2*4) [11] بواسطة مفتاح تشفير معين وعلاقة الانسحاب $p1, B1$ كمايلي:

$$P1 = \begin{bmatrix} 7 & 56 \\ 24 & 57 \end{bmatrix}$$

$$B1 = \begin{bmatrix} 3 & 5 & 7 & 9 & 3 & 5 & 7 & 9 \\ 11 & 13 & 15 & 17 & 11 & 13 & 15 & 17 \end{bmatrix}$$

وعلاقة التشفير: (4) ، أما بالنسبة للمصفوفتين الموجودتين في الصف الثالث والتي كل منهما من الدرجة الرابعة، نريد استخدام مصفوفة هيل P (Hill) ولتكن مثلاً المصفوفتين:

$$P1 = H_4 = \begin{pmatrix} 1 & 0 & 1 & 2 \\ 2 & 0 & 1 & 1 \\ 2 & 1 & 2 & 0 \\ 3 & 3 & 2 & 0 \end{pmatrix}; \det(P1) = |P1| = 9, \gcd(9,256) = 1$$ حيث إن :

$$P2 = H_4 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 2 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 3 & 3 & 1 & 1 \end{pmatrix}$$

$$;\det(P2) = |P2| = 3, \gcd(3,256) = 1$$

ومصفوفة الانسحاب ولتكن:

$$B = \begin{pmatrix} 3 & 4 & 0 & 1 \\ 9 & 4 & 7 & 3 \\ 3 & 6 & 6 & 3 \\ 8 & 5 & 0 & 7 \end{pmatrix}$$

وعلاقة التشفير: (4)

3- باستخدام $p1, B1$ وعلاقة التشفير: (4) حيث المصفوفة P هي:

$$P1 = \begin{bmatrix} 7 & 56 \\ 24 & 57 \end{bmatrix}$$

$$B1 = \begin{bmatrix} 3 & 5 & 7 & 9 & 3 & 5 & 7 & 9 \\ 11 & 13 & 15 & 17 & 11 & 13 & 15 & 17 \end{bmatrix}$$

تتطبق على المصفوفة:

$$X_1 = \begin{pmatrix} 84 & 101 & 110 & 119 & 99 & 121 & 104 & 32 \\ 104 & 114 & 116 & 32 & 104 & 112 & 121 & 70 \end{pmatrix}$$

فنحصل على الشيفرة الرقمية الموضحة بالمصفوفة الجزئية التالية:

$$F(X_1) = C_1^* = \begin{pmatrix} 14 & 184 & 109 & 88 & 164 & 46 & 15 & 175 \\ 7 & 217 & 35 & 71 & 111 & 71 & 176 & 149 \end{pmatrix}$$

وبنفس الشكل تماما نطبق العلاقات التالية:

$$p2, B2$$

وعلاقة التشفير: (4) حيث المصفوفة P هي:

$$P2 = \begin{bmatrix} 3 & 12 \\ 4 & 13 \end{bmatrix}$$

$$B2 = \begin{bmatrix} 7 & 11 & 15 & 19 & 7 & 11 & 15 & 19 \\ 23 & 27 & 31 & 35 & 23 & 27 & 31 & 35 \end{bmatrix}$$

على المصفوفة:

$$X_2 = \begin{pmatrix} 105 & 32 & 115 & 97 & 32 & 116 & 32 & 70 \\ 115 & 112 & 32 & 112 & 111 & 111 & 117 & 77 \end{pmatrix}$$

ونحصل على الشيفرة الرقمية الموضحة بالمصفوفة الجزئية التالية:

$$F(X_2) = C_2^* = \begin{pmatrix} 164 & 183 & 56 & 226 & 147 & 143 & 219 & 109 \\ 122 & 47 & 107 & 51 & 34 & 114 & 112 & 0 \end{pmatrix}$$

وأخيراً بتطبيق التشفير باستخدام مصفوفتي هيل:

$$P3 = H_4 = \begin{pmatrix} 1 & 0 & 1 & 2 \\ 2 & 0 & 1 & 1 \\ 2 & 1 & 2 & 0 \\ 3 & 3 & 2 & 0 \end{pmatrix}; \det(P) = |P| = 9, \gcd(9,256) = 1$$

$$P3 = H_4 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 2 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 3 & 3 & 1 & 1 \end{pmatrix}; \det(P) = |P| = 3, \gcd(3,256) = 1$$

ومصفوفة الانسحاب:

$$B = \begin{pmatrix} 3 & 4 & 0 & 1 \\ 9 & 4 & 7 & 3 \\ 3 & 6 & 6 & 3 \\ 8 & 5 & 0 & 7 \end{pmatrix}$$

وعلاقة التشفير (4) على المصفوفتين على الترتيب:

$$X_3 = \begin{pmatrix} 32 & 114 & 97 & 112 \\ 112 & 101 & 32 & 114 \\ 97 & 115 & 110 & 111 \\ 112 & 101 & 101 & 97 \end{pmatrix}, X4 = \begin{pmatrix} 102 & 103 & 115 & 32 \\ 32 & 114 & 105 & 32 \\ 67 & 97 & 110 & 32 \\ 114 & 112 & 103 & 32 \end{pmatrix}$$

يتم الحصول على الشيفرتين الرقميتين المقابلتين:

$$F(X_3) = C^*_3 = \begin{pmatrix} 147 & 219 & 198 & 210 \\ 218 & 24 & 123 & 39 \\ 227 & 208 & 139 & 214 \\ 137 & 98 & 86 & 125 \end{pmatrix},$$

$$F(X_4) = C_4^* = \begin{pmatrix} 144 & 172 & 175 & 129 \\ 138 & 163 & 194 & 131 \\ 117 & 8 & 49 & 163 \\ 32 & 82 & 112 & 7 \end{pmatrix}$$

وتكون المصفوفة الرقمية التي تصف الشفرة النهائية للنص:

{This paper presents a new approach of Cryptography using FFM **}**

لها بالشكل:

$$F(X) = C^* = \begin{pmatrix} F(X_1) \\ F(X_2) \\ F(X_3) \quad F(X_4) \end{pmatrix} =$$

أي تأخذ الشكل:

$$\left(\begin{array}{c}
\begin{pmatrix} 14 & 184 & 109 & 88 & 164 & 46 & 15 & 175 \\ 7 & 217 & 35 & 71 & 111 & 71 & 176 & 149 \end{pmatrix} \\
\hline
\begin{pmatrix} 164 & 183 & 56 & 226 & 147 & 143 & 219 & 109 \\ 122 & 47 & 107 & 51 & 34 & 114 & 112 & 0 \end{pmatrix} \\
\hline
\left(\begin{array}{cccc|cccc} 147 & 219 & 198 & 210 & 144 & 172 & 175 & 129 \\ 218 & 24 & 123 & 39 & 138 & 163 & 194 & 131 \\ 227 & 208 & 139 & 214 & 117 & 8 & 49 & 163 \\ 137 & 98 & 86 & 125 & 32 & 82 & 112 & 7 \end{array}\right)
\end{array}\right)$$

4- نقوم بتبديل الأرقام ضمن المصفوفة C^* بالمحارف من الجدول (1) فيتم الحصول على النص المشفر الكلي الموضح بالمصفوفة:

$$C = \begin{pmatrix}
\text{'14'} & ¸ & m & X & ¤ & . & & ¯ \\
 & ظ & \# & G & o & G & ° & • \\
¤ & · & 8 & â & “ & & غڈ & m \\
z & / & k & 3 & " & r & p & \text{'0'} \\
" & غ & ئ & ز & گ & \text{'172'} & ¯ & پ \\
ع & \text{'24'} & \{ & ' & ٹ & £ & آ & f \\
م & ذ & › & ض & u & \text{'8'} & 1 & £ \\
‰ & b & V & } & \text{'32'} & R & p & \text{'7'}
\end{pmatrix}$$

ملاحظة 1:

هناك محارف غير ظاهرة في الشفرة لأنها محجوبة عن الظهور لذلك يترك رقمياً كما هو ضمن اشارتي تنصيص مثل العدد "14" في السطر الأول والعدد "0" في السطر الرابع والعدد "172" ...الخ.

هذه الشيفرة لايمكن فكها مالم يعرف ماهي الخوارزميات وكلمات السر المستخدمة وماهي آلية تجزيء المصفوفة وكل جزء بأي خوارزمية تم تشفيره، وبالتالي هذه الطريقة تؤدي إلى زيادة درجة الأمان للرسائل النصية بشكل كبير بحيث لايمكن الوصول للنص الأصلي من الشيفرة بالطرق الكلاسيكية ومحاولات التجريب.

ملاحظة 2:

لفك النص المشفر السابق نقوم بخوارزمية معاكسة أي أولاً نجزأ المصفوفة الرئيسية إلى أربع مصفوفات

$$F(X)=\begin{pmatrix} F(X_1) \\ F(X_2) \\ F(X_3) \quad F(X_4) \end{pmatrix}$$

وبتطبيق علاقة فك التشفير

$X = P^{-1}(F-B)\bmod(256)$ (5) ، على كل مصفوفة مجزأة وفقا لكل خوارزمية تشفير مطبقة، فنحصل على مصفوفة الأرقام X والمقابلة لأحرف الرسالة الأصلية ضمن مصفوفة ويتم قراءة الرسالة عمود بعد استبدال الأرقام بالمحارف من جدول الآسكي (1). ونحصل على الرسالة الأصلية ذاتها.

مثال (3) فك التشفير:

يتم بطريقة معاكسة لعملية التشفير أي بعد تقسيم النص إلى المصفوفات المتفق عليها سابقاً وهي /6/ مصفوفات نطبق علاقات التشفير العكسي عليها.

1) نطبق على مصفوفتي السطر الأول نطبق العلاقة:

$$X1 = P_1^{-1}(F(X1) - B1)mod256$$

2) ونطبق على مصفوفتي السطر الثاني العلاقة العكسية:

$$X2 = P_2^{-1}(F(X2) - B2)mod256$$

3) بينما نطبق على السطر الثالث العلاقتين:

$$X3 = P_3^{-1}(F(X3) - B)mod256$$

$$X4 = P_4^{-1}(F(X4) - B)mod256$$

فنعود إلى النص الأصلي رقمياً.

$$X=\left(\begin{array}{cccc|cccc} 84 & 101 & 110 & 119 & 99 & 121 & 104 & 32 \\ 104 & 114 & 116 & 32 & 104 & 112 & 121 & 70 \\ \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ \\ 105 & 32 & 115 & 97 & 32 & 116 & 32 & 70 \\ 115 & 112 & 32 & 112 & 111 & 111 & 117 & 77 \\ \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ \\ 32 & 114 & 97 & 112 & 102 & 103 & 115 & 32 \\ 112 & 101 & 32 & 114 & 32 & 114 & 105 & 32 \\ 97 & 115 & 110 & 111 & 67 & 97 & 110 & 32 \\ 112 & 101 & 101 & 97 & 114 & 112 & 103 & 32 \end{array}\right)$$

وبتبديل المحارف بالأرقام نعود للنص الأصلي وهو :

$$T=\left(\begin{array}{cccc|cccc} T & e & n & w & c & y & h & - \\ h & r & t & - & h & p & y & F \\ i & - & s & a & - & t & - & F \\ s & p & - & p & o & o & u & M \\ \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ & \_\_ \\ - & r & a & p & f & g & s & - \\ p & e & - & r & - & r & i & - \\ a & s & n & o & C & a & n & - \\ p & e & e & a & r & p & g & - \end{array}\right)$$

أي أن النص الأصلي هو :

{This paper presents a new approach of Cryptography using FFM **}**

ثانياً التشفير المجزأ وفق المصفوفات الشعاعية:

الآن سنعطي مثالاً للتشفير وفك التشفير وفق المصفوفات المجزأة باستخدام المصفوفات الشعاعية (المولدة بشعاع) V.

