****

**ملخص**

ندرس في مشروع البحث هذا خوارزميات التشفير اللامتناظرة و آلية تبادل المفاتيح في شبكة. و سنقدم خوارزمية لامتناظرة من تصميم طلاب المشروع, مع التطبيق العملي بلغات البرمجة.

تقرير مشروع بحث بعنوان :

تقديم الطلاب : محمود قره فلاح – جابر أديب – خالد اسماعيل – مصطفى خليل

الصف : الحادي عشر – الثاني عشر

تاريخ : 13/5/2015

إشراف : المدرس حبيب عيسى

Asymmetric Cryptography

التشفير اللامتناظر

# الفهرس

[المقدمة : 5](#_Toc419228227)

[كيف تستخدم هذا البحث 6](#_Toc419228228)

[1الفصل الأول : خلفية رياضية 7](#_Toc419228229)

[1.1مبادئ في نظرية الأعداد 7](#_Toc419228230)

[1.1.1قابلية القسمة 7](#_Toc419228231)

[1.1.2الأعداد الأولية 7](#_Toc419228232)

[1.1.3القاسم المشترك الأكبر و المضاعف المشترك الأصغر 8](#_Toc419228233)

[1.1.4التطابق الخطي 8](#_Toc419228234)

[1.1.5خوارزمية القسمة 9](#_Toc419228235)

[1.1.6مبرهنة أولر 9](#_Toc419228236)

[1.1.7مبرهنة البواقي الصينية 10](#_Toc419228237)

[1.2رياضيات المنحنيات الناقصية 10](#_Toc419228238)

[1.2.1معادلة منحني ناقصي 10](#_Toc419228239)

[1.2.2تعريف المنحني الناقصي 11](#_Toc419228240)

[1.2.3العمليات الزمرية على المنحنيات الناقصية 12](#_Toc419228241)

[1.3مسائل أساسية في نظرية الأعداد 14](#_Toc419228242)

[1.3.1مسألة التحليل 14](#_Toc419228243)

[1.3.2مسألة الجذور التربيعية بالمقاس  14](#_Toc419228244)

[1.3.3مسألة اللوغاريتم المتقطع 14](#_Toc419228245)

[1.3.4مسألة اللوغاريتم المتقطع الموسعة 14](#_Toc419228246)

[1.3.5مسألة اللوغاريتم المتقطع في المنحنيات الناقصية : 14](#_Toc419228247)

[1.3.6مسألة ديفي-هيلمان 14](#_Toc419228248)

[1.3.7مسألة ديفي-هيلمان الموسعة 15](#_Toc419228249)

[1.3.8مسألة المجموعة الجزئية 15](#_Toc419228250)

[1.4مبادئ الجبر المجرد 15](#_Toc419228251)

[1.4.1العلاقات الثنائية 15](#_Toc419228252)

[1.4.2قوانين التشكيل (العمليات) 16](#_Toc419228253)

[1.4.3البنى الجبرية 16](#_Toc419228254)

[1.4.4المجموعة المولدة لزمرة 18](#_Toc419228255)

[1.4.5الزمرة الدائرية المولدة بمجموعة 18](#_Toc419228256)

[1.5خوارزميات أساسية 18](#_Toc419228257)

[1.5.1الخوارزمية الإقليدية 18](#_Toc419228258)

[1.5.2الخوارزمية الإقليدية الممددة 19](#_Toc419228259)

[1.5.3حساب الأس 21](#_Toc419228260)

[1.5.4حساب مجموع نقطة في منحني ناقصي 22](#_Toc419228261)

[1.5.5اختبارات الأولية 22](#_Toc419228262)

[1.5.5.1اختبار فيرما 23](#_Toc419228263)

[1.5.5.2اختبار ميلر-رابين 24](#_Toc419228264)

[2الفصل الثاني : بعض الخوارزميات اللامتناظرة 25](#_Toc419228265)

[2.1خوارزمية RSA 25](#_Toc419228266)

[2.1.1توليد المفاتيح 25](#_Toc419228267)

[2.1.2التشفير و فك التشفير 26](#_Toc419228268)

[2.1.3برهان صحة خوارزمية RSA 28](#_Toc419228269)

[2.1.4تقنيات تسريع تنفيذ خوارزمية RSA 29](#_Toc419228270)

[2.1.4.1التشفير السريع باستخدام المفاتيح العامة الصغيرة 29](#_Toc419228271)

[2.1.4.2فك التشفير السريع بمبرهنة البواقي الصينية 30](#_Toc419228272)

[2.1.5أمن RSA 32](#_Toc419228273)

[2.1.5.1علاقة الأمن بالتحليل 32](#_Toc419228274)

[2.1.5.2المفاتيح العامة و الرسائل الصغيرة 32](#_Toc419228275)

[2.1.5.3المفاتيح الخاصة الصغيرة 33](#_Toc419228276)

[2.1.5.4المقاسات المشتركة 33](#_Toc419228277)

[2.1.5.5الرسائل اللامخفية 34](#_Toc419228278)

[2.2تبادل مفتاح ديفي-هيلمان 34](#_Toc419228279)

[2.2.1آلية عمل تبادل مفتاح ديفي-هيلمان 34](#_Toc419228280)

[2.2.2برهان صحة تبادل مفتاح ديفي هيلمان 35](#_Toc419228281)

[2.3خوارزميات المنحنيات الناقصية 36](#_Toc419228282)

[2.3.1آلية تبادل مفتاح ديفي ـ هيلمان في المنحنيات الناقصية 36](#_Toc419228283)

[2.3.2برهان صحة تبادل مفتاح ديفي-هيلمان في المنحنيان الناقصية 37](#_Toc419228284)

[2.3.3ملاحظة على التعقيد 38](#_Toc419228285)