مثال (4) التشفير:

لنشفر الكلمة التالية: **{**Informations**}**

الحل:

1- عدد المحارف 12 وبالتالي لنأخذ m=3*4=12، فتعطي ثلاث مصفوفات وكل مصفوفة تحوي 4/ / عناصر ولنشكل مصفوفة المحارف على الشكل التالي:

$$T=\left[\begin{bmatrix} I & f \\ n & o \end{bmatrix},\begin{bmatrix} r & a \\ m & t \end{bmatrix},\begin{bmatrix} i & n \\ o & s \end{bmatrix}\right]$$

المصفوفة T قسمت إلى ثلاث مصفوفات مربعة وكل مصفوفة فيها 4 عناصر.

2- يتم تبديل المحارف بالأرقام ضمن المصفوفة T وفق جدول ASCII (1).

ويتم الحصول على المصفوفة X:

$$X = \left[X1 = \begin{bmatrix} 73 & 102 \\ 110 & 111 \end{bmatrix}, X2 = \begin{bmatrix} 114 & 97 \\ 109 & 116 \end{bmatrix}, X3 = \begin{bmatrix} 73 & 110 \\ 111 & 115 \end{bmatrix}\right]$$

الآن سنعرض أحد إمكانات الحل:

نشفر المصفوفة الأولى $X1$ بطريقة المصفوفات الشعاعية وفق العلاقة $F(X1) = A.X1 + (B1\ mod256)$

الشعاع V=(219,143) فنحصل على مصفوفة مثلثية سفلى A ومصفوفة انسحاب B كما يلي:

$$A = \begin{bmatrix} 219 & 0 \\ 219 & 143 \end{bmatrix}, B1 = \begin{bmatrix} 213 & 32 \\ 212 & 31 \end{bmatrix}$$

حيث إن:

$$F(X1) = A.X1 + B1\ mod256$$

فيكون ناتج المصفوفة المشفرة وفق $mod256$ هو:

$$F(X1) = \begin{bmatrix} 72 & 98 \\ 185 & 98 \end{bmatrix}$$

3- لتشفير المصفوفة الثانية X2 بطريقة أفاين مثلاً بواسطة مفتاح تشفير معين وعلاقة الانسحاب:

$$F(X2) = a * X2 + b\ mod256 = [ax_{ij} + b] mod256$$

حيث إن:

$$a = 231, b = 61$$

يكون ناتج المصفوفة المشفرة وفق $mod256$ هو:

$$\begin{bmatrix} 17 & 4 \\ 224 & 139 \end{bmatrix}$$

4- تشفير المصفوفة الثالثة X3 بطريقة قيصر بواسطة مفتاح تشفير معين:

$$F(X3) = X3 + b1\ mod256 = [x_{ij} + b] mod256$$

حيث إن:

$$b1 = 243$$

يكون ناتج المصفوفة المشفرة وفق $mod256$ هو:

$$\begin{bmatrix} 92 & 97 \\ 98 & 102 \end{bmatrix}$$

فيكون لدينا المصفوفة المشفرة الكلية:

$$C = \left[\begin{bmatrix} 72 & 98 \\ 185 & 98 \end{bmatrix}, \begin{bmatrix} 17 & 4 \\ 224 & 139 \end{bmatrix}, \begin{bmatrix} 92 & 97 \\ 98 & 102 \end{bmatrix}\right]$$

يتم تبديل المحارف بالأرقام ضمن المصفوفة C وفق جدول ASCII (1):

$$C = \left[\begin{bmatrix} H & b \\ 1 & b \end{bmatrix}, \begin{bmatrix} '17' & '4' \\ \text{à} & \text{›} \end{bmatrix}, \begin{bmatrix} \backslash & a \\ b & f \end{bmatrix}\right]$$

يكون الناتج:

{H¹bb′17′à′4′\›baf}

مع ملاحظة أن العددين 17 و4 ليس لهما مقابل محرفي ظاهري لذلك تركا رقميا ضمن اشارتي تنصيص.

مثال (5) فك التشفير:

لفك تشفير الكلمة السابقة نعود بالعكس أي ننطلق من الرسالة المشفرة وهي:

{H¹bb′17′à′4′\›baf}

وفق الخطوات التالية:

1- إن عدد المحارف 12 وبالتالي لنأخذ m=6*2=12، ولنشكل مصفوفة المحارف على الشكل التالي:

$$C = \left[\begin{bmatrix} H & b \\ 1 & b \end{bmatrix}, \begin{bmatrix} '17' & '4' \\ \text{à} & \text{›} \end{bmatrix}, \begin{bmatrix} \backslash & a \\ b & f \end{bmatrix}\right]$$

المصفوفة C قسمت إلى ثلاث مصفوفات مربعة وكل مصفوفة فيها 4 عناصر.

2- يتم تبديل المحارف بالأرقام ضمن المصفوفة C وفق جدول ASCII (1).

ويتم الحصول على المصفوفة X:

$$F(X) = \left[F(X1) = \begin{bmatrix} 72 & 98 \\ 185 & 98 \end{bmatrix}, F(\,X2) = \begin{bmatrix} 17 & 4 \\ 224 & 139 \end{bmatrix}, F(X3) = \begin{bmatrix} 92 & 97 \\ 98 & 102 \end{bmatrix}\right]$$

الآن سنعرض الحل:

3- فك تشفير المصفوفة الأولى X1 بطريقة المصفوفات الشعاعية العكسية (11)

بواسطة مفتاح التشفير العكسي وعلاقة الانسحاب التاليين:

$$\left[A^{-1} = \begin{bmatrix} 83 & 0 \\ 145 & 111 \end{bmatrix}, B = \begin{bmatrix} 213 & 32 \\ 212 & 31 \end{bmatrix}\right]$$

حيث إن:

$$X1 = A^{-1} * (F(X1) - B)\ mod 256$$

يكون ناتج مصفوفة النص الأصلي وفق $mod256$ هو:

$$X1 = \begin{bmatrix} 73 & 102 \\ 110 & 111 \end{bmatrix}$$

4- فك تشفير المصفوفة الثانية X2 بطريقة أفاين بواسطة مقلوب مفتاح تشفير معين وعلاقة الانسحاب:

$$X2 = a^{-1}.(F(X2) - b)\ mod256$$

حيث إن:

$$a^{-1} = 21, b = 61$$

من جدول النظير الضربي (1)

يكون ناتج مصفوفة النص الأصلي وفق $mod256$ هو:

$$X2 = \begin{bmatrix} 114 & 97 \\ 109 & 116 \end{bmatrix}$$

5- فك تشفير المصفوفة الثالثة X3 بطريقة قيصر بواسطة مفتاح تشفير معين:

$$X3 = F(X3) - b\ mod256$$

حيث إن:

$$b = 243$$

يكون ناتج مصفوفة النص الأصلي وفق $mod256$ هو:

$$\begin{bmatrix} 73 & 110 \\ 111 & 115 \end{bmatrix}$$

فيكون لدينا مصفوفة النص الأصلي الكلية:

$$X = \left[\begin{bmatrix} 73 & 102 \\ 110 & 111 \end{bmatrix}, \begin{bmatrix} 114 & 97 \\ 109 & 116 \end{bmatrix}, \begin{bmatrix} 73 & 110 \\ 111 & 115 \end{bmatrix}\right]$$

وبتبديل المحارف بالأرقام ضمن المصفوفة C وفق جدول ASCII (1):

$$T = \left[\begin{bmatrix} I & f \\ n & o \end{bmatrix}, \begin{bmatrix} r & a \\ m & t \end{bmatrix}, \begin{bmatrix} i & n \\ o & s \end{bmatrix}\right]$$

وبالتالي يكون النص الأصلي: {Informations}

ملخص الفصل الثالث:

لقد قمنا في هذا الفصل بالتشفير باستخدام المصفوفات المجزأة معتمدين على المبرهنات الجديدة (1)،(2)،(3) أي تقسيم النص الأصلي إلى مجموعة نصوص واستخدمنا نمطين من التشفير الهجين .

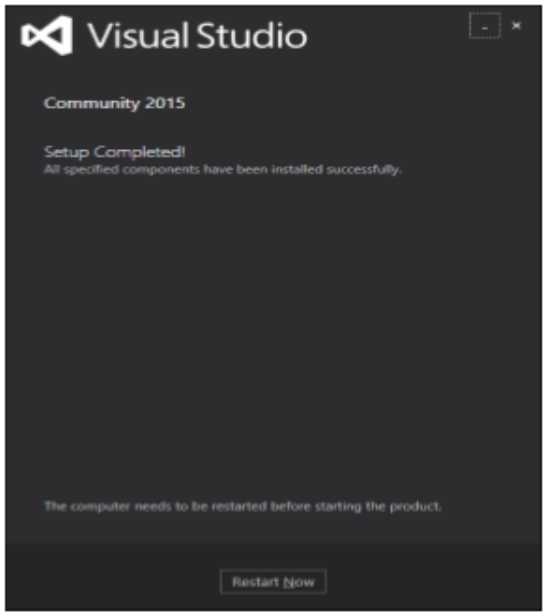
1) *نمط تشفير هيل الذي يتغير بتغير مصفوفة هيل ومصفوفة الانسحاب مثال(1)،(2) أي بتغير :* $pi, Bi : i = 1,2, \dots$
2) *نمط تشفير المصفوفة الشعاعية للشعاع الخاص V الذي يتغير بتغير المصفوفة الشعاعية* A_V *أو* B_V *كما في المثاليين (3)،(4).*
3) *يمكن سرد أمثلة كثيرة لتشفير الهجين حيث تتطبق على المصفوفات المجزأة ، فتارةً نستخدم المصفوفة الشعاعية* A_V *وأخرى تستخدم مصفوفة هيل ومصفوفة الانسحاب* pi, Bi *كما سنراه بالفصل الرابع.*

الفصل الرابع

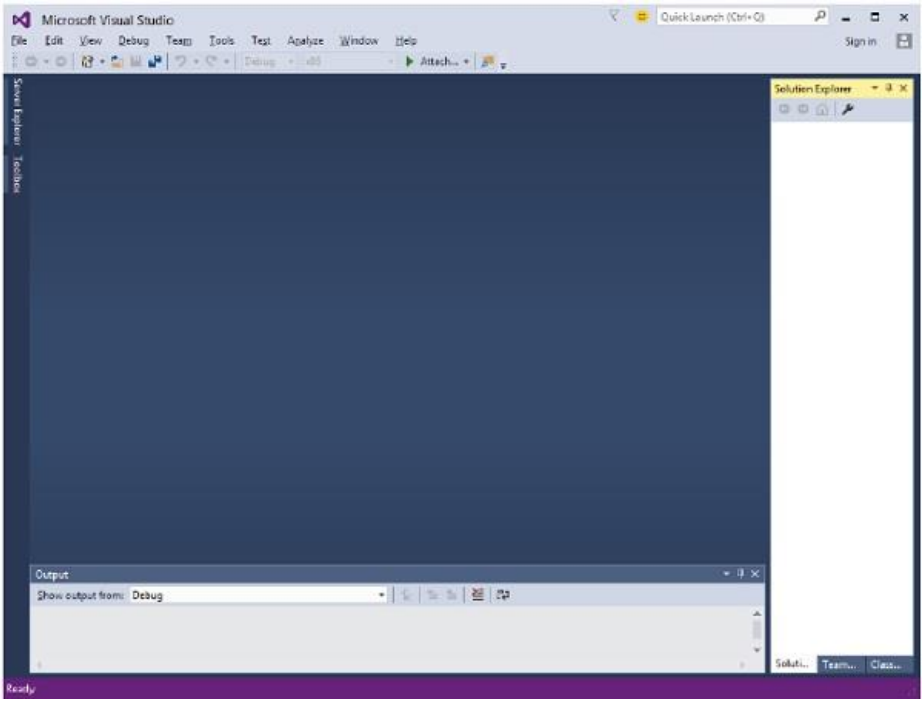
مدخل إلى لغة C#

مقدمة:

لغة C# وتلفظ (سي شارب) هي لغة برمجة غرضية التوجه بسيطة وحديثة تستخدم لأغراض متعددة، تم تطويرها من قبل شركة مايكروسوفت في بيئة دوت نت (.NET). عند تنصيب (بيئة .NET) تظهر لنا الواجهة التالية:



ثم تظهر الصفحة الرئيسية لـ (Microsoft Visual Studio).



الأسباب التالية تجعل من لغة #C واسعة الاستخدام:

- لغة برمجة حديثة ومتعددة الاستخدامات.
- لغة غرضية التوجه (Object Oriented).
- سهلة التعلم.
- تنتج برامج فعالة.
- متوافقة مع مجموعات متنوعة من منصات الكمبيوتر.
- تستخدم في تطبيقات الويب.
- هي جزء من بيئة دوت نت(.NET).

بيئة دوت نت (The .Net Framework):

هي منصة تساعد على كتابة الأنواع التالية من التطبيقات:

- تطبيقات الويندوز (Windows applications).
- تطبيقات الويب (Web applications).
- خدمات الويب (Web services).

تطبيقات .NET Framework. هي تطبيقات متعددة المنصات وقد تم تصميمها بحيث يمكن استخدامها من أي من اللغات التالية (#C،C++، Visual Basic، Jscript).

أنواع البيانات (Data Type) في لغة #C:

نوع	الوصف	المدى
bool	Boolean value	True or False
byte	8-bit unsigned integer	0 to 255
char	16-bit Unicode character	U +0000 to U +ffff
decimal	128-bit precise decimal values with 28-29 significant digits	$(-7.9 \times 10^{28}$ to $7.9 \times 10^{28}) / 10^{0 \text{ to } 28}$
double	64-bit double-precision floating point type	$(+/-)5.0 \times 10^{-324}$ to $(+/-)1.7 \times 10^{308}$
float	32-bit single-precision floating point type	-3.4×10^{38} to $+ 3.4 \times 10^{38}$

int	32-bit signed integer type	-2,147,483,648 to 2,147,483,647
long	64-bit signed integer type	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
sbyte	8-bit signed integer type	-128 to 127
short	16-bit signed integer type	-32,768 to 32,767
uint	32-bit unsigned integer type	0 to 4,294,967,295
ulong	64-bit unsigned integer type	0 to 18,446,744,073,709,551,615
ushort	16-bit unsigned integer type	0 to 65,535

المتحولات (Variables) في لغة #C:

المتحول هو عبارة عن الاسم الذي يطلق على الموقع المحجوز في الذاكرة، ليتم التحكم به من قبل برنامجنا. كل متحول في لغة #C له نوع معين، الذي يحدد حجمه في الذاكرة.

التعليمة البرمجية لتعريف متحول في لغة #C هي:

<data_type><variable_name>;

أمثلة:

```
int d = 3, f = 5;
byte z = 22;
double pi = 3.14159;
char x = 'x';
string = 'Bill';
```

المعاملات الحسابية والمنطقية (Operators) في لغة #C:

المعامل (operator) هو رمز خاص يخبر المترجم (Compiler) لإنجاز عمل ما مثل العمليات المنطقية والحسابية. لغة #C غنية بالكثير من المعاملات ويوجد عدة أنواع منها:

- معاملات حسابية (Arithmetic Operators)
- معاملات المقارنة (Relational Operators)
- معاملات منطقية (Logical Operators)
- معاملات الاسناد (Assignment Operators)

معاملات حسابية (Arithmetic Operators):

الجدول التالي يبين المعاملات الحسابية ووصف لاستخدامها حيث أن (B=20) (A=10)

المعامل	الوصف	مثال
+	جمع رقمين	A + B = 30
−	يستخدم لطرح الرقم الثاني من الأول	A – B = −10
*	يستخدم لعملية الضرب	A * B = 200
/	يستخدم لعملية القسمة	B / A = 2
%	يستخدم لإيجاد باقي القسمة	B % A = 0
++	زيادة بمقدار واحد	A++ = 11
−−	نقصان بمقدار واحد	A−− = 9

معاملات المقارنة (Relational Operators):

الجدول التالي يبين معاملات المقارنة ووصف لاستخدامها حيث أن (B=20) (A=10)

المعامل	الوصف	مثال
==	يتم المقارنة بين قيمتين إذا كانتا متساويتين او لا، في حال كانتا متساويتين يعيد قيمة صحيحة(true)	(A==B) is not true.
=!	يتم المقارنة بين قيمتين إذا كانتا متساويتين او لا، في حال كانتا غير متساويتين يعيد قيمة صحيحة (true)	(A!=B) is true.

>	يتم التحقق ما إذا كانت القيمة التي على يسار المعامل أكبر من القيمة التي على يمين المعامل	(A > B) is not true.
<	يتم التحقق ما إذا كانت القيمة التي على يسار المعامل أصغر من القيمة التي على يمين المعامل	(A < B) is true.
>=	يتم التحقق ما إذا كانت القيمة التي على يسار المعامل أكبر أو تساوي القيمة التي على يمين المعامل	(A >= B) is not true.
<=	يتم التحقق ما إذا كانت القيمة التي على يسار المعامل أصغر أو تساوي القيمة التي على يمين المعامل	(A <= B) is true.

المعاملات المنطقية (Logical Operators):

الجدول التالي يبين المعاملات المنطقية (AND, OR, NOT) ووصف لاستخدامها حيث أن (A=true) (B=false)

المعامل	الوصف	مثال
&&	يسمى بالمعامل (AND) إذا كانت القمتين تعيد كل منهما صحيح(true) فان الناتج صحيح(true)	(A && B) is false.
||	يسمى بالمعامل (OR) إذا كانت إحدى القمتين صحيح (true) فيعيد الناتج صحيح (true)	(A || B) is true.
!	يسمى بالمعامل (NOT) وهو يعكس الصحيح (true) ليصبح خطأ (false) والخطأ ليصبح صحيح (true)	! (A && B) is true.

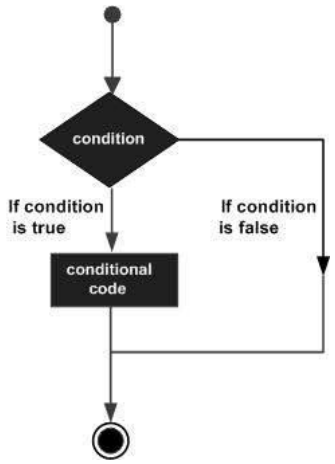
معاملات الإسناد (Assignment Operators):

الجدول التالي يبين معاملات الاسناد ووصف لاستخدامها في لغة C#:

المعامل	الوصف	مثال
=	يستخدم لإسناد قيمة إلى متحول	A = 30
+=	يستخدم لإضافة القيمة التي على يمين المعامل إلى القيمة التي على يسار المعامل وإسناد الناتج إلى قيمة المتحول الذي على اليسار	C += A is equivalent to C = C + A
−=	يستخدم لطرح القيمة التي على يمين المعامل من القيمة التي على يسار المعامل وإسناد الناتج إلى قيمة المتحول الذي على اليسار	C −= A is equivalent to C = C – A
*=	يستخدم لضرب القيمة التي على يمين المعامل مع القيمة التي على يسار المعامل وإسناد الناتج إلى قيمة المتحول الذي على اليسار	C *= A is equivalent to C = C * A
/=	يستخدم لقسمة القيمة التي على يسار المعامل على القيمة التي على يمين المعامل وإسناد الناتج إلى قيمة المتحول الذي على اليسار	C /= A is equivalent to C = C / A
%=	يستخدم لإيجاد باقي قسمة قيمتين وإسناد الناتج إلى قيمة المتحول الذي على اليسار	C %= A is equivalent to C = C % A

الجمل الشرطية (Decision Making) في لغة C#:

هيكلية الجمل الشرطية تتطلب من المبرمج تحديد شرط واحد أو أكثر، ليتم تقييمها أو اختبارها من قبل البرنامج، جنبا إلى جنب مع الجملة الشرطية، ليتم تنفيذها إذا كان الشرط هو الصحيح يتم تنفيذه وفي حال لم يكن اما أن يخرج أو ينفذ الخيار الافتراضي (Default) إن وجد.المخطط العام للجمل الشرطية في معظم لغات البرمجة:



- تبدأ وتنتهي الجملة البرمجية الشرطية بأقواس {Statement} if(Condition)
- تعليمة الكتابة: Console.WriteLine();
- تعليمة القراءة: Console.ReadLine();
- {0}: يحجز مكان داخل الذاكرة حتى يضع فيه قيمه المتغير الذى يليه في الترتيب فقط.

الجمل الشرطية	الوصف	مثال

جملة if	جملة شرطية أحادية إذا تحقق الشرط أي أعاد قيمة صحيحة يتم تنفيذه	int a = 5;
		if (a <10)
		{
		Console.WriteLine("a is less than10");
		}
		Console.ReadLine();
		يكون الناتج: (a is less than10)

جملة if...else	جملة شرطية تحتوي على أكثر من شرط يتم تنفيذ أول شرط صحيح، يتم تنفيذه فقط ويخرج من الجملة	int a = 100;
		if (a < 20)
		{
		Console.WriteLine("a is less than 20");
		}
		else
		{

Console.WriteLine("a is not less than 20"); } Console.ReadLine(); يكون الناتج: (a is not less than 20)		

int a = 100;
int b = 200;
if (a == 100)
{
if (b == 200)
{
Console.WriteLine("Value of a is 100 and b is 200");
}
}
Console.WriteLine("Exact value of a is : {0}", a);
Console.WriteLine("Exact value of b is : {0}", b);
Console.ReadLine();
يكون الناتج:
Value of a is 100 and b is 200
Exact value of a is : 100
Exact value of b is : 200
	جملة شرطية تحتوي على جملة شرطية أخرى ضمن جملة شرطية في حال كان الشرط صحيح يتم الانتقال الى الشرط الآخر وتنفيذه	جملة nested if

char grade = 'B'; switch (grade) { case 'A': Console.WriteLine("Excellent!"); break;	جملة شرطية يتم تنفيذ الشرط عن طريق اختبار المتحول مع عدد من القيم	جملة switch

case 'B': case 'C': Console.WriteLine("Well done"); break; case 'D': Console.WriteLine("You passed"); break; case 'F': Console.WriteLine("Better try again"); break; default: Console.WriteLine("Invalid grade"); break; } Console.WriteLine("Your grade is {0}", grade); Console.ReadLine(); يكون الناتج: Well done Your grade is B		

int a = 100;
int b = 200;
switch (a)
{
case 100:
Console.WriteLine("This is part of outer
switch ");
switch (b)
{
case 200:
	جملة شرطية كالسابق ولكن تحتوي على جملة من نوعها او نوعif	جملة nested switch

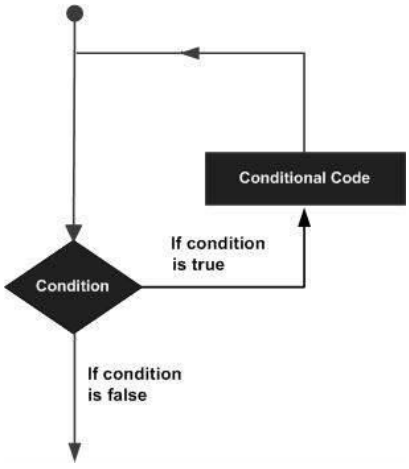
```
Console.WriteLine("This is part of inner
switch ");
break;
}
break;
}
Console.WriteLine("Exact value of a is :
{0}", a);
Console.WriteLine("Exact value of b is :
{0}", b);
Console.ReadLine();
```

يكون الناتج:

```
This is part of outer switch
This is part of inner switch
Exact value of a is : 100
Exact value of b is : 200
```

جمل التكرار (Loops) في لغة #C:

هناك بعض الحالات التي تتطلب تنفيذ جملة برمجية لعدد من المرات وبشكل متكرر لذلك لغات البرمجة ومنها لغة #C تقدم عدد من جمل التكرار، المخطط التالي لجمل التكرار في معظم لغات البرمجة:



جمل التكرار	الوصف	مثال

while loop	تستخدم لتكرار تعليمة برمجية أو مجموعة تعليمات برمجية عندما يكون الشرط محقق. تقوم بالتحقق من الشرط قبل البدء بتكرار التعليمة.	int a = 10;
		while (a < 20)
		{
		Console.WriteLine("value of a: {0}", a);
		a++;
		}
		Console.ReadLine();
		يكون الناتج:
		value of a: 10
		value of a: 11
		value of a: 12
		value of a: 13
		value of a: 14
		value of a: 15
		value of a: 16
		value of a: 17
		value of a: 18
		value of a: 19

for loop	تقوم بتكرار التعليمة البرمجية المعطاة عدد من المرات وفقا للشرط حتى يصبح قيمته خطأ(false)	for (int a = 10; a < 20; a = a + 1)
		{
		Console.WriteLine("value of a: {0}", a);
		}
		Console.ReadLine();
		يكون الناتج:
		value of a: 10
		value of a: 11

value of a: 12 value of a: 13 value of a: 14 value of a: 15 value of a: 16 value of a: 17 value of a: 18 value of a: 19		

int a = 10;
do
{
Console.WriteLine("value of a:
{0}", a);
a = a + 1;
}
while (a < 20);
Console.ReadLine();
يكون الناتج:
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15
value of a: 16
value of a: 17
value of a: 18
value of a: 19
	تستخدم لتكرار تعليمة برمجية أو مجموعة تعليمات برمجية عندما يكون الشرط محقق.
	تقوم بالتحقق من الشرط في نهاية التعليمة البرمجية.
		do…while loop

int i, j;	يمكن استخدام جملة	nested

loops	تكرارية أو أكثر داخل جملة تكرارية أخرى	for (i = 2; i < 100; i++)
		{
		for (j = 2; j <= (i / j); j++)
		if ((i % j) == 0) break;
		if (j > (i / j))
		Console.WriteLine("{0} is prime",
		i);
		}
		Console.ReadLine();
		يكون الناتج:
		2 is prime
		3 is prime
		5 is prime
		7 is prime
		11 is prime
		13 is prime
		17 is prime
		19 is prime
		23 is prime
		29 is prime
		31 is prime
		37 is prime
		41 is prime
		43 is prime
		47 is prime
		53 is prime
		59 is prime
		61 is prime
		67 is prime
		71 is prime

73 is prime 79 is prime 83 is prime 89 is prime 97 is prime		

جمل التحكم بحلقات التكرار (Loop Control Statements):

جمل التحكم	الوصف	مثال

break	تستخدم لتوقيف تنفيذ الحلقات التكرارية.عند استخدامها يتم توقف تنفيذ الجملة البرمجية والانتقال مباشرة لتنفيذ الجملة البرمجية التالية.	int a = 10;
		while (a < 20)
		{
		Console.WriteLine("value of a: {0}",
		a);
		a++;
		if (a > 15)
		{
		break;
		}
		}
		Console.ReadLine();
		يكون الناتج:
		value of a: 10
		value of a: 11
		value of a: 12
		value of a: 13
		value of a: 14
		value of a: 15

continue	تستخدم لتخطي جملة برمجية معينة ثم الاستمرار	int a = 10; do

	لتكملة الجملة البرمجية التي تليها او حلقة التكرار التي تليها.	{
		if (a == 15)
		{
		a = a + 1;
		continue;
		}
		Console.WriteLine("value of a: {0}",
		a);
		a++;
		}
		while (a < 20);
		Console.ReadLine();
		يكون الناتج:
		value of a: 10
		value of a: 11
		value of a: 12
		value of a: 13
		value of a: 14
		value of a: 16
		value of a: 17
		value of a: 18
		value of a: 19

الدوال (Methods / Functions) في لغة #C:

الدالة هي مجموعة من التعليمات البرمجية تقوم بتنفيذ مهمة محددة. والهدف من استخدام الدوال في البرمجة:

- استخدام الدالة أكثر من مرة
- سهولة كشف الأخطاء
- تنظيم التعليمات البرمجية

- اختصار التعليمات البرمجية

الشكل العام للدوال في لغة #C:

```
<Access Specifier><Return Type><Method Name>
(Parameter List)
{
  Method Body
}
```

حيث إن:

- Access Specifier: محدد الوصول (عام, خاص)
- Return Type: نوع القيمة التي سوف تعيدها الدالة (integer, string)
- Method Name: اسم الدالة
- Parameter: معامل متغير
- { تعني بداية الجملة البرمجية للدالة
- } تعني نهاية الجملة البرمجية للدالة

لاستخدام الدوال نحتاج إلى:

- تعريف الدالة

المثال التالي عبارة عن دالة تأخذ قيميتين لأعداد صحيحة وتعيد القيمة الأكبر.

```
public int FindMax(int num1,
int num2)
  {
    int result;
    if (num1 > num2)
      result = num1;
    else
      result = num2;
    return result;
  }
```

- استدعاء الدالة

يتم استدعاء الدالة عن طريق اسم الدالة.

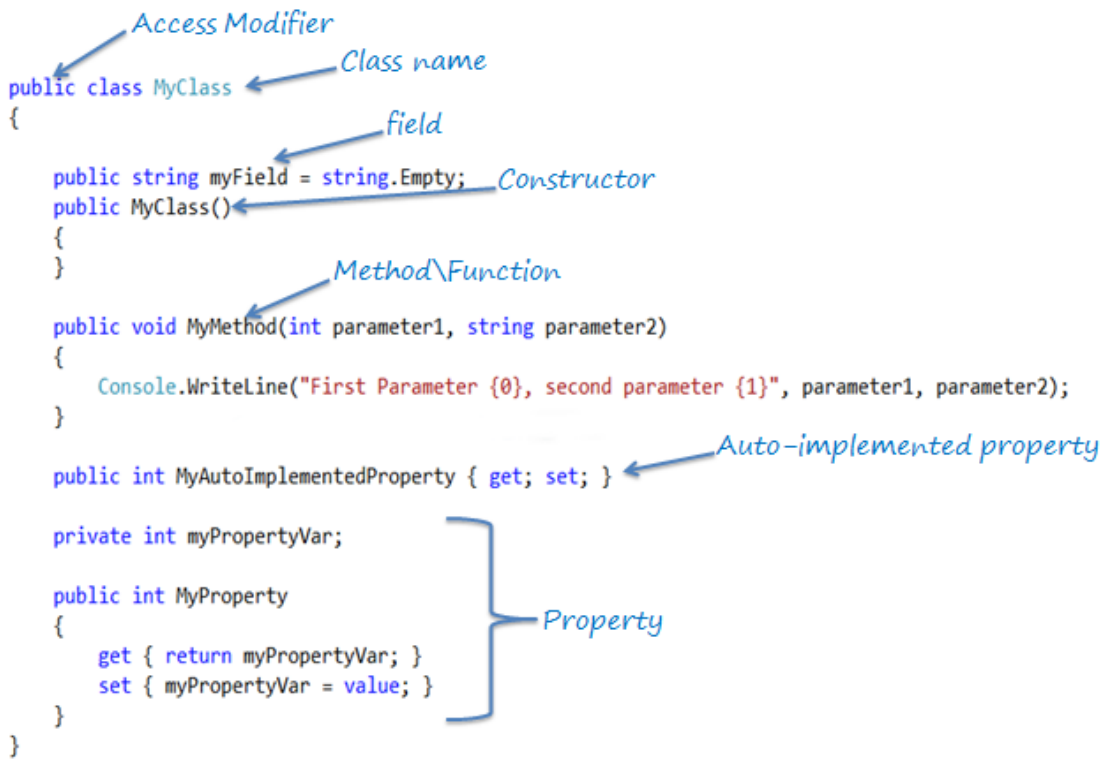
```
int a = 100;
int b = 200;
int ret;
ret = FindMax(a,b);
Console.WriteLine("Max value is : {0}", ret );
Console.ReadLine();
```

يكون الناتج:

```
Max value is: 200
```

الصفوف (Classes) في لغة #C:

الصفوف هي أساس البرمجة غرضية التوجه Object Oriented Programming التي تهدف إلى تضمين المعطيات والدوال ضمن مكونات تسمى الصفوف، نسمي أي اشتقاق من الصف الكائن (Object).الشكل التالي يبين أهم أعضاء الصف (Class) في لغة C#



حيث إن:

- Access Modifier: محددات الوصول تطبق على الصف والدوال والخصائص والأعضاء الأخرين الموجودين بالصف.
- Field: عبارة عن متغير له قيمة.
- Constructor: هو عبارة عن دالة تحمل اسم الصف (Class) نفسه، حيث يتم تنفيذه تلقائيا عند تعريف غرض (Instance) من الصف.
- Method: عبارة عن دالة ضمن الصف (class)
- Property: تغليف الخاصية تكون خاصة (Private) تستخدم (get{}) لاستعادة قيمة المتغير وتستخدم (set{}) لإعطاء قيمة للمتغير.
- Auto-implemented Property: تستخدم هذه الخاصية بشكل مختصر اذا لم نرغب إضافة تعليمات برمجية منطقية الى الخاصية.

الوراثة (Inheritance) في لغة #C:

الوراثة هي أحد المفاهيم الأساسية في البرمجة غرضية التوجه وهي تعني وراثة صف (class)لصف آخر فهي تسهل عملية بناء وصيانة التطبيقات. عندما يقوم المستخدم بإنشاء صف جديد فبدلا من إعادة كتابة عناصر وأعضاء (Properties،Methods) الصف الجديد المشتق (Derived class) يتم وراثتها من الصف الأساسي (Base Class). المثال التالي نفترض ان الصف الأساسي هو الشكل وان الصف المورث هو المستطيل:

```
class Shape
  {
    public void setWidth(int w)
    {
      width = w;
    }
    public void setHeight(int h)
    {
      height = h;
    }
    protected int width;
    protected int height;
  }
```

الصف المشتق:

```
class Rectangle: Shape
{
   public int getArea()
   {
      return (width * height);
   }
}
```

تنفيذ البرنامج:

```
class RectangleTester
  {
     static void Main(string[] args)
     {
        Rectangle Rect = new Rectangle();
        Rect.setWidth(5);
        Rect.setHeight(7);
        Console.WriteLine("Total area: {0}",  Rect.getArea());
        Console.ReadKey();
     }
  }
```

يكون الناتج: Total area: 35

الواجهات (Interfaces) في لغة #C:

الواجهات في لغة #C تحتوي فقط على تعريف الدوال والخصائص دون أن تحتوي على التعليمات البرمجية بداخلها. يتم إضافة التعليمات البرمجية وتنفيذها داخل الصف (Class) الذي يرث الواجهة. بمعنى آخر تقيد الصف من التعديل على البنية الأساسية للتعليمات البرمجية (Methods, Properties).فهي تنشأ هيكل مقياسي للصفوف التي ترث الواجهة.

```
interface ILog
{
   void Log(string msgToLog);
}
class MyClass:ILog
{
   public void Log(string msgToLog)
   {
      Console.Write(msgToLog);
      Console.Read();
   }
}
```

تنفيذ البرنامج:

```
MyClass get_info = newMyClass();
get_info.Log("Hello");
```

يكون الناتج: Hello

التغليف (Encapsulation) في لغة #C:

التغليف هو عملية تغليف عنصر أو مجموعة عناصر داخل حزمة فيزيائية أو منطقية. التغليف هو احدى ميزات البرمجة غرضية التوجه. يتم تطبيق عملية التغليف من خلال محددات الوصول (Access Specifiers). لغة #C تدعم محددات الوصول التالية:

محدد الوصول	الوصف	مثال

العام(Public)	تكون الأعضاء العامة متاحة ضمن الصف وخارج الصف	class Rectangle
		{
		public double length;
		public double width;
		
		public double GetArea()
		{
		return length * width;
		}
		public void Display()
		{
		Console.WriteLine("Length: {0}", length);
		Console.WriteLine("Width: {0}", width);
		Console.WriteLine("Area: {0}", GetArea());
		}
		}
		Rectangle r = new Rectangle();
		r.length = 4.5;
		r.width = 3.5;
		r.Display();
		Console.ReadLine();

الخاص (Private)	محدد الوصول الخاص وهو افتراضي تكون الأعضاء الخاصة متاحة فقط ضمن الـ class	

```
class Rectangle
  {

      private double length;
      private double width;

      public void Acceptdetails()
      {
         Console.WriteLine("Enter Length: ");
         length = Convert.ToDouble(Console.ReadLine());
         Console.WriteLine("Enter Width: ");
         width = Convert.ToDouble(Console.ReadLine());
      }
      public double GetArea()
      {
         return length * width;
      }
      public void Display()
      {
         Console.WriteLine("Length: {0}", length);
         Console.WriteLine("Width: {0}", width);
         Console.WriteLine("Area: {0}", GetArea());
```

}
}
 	
Rectangle r = new Rectangle();
r.Acceptdetails();
r.Display();
Console.ReadLine();		

class Shape
{
public void setWidth(int w)
{
width = w;
}
public void setHeight(int h)
{
height = h;
}
protected int width;
protected int height;
}
	محدد الوصول المحمي (خاص بالوراثة)تكون الأعضاء متاحة ضمن الصف والصفوف المورثة	المحمي(protected)

class Rectangle { internal double length; internal double width; double GetArea()	محدد الوصول الداخلي يحدد الصف للاستخدام ضمن المشروع الحالي ولا يمكن استخدامه خارج المشروع	الداخلي(Internal)

```
    {
        return length * width;
    }
    public void Display()
    {
        Console.WriteLine("Length: {0}",
length);
        Console.WriteLine("Width: {0}",
width);
        Console.WriteLine("Area: {0}",
GetArea());
    }
}

class ExecuteRectangle
{
    static void Main(string[] args)
    {
        Rectangle r = new Rectangle();
        r.length = 4.5;
        r.width = 3.5;
        r.Display();
        Console.ReadLine();
    }
}
```

مدخل إلى برمجة تطبيقات الويب

مقدمة:

معظم التطبيقات في وقتنا الحالي تركز على استخدام الانترنت، حيث تعرض التطبيقات من خلال متصفح الانترنت (Browser). صفحات الويب البسيطة الثابتة (Static) تعتمد على HTML و CSS. ولكن لبناء تطبيقات ويب ديناميكية نحتاج إلى أدوات أكثر تقدما وفعالية.

متصفح الانترنت (Web Browser):

الغاية من متصفح الانترنت هو قراءة ملفات HTML. المتصفح لا يعرض لغة HTML وإنما يقوم بترجمة وتفسير محتوى الصفحة التي تضم (HTML Tags). متصفحات الانترنت المستخدمة في وقتنا الحالي:

- Internet Explorer (Microsoft)
- Firefox(Mozilla)
- Chrome (Google)
- Safari (Apple)
- Opera(Opera from Norway)

لغة الـ (HTML):

هي اختصار لـ (Hypertext Markup Language)، وتعني لغة ترميز أو تأشير النصوص التشعبية، تعتبر هذه اللغة من أقدم اللغات والأوسع استخداما لبناء صفحات الويب.

تتألف تعليمات HTML من (Tags)، مثل (<html>) وكل (Tag) (عادة يحتوي على (Tag)بداية و(Tag) اغلاق مثل (<h1></h1>)، وبين هذه (Tags) يقوم مصممين الويب بإضافة النصوص، الصور، الفيديو، الصوت ...الخ. المثال التالي عبارة عن صفحة HTML بسيطة:

```
<!DOCTYPE html>
<html>
    <body>
        <h1>My First Heading</h1>
        <p>My first paragraph</p>
    </body>
</html>
```

لغة الـ(CSS):

هي اختصار لـ (Cascading Style Sheets)، وتعني أوراق التنسيق المتعاقبة تقوم بإعطاء تنسيق شكل عناصر صفحة الويب من حجم وألوان وطريقة العرض. يمكن ان تتحكم CSS بتنسيق عدة صفحات ويب وهذا يوفر علينا الكثير من الجهد والوقت. المثال التالي يعطي اللون الأحمر ومحاذاة للمنتصف لجميع النصوص الموجودة في صفحة الويب:

```
P {
    color: red;
    text-align: center;
}
```

لغة الـ (Java Script):

هي عبارة عن لغة برمجة يتم تنفيذها ضمن ملف HTML، تضيف إلى صفحة الويب ديناميكية، من خلالها يتم تغيير محتوى صفحة الويب (HTML). في المثال التالي يتم تغيير النص عند النقر على الزر.

```
<!DOCTYPE html>
<html>
<body>
<h1>What Can JavaScript Do?</h1>
<p id="demo">JavaScript can change HTML content.</p>
<button type="button" onclick='document.getElementById("demo").innerHTML
= "Hello JavaScript!"'>Click Me!</button>
</body>
</html>
```

بروتوكول الـ (HTTP):

هو اختصار لـ (Hypertext Transfer Protocol)، ويعني بروتوكول نقل النص الفائق، يعتبر لغة التخاطب بين الزبون ومخدم تطبيق الويب. يعتمد عمله على مبدأ الطلب والاستجابة (Request-Response)،

تطبيقات الويب (Web Application):

تطبيقات الويب تتطلب وجود (حاسبين) client –server ويكفي وجود المتصفح (Browser) عند كل زبون(client) ونسخة واحدة من التطبيق على (Server). بينما في تطبيقات الويندوز بحاجة لتنزيل نسخة من التطبيق على كل حاسب مع البرامج المطلوبة لتشغيل التطبيق.

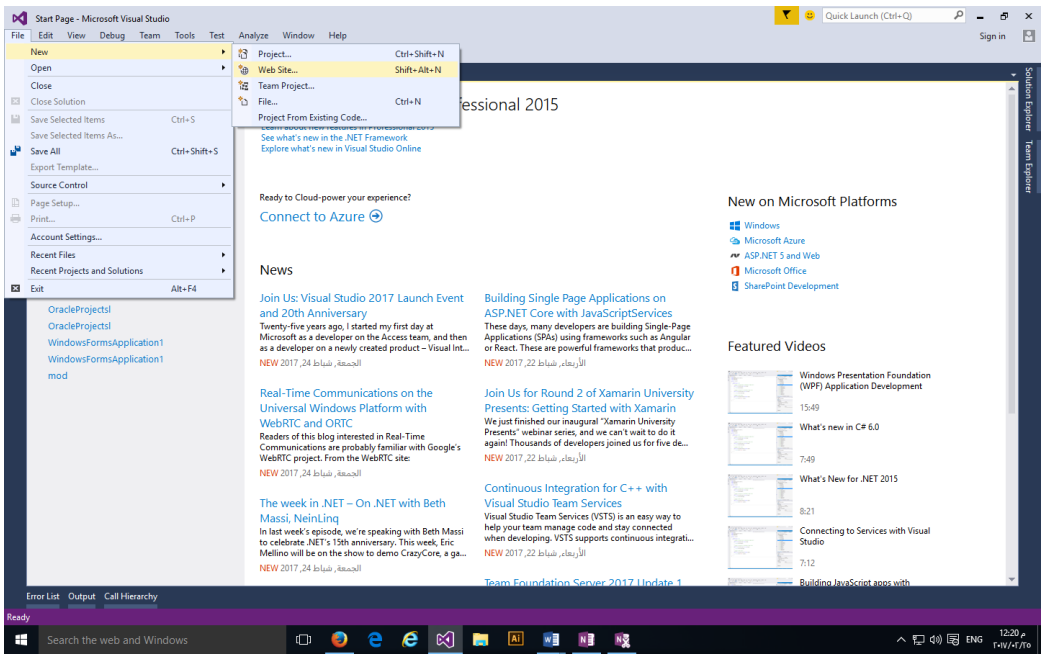
لذلك فإن التعديل والتطوير على تطبيقات الويب هي عملية أسهل وأسرع بكثير من تطبيقات الويندوز لأن عملية التعديل تحدث على النسخة الوحيدة الموجودة على حاسب (Server).

بيئة (Asp.net):

هي اختصار لكلمة (Active Server Pages) عبارة عن بيئة عمل خاصة بتطبيقات الويب طورت من قبل شركة مايكروسوفت تستخدم لبناء تطبيقات ويب ديناميكية. تسمح باستخدام جميع خصائص لغات البرمجة مثل لغات (VB.NET،C#). تقدم شركة مايكروسوفت نسخة مجانية من فيجوال استديو.

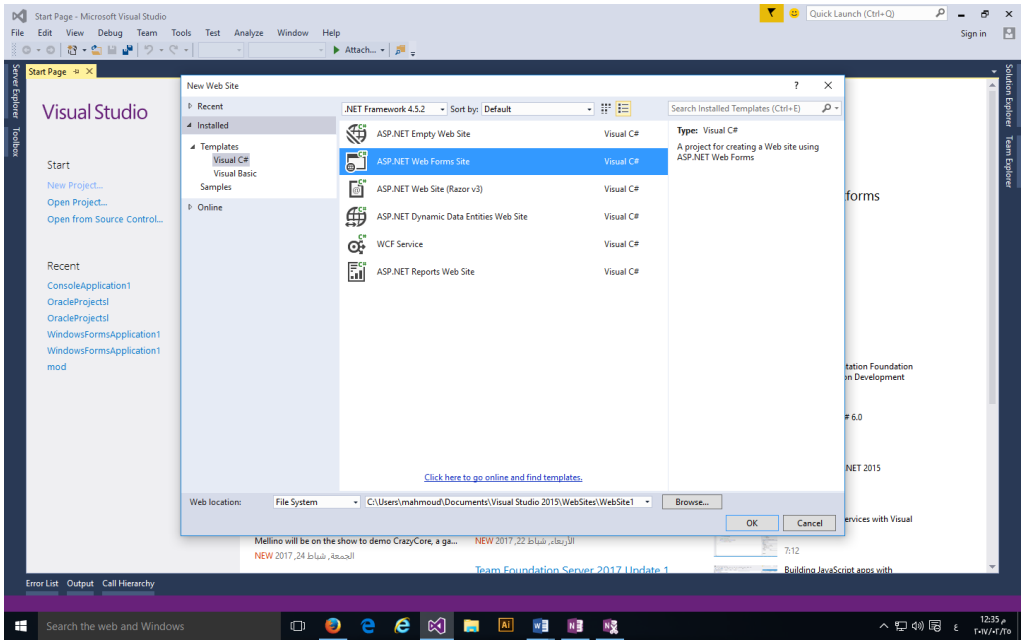
يقوم المستخدم بإنشاء مشروع جديد وذلك بالنقر على خيار ملف (File) ← جديد (New) ← نختار (Web Site)

كما هو مبين بالشكل أدناه.



بعد النقر على خيار (Web Site) تظهر نافذة بعنوان (New Web Site) نختار من على يسار النافذة لغة البرمجة

(Visual C#) ومن ثم نختار (Asp.net Web forms Site) كما هو مبين بالشكل أدناه.



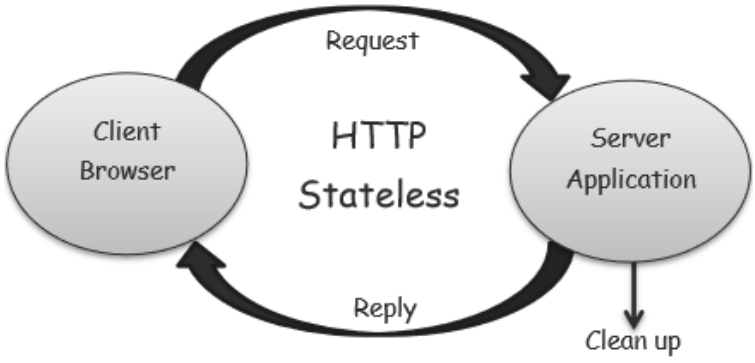
أنواع الملفات في (Asp.net):

اللاحقة	الوصف
.aspx	هذه اللاحقة خاصة بصفحات الويب Asp.net تحتوي على لغة التأشير HTML التي تحدد تصميم واجهة المستخدم
.ascx	تستخدم هذه اللاحقة مع ملفات عناصر التحكم الخاصة بالمستخدم. هذه الصفحات مشابهة لسابقاتها إلا فيما يتعلق بإمكانية الوصول إليها مباشرة. لاستخدام هذه العناصر لابد من استضافتها ضمن صفحة أخرى. تمكن هذه العناصر من تصميم قطع من واجهة المستخدم قابلة لإعادة الاستخدام
.asmx	تمثل الملفات التي تستخدم هذه اللاحقة مجموعات الطرائق الخاصة بخدمات الويب
.master	هذه اللاحقة عبارة عن صفحة تصميم الصفحات الأخرى
.config	يستخدم ملفweb.config تنسيق XML يتضمن هذا الملف اعدادات خاصة بالتطبيق مثل اعدادات الاتصال بقاعدة البيانات Connection String

.cs	يحتوي هذا الملف على التعليمات البرمجية المكتوبة بلغة C# الذي يدعى Code Behind
.dll	هو عبارة عن ملف Assemblyأي المكتبات البرمجية

آلية عمل تطبيقات الويب (Web Application):

يقوم الزبون بإرسال طلب إلى المخدم يدعى Request بينما يرد المخدم على طلب الزبون برد يدعى Replay والعملية تتم بوساطة بروتوكول HTTP والذي يتصف بأنه (Stateless) أي لا يحتفظ ببيانات المستخدم بين كل استدعاء للصفحة، حيث يقوم المخدم بعملية تدعى Clean Up والتي تقوم بحذف المعلومات الخاصة بالزبون التي تنتقل مع الصفحة وذلك للمحافظة على موارد المخدم (الذاكرة) لأنه عند تصفح أعداد كبيرة من المستخدمين لصفحة معينة وتخزين معلومات كل مستخدم على (Server) سيؤدي ذلك إلى فشل المخدم (Server) في تخديم الجميع نتيجة للضغط على الذاكرة لذلك يقوم بعملية Clean Up، لكننا بحاجة لتخزين هذه المعلومات بطريقة ما لذلك تم إيجاد عدة مفاهيم تدعى إدارة الحالة يوجد نوعين من تقنيات إدارة الحالة (State Management) هي:



من جانب الزبون (Client-Side):

- Hidden Field
- View State
- Cookies
- Query Strings

من جانب المخدم (Server-Side):

- Session
- Application

في البداية ما هو الـ (PostBack):

هو الذي يحدد فيما إذا كانت الصفحة قد تم استدعائها لأول مرة أم لا، ويتم ذلك عن طريق متحولBooleanضمن بيئة Net. يدعى IsPostBack.مثلاً: استدعاء الصفحة أول مرة يكون بالدخول للصفحة عن طريق URL ، هنا في هذه المرحلة يكون الـ IsPostBack = false; لأنه أول استدعاء لصفحة ،ولكن استخدام أحد عناصر التحكم (Controls) ضمن الصفحة مثل النقر علىButton لتعبئة جدول ببيانات معينة ضمن الصفحة ففي هذه الحالة تذهب

الصفحة على المخدم (Server) لتنفيذ الحدث OnClick ثم يتم إعادة كتابة الصفحة كما تمت كتابتها بلغة HTML ثم يتم تفعيل الحدث ويملأ الجدول بالبيانات يؤدي إلى استدعاء الصفحة لنفسها وفي هذه الحالة يكون هذا الاستدعاء يدعى PostBack ويكون فيه IsPostBack = true; أي أن الـ PostBack يسمح لتفريق بين استدعاء الصفحة لأول مرة واستدعائها لأكثر من مرة.

Hidden Field:

هو عبارة عن عنصر تحكم في بيئة Asp.net، يستخدم لحفظ بيانات صغيرة الحجم. يقوم بحفظ قيمة وحيدة من متغير، ويفضل استخدامه عندما تكون قيمة المتغير تتغير باستمرار.

```
<asp:HiddenField ID="HiddenField1" runat="server" />
protected void Page_Load(object sender, EventArgs e)
{
   if (HiddenField1.Value != null)
   {
    int val= Convert.ToInt32(HiddenField1.Value) + 1;
    HiddenField1.Value = val.ToString();
    Label1.Text = val.ToString();
   }
}
```

View State:

وظيفتها هي حفظ حالة الصفحة بين استدعاءين متتاليين أي تحفظ المعلومات الخاصة بالزبون (Client) بشكل مؤقت، حيث تقوم بحفظ المعلومات ضمن حقل مخفي (Hidden Field). الـ View Stateتقوم بحفظ المعلومات على مستوى صفحة ويب فقط فإذا انتقل الزبون إلى صفحة أخرى من التطبيق تضيع المعلومات.

```
protected void Page_Load(object sender,
EventArgs e)
{
    if (IsPostBack)
    {
        if (ViewState["count"] != null)
        {
            int ViewstateVal =
Convert.ToInt32(ViewState["count"]) + 1;
            Label1.Text = ViewstateVal.ToString();

ViewState["count"]=ViewstateVal.ToString();
        }
        else
        {
            ViewState["count"] = "1";
        }
    }
}
```

Cookies:

هي عبارة عن ملف نصي صغير يتم إنشاءه من قبل متصفح الإنترنت (Browser)، ويحفظ على حاسوب الزبون من قبل المتصفح، ولا يستخدم ذاكرة حاسوب المخدم (Server)، عادة يحفظ فيه معلومات تعريفية بالزبون.

```
Response.Cookies["nameWithCookies"].Value = "User-Name";        إسناد القيمة
Label2.Text = Request.Cookies["nameWithCookies"].Value;         قراءة القيمة
```

Query Strings:

تستخدم لإضافة معلومات إلى الرابط (URL)، عادةً المعلومات التي تضاف مثل (رقم الصفحة، شروط البحث) أي معلومات عامة.

```
string queryStringData = Request.QueryString["data"];                    إسناد القيمة
Response.Redirect("default.aspx?number=" + queryStringData); إضافة المتغير إلى URL
```

Session:

تعد من أقوى تقنيات إدارة الحالة من جهة المخدم، تستخدم عادة لحفظ معلومات الزبون (رقم المعرف، اسم المستخدم). الـ Session يقوم بالاحتفاظ بالمعلومات إذا انتقل الزبون من صفحة إلى صفحة أخرى من التطبيق، على عكس (View State) التي تحتفظ بالمعلومات على مستوى الصفحة فقط. يقوم المخدم (Server)بإدارة معلومات الزبون عن طريق استخدام رقم تعريفي (Session ID). عندما يقوم الزبون بإرسال طلب Request إلى المخدم (Server) من دون رقم تعريفي (Session ID)، تقوم (Asp.net) بإنشاء رقم تعريفي (Session ID) للزبون وترسله مع كل طلب Request واستجابة Response بين الزبون والمخدم.

```
Session["Count"] = Convert.ToInt32(Session["Count"]) + 1;    إسناد القيمة
Label2.Text = Session["Count"].ToString();                   قراءة القيمة
```

Application:

هو مخزن للبيانات، تكون هذه البيانات متوفرة لجميع صفوف تطبيق Asp.net. الـ (Application) يخزن في ذاكرة المخدم وهو أسرع في تخزين المعلومات واستعادتها من قاعدة البيانات. على عكس (Session) حيث يحتفظ بمعلومات خاصة بمستخدم معين بينما المعلومات التي يحتفظ بها الـ (Application) تكون على مستوى التطبيق.

```
Application["Count"] = Convert.ToInt32(Application["Count"]) + 1;    إسناد القيمة
Label1.Text = Application["Count"].ToString();                       قراءة القيمة
```

أحداث دورة حياة صفحة الويب في (Asp.net):

عندما يقوم الزبون بإرسال طلب Request إلى المخدم (Server)، تمر صفحة الويب بعدة مراحل متسلسلة هي:

الحدث	الوصف
PreInit	تتحقق ما إذا كان هذا الاستدعاء الأول للصفحة إنشاء عناصر التحكم بشكل ديناميكي وضع صفحة رئيسية (Master Page) للصفحة وضع (Themes) لصفحة
Init	بعد إنشاء عناصر التحكم في المرحلة السابقة يمكن هنا تغيير أي عنصر
InitComplete	تحميل عناصر الصفحة يبدأ تكوين (view state) لذلك يمكن استخدام هذا الحدث للتعديل عليها
OnPreLoad	تحميل بيانات (view state) لعناصر التحكم يبدأ تكوين الصفحة
Load	يتم تحميل جميع عناصر التحكم
Control PostBack Event(s)	يتم في هذه المرحلة استدعاء أحداث عناصر التحكم التي سببت الـ (PostBack) مثل الحدث (Click) الخاص بـ (Button)
LoadComplete	تكون عناصر التحكم انتهت من التحميل في هذه المرحلة
OnPreRender	بعد هذا الحدث لم نعد تستطيع تغيير خصائص عناصر التحكم و الـ(view state) لذلك أي تغيير أو تعديل يجب أن يحدث في هذه المرحلة
OnSaveStateComplete	قبل هذه المرحلة تم حفظ قيمة الـ(view state) وعناصر التحكم لذلك أي تغيير وتعديل يتم تجاهله في هذه المرحلة
UnLoad	في هذه المرحلة تتم عملية تدعى (Cleanup Code) وتتطبق على: أغراض الصفوف اغلاق الملفات المفتوحة اغلاق الاتصال مع قاعدة البيانات

أهم عناصر التحكم في (Asp.net):

عناصر التحكم عبارة عن أدوات تُعرض في واجهة المستخدم وتتضمن، مربعات النصوص (Text Boxes)، الأزرار (Buttons)، العناوين (Labels) .... الخ. عناصر تحكم الويب في (Asp.net):

- عناصر التحكم الخاصة بـ(HTML).
- عناصر التحكم الخاصة بـ(HTML) من جهة المخدم.
- عناصر التحكم الخاصة بـ(Asp.net) من جهة المخدم.
- عناصر التحكم الخاصة بـ(Ajax) من جهة المخدم.

عناصر التحكم الخاصة بـ(HTML):

عبارة عن التعليمات الأساسية للغة التأشير (HTML) التي تحدثنا عنها سابقاً، مثال:

```
<input type="text" size="40">
```

عناصر التحكم الخاصة بـ(HTML) من جهة المخدم:

عبارة عن التعليمات الأساسية للغة التأشير (HTML)، ولكن يتم التحكم بها من قبل المخدم، وذلك من خلال إضافة خصائص لـ(Tag)الـ(HTML)، هي (runat="server") وأيضاً إضافة رقم معرف (id). مثال:

```
<"input type="text" id="testtext" size="40" runat="server>
```

الجدول التالي يصف تعليمات الـ(HTML):

الوصف	HTML Tag
ترويسة	<head>element
إدخال من نوع زر	<input type=button|submit|reset>
مربع اختيار	<input type=checkbox>
إدخال من نوع	<input type = file>
حقل مخفي	<input type = hidden>
إدخال من نوع صورة	<input type = image>
إدخال من نوع كلمة سر تكون على شكل	<input type = password>

(**********)	
زر اختياري	<input type = radio>
إدخال من نوع نص	<input type = text>
صورة	<img> element
رابط تشعبي	<a> element
زر	<button> element
النماذج	<form> element
جدول	<table> element
خلية داخل الجدول	<td> and <th>
صف من جدول	<tr> element
عنوان	<title> element
قائمة منسدلة	<select&t; element

عناصر التحكم الخاصة بـ(Asp.net) من جهة المخدم:

هي عناصر التحكم الأساسية الموجودة في (Asp.net)، الجدول التالي يعرض أهم خصائص عناصر التحكم:

الوصف	الخاصية
لون الخلفية	BackColor
لون الإطار	BorderColor
شكل الإطار	BorderStyle
عرض الإطار	BorderWidth
التحقق من القيمة المدخلة	CausesValidation
رقم المعرف للغة التأشير (HTML)	ClientID
شكل عنصر التحكم	ControlStyle
صف لغة (CSS)	CssClass
تفعيل عنصر التحكم (True or False)	Enabled
الخط	Font
لون الخط	Forecolor
تحديد طول العنصر	Height

ID	رقم المعرف من جهة المخدم
Visible	ظهور العنصر أو اختفاءه (True or False)
Width	تحديد عرض العنصر
ToolTip	أداة الإرشاد

الجدول التالي يعرض أهم أحداث (Events) عناصر التحكم في (Asp.net):

عنصر التحكم	اسم الخاصية	الحدث
Button, image button, link button, image map	OnClick	Click
Button, image button, link button	OnCommand	Command
Text box	OnTextChanged	TextChanged
Drop-down list, list box, radio button list, check box list.	OnSelectedIndexChanged	SelectedIndexChanged
Check box, radio button	OnCheckedChanged	CheckedChanged

سوف نتحدث عن عناصر التحكم الأساسية (Basic Controls) في (Asp.net) وتتضمن:

الزر (Button Control):

هناك ثلاث أنواع هي:

- Button: يعرض على شكل نص داخل مستطيل
- Link Button: يُعرض على شكل نص
- Image Button: يعرض على شكل صورة

التعليمة الأساسية البرمجية لزر (Button):

```
<asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Click" / >
```

الجدول التالي يعرض الخصائص الأساسية لزر (Button):

اسم الخاصية	الوصف
Text	النص الذي يظهر على الزر
ImageUrl	مسار الصورة التي سوف تظهر على شكل زر
AlternateText	النص الذي يظهر في حال المتصفح لم يعرض الصورة
CausesValidation	التحقق من القيم عند النقر على الزر
CommandName	عبارة عن قيمة من نوع (String) يتم تمريرها عند النقر على الزر
CommandArgument	عبارة عن قيمة من نوع (String) يتم تمريرها عند النقر على الزر
PostBackUrl	رابط الصفحة الذي تتطلب عند النقر على الزر

مربعات النصوص والعناوين (Text Boxes and Labels Controls):

المربع النصي (Text Box) عادةً يستخدم للإدخال من قبل المستخدم. ويمكن ادخال سطر واحد أو عدة أسطر من النصوص اعتماداً على خواص المربع النصي.

```
<asp:TextBox ID="txtstate" runat="server" ></asp:TextBox>
```

العنوان (Label) طريقة سهلة تستخدم لعرض النصوص.

```
<asp:Label ID="label1" runat="server" ></asp:Label>
```

الجدول التالي يبين أهم خصائص مربع النصوص والعناوين:

اسم الخاصية	الوصف
TextMode	تحديد نوع المربع النصي
Text	محتوى النص الموجود بالمربع النصي أو العنوان
MaxLength	تحديد العدد الأقصى للمحارف المدخلة
ReadOnly	تحديد ما إذا كان المستخدم يستطيع التعديل على النص أم لا

مربعات وأزرار الاختيار (Check Boxes and Radio Buttons):

مربع الاختيار (Check Box) يعرض خيار وحيد فقط من خلاله يمكن للمستخدم أن يختار أو ألا يختار (check or uncheck).

```
<asp:CheckBox ID= "chkoption" runat= "Server"></asp:CheckBox>
```

زر الاختيار (Radio Button) يعرض مجموعة من الخيارات حيث يمكن للمستخدم اختيار خيار واحد فقط.

```
<asp:RadioButton ID= "rdboption" runat= "Server"></asp: RadioButton>
```

الجدول التالي يبين أهم خصائص مربع وزر الاختيار:

اسم الخاصية	الوصف
Text	النص الذي يظهر بجانب مربع أو زر الاختيار
Checked	تحديد إذا ما تم الاختيار أم لا
GroupName	اسم المجموعة التي ينتمي إليها العنصر

القائمة المنسدلة (Drop-down list):

القائمة المنسدلة (Drop-down list) تتضمن خيار وحيد أو عدة خيارات يمكن للمستخدم الاختيار منها.

```
<asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged">
</asp:DropDownList>
```

الجدول التالي يبين أهم خصائص القائمة المنسدلة:

اسم الخاصية	الوصف
Items	مجموعة الخيارات الموجودة في القائمة المنسدلة
SelectedIndex	مؤشر العنصر المحدد حاليا
SelectedValue	قيمة العنصر المحدد حاليا

الجدول التالي يبين أهم خصائص الخيارات (Items):

اسم الخاصية	الوصف
Text	النص الذي يظهر للخيار
Selected	يشير إلى ما إذا تم تحديد العنصر
Value	القيمة المرتبطة مع الخيار

الصورة (Image):

من خلال هذا العنصر يتم عرض الصور في صفحة الويب أو النص البديل في حال لم تظهر الصورة.

```
<asp:Image ID="Image1" runat="server">
```

الجدول التالي يعرض خصائص عنصر التحكم الصورة:

اسم الخاصية	الوصف
AlternateText	النص البديل الذي يظهر بدلاً من الصورة في حال لم تظهر
ImageAlign	محاذاة الصورة (يمين ،يسار،منتصف)
ImageUrl	مسار الصورة التي سوف تعرض في صفحة الويب

عناصر التحكم الخاصة بـ(Ajax) من جهة المخدم:

هي اختصار لـ (Asynchronous JavaScript and XML) هذه التقنية مسؤولة عن تسريع وقت الاستجابة Response.نستطيع من خلالها:

- تحديث صفحة الويب من دون إعادة تحميل الصفحة
- طلب بيانات من المخدم بعد تحميل الصفحة
- استقبال بيانات من المخدم بعد تحميل الصفحة

الـ (Asp.net) تحتوي على عناصر تحكم (Ajax) هي:

The ScriptManager Control:

يعتبر أهم عنصر تحكم ويجب إضافته إلى صفحة (Asp.net) لكي تعمل باقي العناصر.

```
<asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
```

The UpdatePanel Control:

هي عبارة عن حاوية لعناصر التحكم الأخرى ولا تظهر على واجهة المستخدم. عندما يقوم عنصر تحكم داخل (UpdatePanel) بعملية (PostBack) أي إعادة تحميل الصفحة تتدخل (UpdatePanel) وتقوم بتحديث جزء من الصفحة وليس إعادة تحميل الصفحة كلها. على سبيل المثال إذا كان عنصر التحكم (Button) داخل (UpdatePanel) وتم النقر عليه (Click) في هذه الحالة فقط عنصر التحكم الذي داخل (UpdatePanel) سوف يتأثر بالـ (PostBack)، عناصر التحكم الموجودة في الأجزاء الأخرى من الصفحة لن تتأثر، وهذا ما يسمى إعادة تحميل جزئي لصفحة أو التحميل غير المتزامن (Asynchronous Post Back).

```
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
<ContentTemplate>
<asp:Button ID="btnpartial" runat="server" onclick="btnpartial_Click"
Text="Partial PostBack"/>
<br />
<br />
<asp:Label ID="lblpartial" runat="server"></asp:Label>
</ContentTemplate>
</asp:UpdatePanel>
```

The UpdateProgress Control:

هذا العنصر يقدم نوع من التغذية العكسية لمتصفح الانترنت (Browser) بينما يتم تحديث عنصر أو عدة عناصر تحكم. على سبيل المثال إذا قام المستخدم بعملية تسجيل دخول (Login) وانتظر المخدم حتى يستجيب يظهر على الواجهة عبارة مثل (..... Loading) مبينا أن العملية قيد الإنجاز.

```
<asp:UpdateProgress ID="UpdateProgress1" runat="server"
DynamicLayout="true" AssociatedUpdatePanelID="UpdatePanel1" >
<ProgressTemplate>
    Loading...
</ProgressTemplate>
</asp:UpdateProgress>
```

Asp.Net SignalR

مقدمة:

هي عبارة عن مكتبة برمجية تتيح لمبرمجي الـ Asp.net بإضافة وظائف إلى تطبيق الويب تعمل في الزمن الحقيقي ببساطة. تطبيقات الويب في الزمن الحقيقي هي إعطاء القدرة للمخدم ليقوم بما يسمى بعملية الدفع (Push) للمحتوى لجميع الزبائن المتصلين بالمخدم مثل تطبيقات وسائل التواصل الاجتماعي، الألعاب التي تتضمن أكثر من مستخدم، التطبيقات المالية. وهي تتضمن مكتبة ASP.NET من جهة المخدم ومكتبة JavaScript من جهة الزبون وذلك لتسهيل عملية إدارة الاتصال بين الزبون والمخدم ودفع التحديثات التي تحدث على المحتوى إلى الزبون.

خطوات تنصيب مكتبة (SignalR):

- إضافة مكتبة (SignalR) إلى تطبيق الويب ASP.NET
- إنشاء صف (Hub Class) ليقوم بعملية الدفع (Push) للمحتوى إلى جميع الزبائن

```
using System;
using System.Web;
using Microsoft.AspNet.SignalR;
namespace SignalRChat
{
    public class ChatHub : Hub
    {
        public void Send(string name, string message)
        {
            Clients.All.broadcastMessage(name, message);
        }
    }
}
```

- إنشاء صف (OWIN Startup Class) لتهيئة التطبيق

```
using Microsoft.Owin;
using Owin;
[assembly: OwinStartup(typeof(SignalRChat.Startup))]
namespace SignalRChat
{
    public class Startup
    {
        public void Configuration(IAppBuilder app)
        {
            app.MapSignalR();
        }
    }
}
```

- استخدام مكتبة (SignalR jQuery) في صفحة الويب

يتم إضافة دالة إلى صفحة HTML لكي يستطيع المخدم استدعائها ليقوم بعملية دفع المحتوى إلى الزبون.

```
var chat = $.connection.chatHub;
```

من خلال الجملة البرمجية التالية يتم إنشاء دالة استدعاء بلغة (JavaScript). الصف الموجود على المخدم (hub class) والذي تم انشاءه مسبقاً يقوم باستدعاء هذه الدالة لدفع المحتوى إلى الزبون.

```
chat.client.broadcastMessage = function (name, message)}
        var encodedName = $('<div />').text(name).html;()
        var encodedMsg = $('<div />').text(message).html;()
$        ('#discussion').append('<li><strong>' + encodedName
/>' +            strong>:  ' + encodedMsg + '</li;('<
    };
```

من خلال الجملة البرمجية التالية يتم فتح الاتصال مع الصف الموجود على المخدم، وتقوم بإرسال الرسالة بعد النقر على الزر الموجود في صفحة HTML.

```
$.connection.hub.start().done(function () {
        $('#sendmessage').click(function () {
            chat.server.send($('#displayname').val(), $('#message').val());
            $('#message').val('').focus();
        });
    });
```

نضيف إلى صفحة الويب الجملة البرمجية التالية بلغة HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>SignalR Simple Chat</title>
    <style type="text/css">
        .container {
            background-color: #99CCFF;
            border: thick solid #808080;
            padding: 20px;
            margin: 20px;
        }
    </style>
</head>
<body>
    <div class="container">
        <input type="text" id="message" />
        <input type="button" id="sendmessage" value="Send" />
        <input type="hidden" id="displayname" />
        <ul id="discussion">
        </ul>
    </div>
</body>
</html>
```

أمثلة عامة

إيجاد جميع النظائر الضربية (mod = 256) بلغة C#:

```
int b, k = 0;
int mod = 256;
for (int a = 1; a < mod; a++)
{
    for (int i = 1; i < mod; i++)
    {
        k = i * mod + 1;
        if (k >= mod)
        {
            k = k - mod;
        }
        b = k / a;

        if (a * b == k)
        {
            Console.WriteLine(b);
            break;
        }
    }
}
Console.ReadLine();
```

إيجاد النظير الضربي للعدد (a = 17 mod 256) بلغة C#:

```
int b, k = 0;
int mod = 256;
int a = 17;
for (int i = 1; i < mod; i++)
{
    k = i * mod + 1;
    if (k >= mod)
    {
        k = k - mod;
    }
    b = k / a;

    if (a * b == k)
    {
        Console.WriteLine(b);
        break;
    }
}
Console.ReadLine();
```

يكون الناتج = (241)

المقابل العددي العشري لحروف ASCII بلغة C#:

```
string letter = "B";
byte[] asciiBytes = Encoding.GetEncoding(1256).GetBytes(letter);
Console.WriteLine(asciiBytes[0]);
Console.ReadLine();
```

يكون الناتج = (66)

الحروف المقابلة للأعداد العشرية في جدول ASCII بلغة #C:

```
int letter =66;
string decode =
Encoding.GetEncoding(1256).GetString(BitConverter.GetBytes(letter), 0, 1);
Console.WriteLine(decode);
Console.ReadLine();
```

يكون الناتج = (B)

تطبيق التشفير بطريقة قيصر بلغة #C:

```
using System.Text.RegularExpressions;

string msg = "BOOK";
int key = 10;
string[] characters = Regex.Split(msg, string.Empty);
msg = "";
foreach (string plain_text in characters)
{
    if (plain_text != "")
    {
        byte[] alpha_num =
Encoding.GetEncoding(1256).GetBytes(plain_text);
        int cipher_text = Convert.ToInt32(alpha_num[0]) + key;
        msg +=
Encoding.GetEncoding(1256).GetString(BitConverter.GetBytes(cipher_text),
0, 1);
    }
}
Console.WriteLine(msg);
Console.ReadLine();
```

تطبيق فك التشفير بطريقة قيصر بلغة #C:

```
using System.Text.RegularExpressions;

string msg = "LYYU";
int key = 10;
string[] characters = Regex.Split(msg, string.Empty);
msg = "";
foreach (string cipher_text in characters)
{
    if (cipher_text != "")
    {
        byte[] alpha_num =
Encoding.GetEncoding(1256).GetBytes(cipher_text);
        int plain_text = Convert.ToInt32(alpha_num[0]) - key;
        msg +=
Encoding.GetEncoding(1256).GetString(BitConverter.GetBytes(plain_text),
0, 1);
    }
}
Console.WriteLine(msg);
Console.ReadLine();
```

يكون النص الأصلي= (BOOK)

تطبيق التشفير بطريقة أفاين بلغة #C:

```
using System.Text.RegularExpressions;

string msg = "BOOK";
int key_A = 10;
int key_B = 11;
string[] characters = Regex.Split(msg, string.Empty);
msg = "";
foreach (string plain_text in characters)
{
    if (plain_text != "")
    {
        byte[] alpha_num =
Encoding.GetEncoding(1256).GetBytes(plain_text);
        int cipher_text = (alpha_num[0] * key_B + key_A) % 256;
        msg +=
Encoding.GetEncoding(1256).GetString(BitConverter.GetBytes(cipher_text),
0, 1);
    }
}

Console.WriteLine(msg);
Console.ReadLine();
```

يكون النص المشفر= (àooC)

تطبيق فك التشفير بطريقة أفاين بلغة #C:

```
using System.Text.RegularExpressions;
string msg = "àooC";
int key_A = 10;
int key_B = 11;
if (key_B >= 256)
{
    key_B = key_B % 256;
}
for (int i = 1; i < 256; i++)
{
    int k = i * 256 + 1;
    int b = k / key_B;
    if (key_B * b == k)
    {
        key_B = b;
        break;
    }
}
string[] characters = Regex.Split(msg, string.Empty);
msg = "";
foreach (string cipher_text in characters)
{
    if (cipher_text != "")
    {
        byte[] alpha_num = Encoding.GetEncoding(1256).GetBytes(cipher_text);
        int plain_text = key_B * (alpha_num[0] - key_A);
        if (plain_text < 0)
        {
            plain_text = plain_text * -1;
            int result = plain_text % 256;
            plain_text = 256 - result;
        }
        else
        {
            plain_text = plain_text % 256;
        }
        msg += Encoding.GetEncoding(1256).GetString(BitConverter.GetBytes(plain_text), 0, 1);
    }
}
Console.WriteLine(msg);
Console.ReadLine();
```

يكون النص الأصلي= (BOOK)

تطبيق التشفير بطريقة هيل الشعاعية بلغة C#:

```
string msg = "BOOK";
int[,] plaintext = new int[2, 2];
int[,] key = new int[2, 2];
int[,] cipher_text = new int[2, 2];
string[] characters = new string[plaintext.Length];
int count = 0;
foreach (string plain_text in msg.Select(x =>
x.ToString()).ToArray())
{
    characters[count] = plain_text;
    count++;
}
count = 0;
for (int column = 0; column < 2; column++)
{
    for (int row = 0; row < 2; row++)
    {
        byte[] alpha_num =
Encoding.GetEncoding(1256).GetBytes(characters[count]);
        plaintext[row, column] = Convert.ToInt32(alpha_num[0]);
        count++;
    }
}
msg = "";
key[0, 0] = 17;
key[1, 0] = 17;
key[0, 1] = 0;
key[1, 1] = 9;
for (int column = 0; column < 2; column++)
{
    for (int row = 0; row < 2; row++)
    {
        for (int l = 0; l < 2; l++)
        {
            cipher_text[row, column] += key[row, l] * plaintext[l,
column];
        }
        if (cipher_text[row, column] >= 256)
        {
            cipher_text[row, column] %= 256;
        }
        msg +=
Encoding.GetEncoding(1256).GetString(BitConverter.GetBytes(cipher_
text[row, column]), 0, 1);
    }
}
Console.WriteLine(msg);
Console.ReadLine();
```

يكون ناتج النص المشفر= (b)?â

تطبيق فك التشفير بطريقة هيل الشعاعية بلغة C#:

```
string msg = "b)?â";
int[,] plaintext = new int[2, 2];
int[,] key = new int[2, 2];
int[,] cipher_text = new int[2, 2];
string[] characters = new string[plaintext.Length];
int count = 0;
foreach (string plain_text in msg.Select(x =>
x.ToString()).ToArray())
{
    characters[count] = plain_text;
    count++;
}
count = 0;
for (int column = 0; column < 2; column++)
{
    for (int row = 0; row < 2; row++)
    {
        byte[] alpha_num =
Encoding.GetEncoding(1256).GetBytes(characters[count]);
        plaintext[row, column] = Convert.ToInt32(alpha_num[0]);
        count++;
    }
}
msg = "";
key[0, 0] = 241;
key[1, 0] = 199;
key[0, 1] = 0;
key[1, 1] = 57;
for (int column = 0; column < 2; column++)
{
    for (int row = 0; row < 2; row++)
    {
        for (int l = 0; l < 2; l++)
        {
            cipher_text[row, column] += key[row, l] * plaintext[l,
column];
        }
        if (cipher_text[row, column] >= 256)
        {
            cipher_text[row, column] %= 256;
        }
        msg +=
Encoding.GetEncoding(1256).GetString(BitConverter.GetBytes(cipher_
text[row, column]), 0, 1);
    }
}

Console.WriteLine(msg);
Console.ReadLine();
```

يكون ناتج النص الأصلي= (BOOK)

ملخص الفصل الرابع:

قدمنا في هذا الفصل نمطاً جديدا من التشفير باستخدام المصفوفات المنتهية المجزاة وعرضنا من خلال الأمثلة أحد الامكانات لتجزيء المصفوفة إلى عدة مصفوفات وتطبيق أنواع خاصة من الخوارزميات ضمن كل مصفوفة لتشفيرها.

النتائج والتوصيات:

من التوصيات الهامة نقترح مايلي:

1. نوصي باستخدام المصفوفات المجزأة بحيث نشفر كل جزء منها مشفر بوساطة خوارزمية RSA مع تغيير المفاتيح المكونة لهذه الطريقة .
2. نشفر أجزاء منها بوساطة خوارزمية RSA ، و أجزاء أخرى بتشفير كلاسيكي مع تغيير المفاتيح المكونة لكل طريقة.
3. كتابة كود برمجي بلغة برمجية (C#) مثلاً وتطبيقه ضمن الحاسوب بحيث يتم تجزيء المصفوفة واختيار خوارزميات التشفير والمفاتيح المختلفة اعتمادا على المستخدم أو يتم برمجته مسبقا للتنفيذ وفقا لشروط خاصة واعتمادا على آليات ومبرهنات التقسيم للمصفوفات إلى مصفوفات جزئية.

المراجع العربية

1. محمد نور شمه، عبد الباسط الخطيب، **التشفير العربي المطور"التعمية المطورة "**، مجلة جامعة البعث للعلوم الهندسية، مجلد / 30 /عدد/17 /صفحة 9-24 2008 .
2. محمد نور شمه، عبد الباسط الخطيب، أحمد صطيف ،**التشفير باستخدام دالة فيثاغورثية خاصة** ، مجلة جامعة البعث مجلد / 38 / لعام 2016.
3. محمد نور شمه، عبد الباسط الخطيب، أحمد صطيف ،**التشفير باستخدام المصفوفة الفيثاغورثية الأولية** ، مجلة جامعة البعث مجلد / 38 / لعام 2016.
4. محمد نور شمه، عبد الباسط الخطيب، **استخدام المصفوفات الجزئية المنتهية لتشفير Hill للرسائل المرمزة بنظام ASCII**، مجلة جامعة دمشق مجلد / 38 / لعام 2015.
5. محمد نور شمه، محمود اللحام، **التشفير باستخدام متجهات تولد مصفوفات مثلية ذات طبيعة خاصة**، مجلة جامعة البعث مجلد / 39 / لعام 2017.

References

1. Christof Paar & Jan Pelzl, *Understanding Cryptography*, © Springer-Verlag Berlin Heidelberg 2010
2. GARRETT, P.: *Making, Breaking Codes. An Introduction to Cryptology.* Prentice–Hall(2007)
3. GOLDREICH, O.: *Foundations of Cryptography. Basic Applications.*Cambridge University Press (2009)
4. KATZ, J. & LINDELL, Y.: *Introduction to Modern Cryptography.* Chapman & Hall /CRC (2008)
5. Keijo Ruohonen , *Mathematical Cryptology*,Translation by Jussi Kangas and Paul Coughlan(2014)
6. ROSEN, K.H..: *Elementary Number Theory.* Longman (2010)
7. Rosen, K.H: *Discrete mathematics and its applications*. 7th ed., The McGraw-Hill Companies ,(2012)
8. Christof Paar & Jan Pelzl، ***Understanding Cryptography***، © Springer-Verlag Berlin Heidelberg 2010
9. GARRETT، P.: ***Making، Breaking Codes. An Introduction to Cryptology***. Prentice–Hall(2007)
10. GOLDREICH، O.: Foundations of Cryptography. Basic Applications.Cambridge University Press (2009)

11. KATZ، J. & LINDELL، Y.: ***Introduction to Modern Cryptography***. Chapman & Hall /CRC (2008)

12. Keijo Ruohonen ، ***Mathematical Cryptology***،Translation by Jussi Kangas and Paul Coughlan(2014)

13. ROSEN، K.H..: ***Elementary Number Theory***. Longman (2010)

14. Rosen، K.H: ***Discrete mathematics and its applications***. 7th ed.، The McGraw-Hill Companies ،(2012)

15. M.N. Shamma، A. Al-Khatib، 2009. **On the modern cryptology method of Hill for encoded letters with ASCII system**، Far East Journal of Mathematical Education FJME Volume 3 No. 2 ، June ،pp. 183 – 193.