[3الفصل الثالث نتاج المشروع 40](#_Toc419228286)

[3.1الخوارزمية 40](#_Toc419228287)

[3.1.1الزمرة الدائرية 40](#_Toc419228288)

[3.1.2تبادل مفتاح ديفي-هيلمان في الدوائر القطبية 41](#_Toc419228289)

[3.1.3برهان صحة تبادل مفتاح ديفي هيلمان في الدوائر القطبية 41](#_Toc419228290)

[3.1.4الحساب السريع 42](#_Toc419228291)

[3.2البرنامج 42](#_Toc419228292)

[3.2.1شيفرة RSA 42](#_Toc419228293)

[3.2.2شيفرة الدوائر القطبية 45](#_Toc419228294)

[3.2.3الشيفرة المتناظرة Trivium 47](#_Toc419228295)

[3.2.3.1 الكود 48](#_Toc419228296)

[3.2.3.2تنفيذ البرنامج 50](#_Toc419228297)

[4الخاتمة 52](#_Toc419228298)

[5المراجع : 55](#_Toc419228299)

# المقدمة :

انطلق علم التشفير منذ آلاف السنين, مع الجنرال خوليوس قيصر. قاد الجنرال قيصر فتوحات عظيمة في أوروبا, و كان يراسل قادته بالمراسلات التقليدية. لكن واجهته مشكلة. في بعض الحالات, قبض أعداءه على رسله, فكشفوا بعض خططه, و منعوا البعض الآخر من الوصول. هذا ما كلفه العديد من الضحايا و الخسارات. فقام الجنرال قيصر بابتكار حل عبقري لمشكلته, بدل كل حرف "A" في رسالته بحرف "D", و كل حرف "B" بحرف "E" و هكذا بدل كل حرف بالأبجدية بالحرف الذي يليه بثلاث مراتب. و يقوم المرسل إليه بتبديل كل حرف بالحرف الذي يسبقة بثلاث مراتب ليقرأ الرسالة الأصلية. ففي حال قبض أعدائه على رسله, فلن يفهموا أي شيء مكتوب في مراسلاته. كان حل قيصر هو انطلاقة علم التشفير. و الآن بعد مئات السنين, على الرغم من التطورات الهائلة في علم الحاسوب و خوارزميات التشفير, نحن لا نزال نواجه نفس مشكلة الجنرال قيصر. يجب علينا أن نرسل رسائل خاصة ضمن وسط ليس آمناً لترسل فيه. و في الواقع, حللنا مشكلتنا بنفس الطريقة التي حلها لها الجنرال قيصر. فظهر علم التشفير, و هو العلم الذي يهتم بدراسة الخوارزميات و الطرق الرياضية لتشفير و فك تشفير الرسائل بهدف إخفاء معناها عن الأطراف الغير المصرح بها.

لعلم التشفير أثر هائل في حياتنا. لنذكر الحرب العالمية الثانية. اعتمد الألمان آلة "إنيغما" لتشفير رسائلهم و إخفاء معناها عن العدو. حتى جاء العبقري ألان تيورينغ, الذي قام ببناء الآلة الشهيرة "كولوسيس" التي بدورها كسرت تشفير آلة إنيغما. و فجأة, أصبحت جميع مراسلات الألمان مفضوحة أمام الحلفاء. هذا ما جعلهم يخسرون معركة شمال الأطلسي و فيما بعد, الحرب العالمية الثانية.

تطورت خوارزميات التشفير بشكل كبير, و كذلك خوارزميات كسر التشفير. فاضطررنا إلى بناء خوارزميات جديدة مضادة لخوارزميات الكسر. و بعد تراكم الخبرات في التشفير و كسر التشفر وصلنا الآن إلى حد كبير من الأمان في المراسلات عبر شبكة الانترنت. لكن, لا تزال الفكرة العامة هي نفسها التي استعملها قيصر و الألمان. يقوم الطرف الأول بتشفير الرسالة  بواسطة المفتاح , و يرسل هذه الرسالة في الوسط الغير آمن لتصل هذه الرسالة إلى الطرف الآخر حيث يقوم الطرف الآخر, بمعرفة المفتاح  بفك تشفير هذه الرسالة عن طريق تنفيذ خوارزمية عكسية.

بنيت جميع خوارزميات التشفير على افتراض أن الطرفين اللذين يتبادلان الرسائل متفقان على المفتاح . و لذلك فهو يدعى التشفير المتناظر, لأن مفتاح التشفير هو مفتاح فك التشفير ذاته. لكن, كيف من الممكن أن يتفق طرفان على مفتاح التشفير؟ فقد تراسل في بعض الحالات أشخاصاً لم تتمكن و لن تتمكن من التواصل معهم إلا عبر شبكة غير آمنة. فكيف ستتفقان على مفتاح من دون أن تعرفه أي أطراف أخرى؟

هنا يأتي التشفير اللامتناظر. ابتكر هذا النوع من التشفير أساساً لمشكلة تبادل المفاتيح, لكن فيما بعد أصبح شائعاً من أجل استخدامات أخرى مثل البصمة الإلكترونية. الفكرة الأساسية للتشفير اللامتناظر هي في اسمه, مفتاح التشفير مختلف عن مفتاح فك التشفير, و هذا ما سندرسه في بحثنا هذا.

أولاً, سنقوم بالتعريف عن المفاهيم الأساسية للتشفير اللامتناظر. و دراسة آليات و فك تشفير بعض الخوارزميات اللامتناظرة المهمة بالتفصيل, مع شرح أهم الهجومات التي حدثت عليها. كذلك سنمر بالذكر على بعض الخوارزميات الأقل أهمية. ثم سنقوم بشرح آلية خوارزميتنا التي أوجدناها في بحثنا هذا, مع شرح النتاج العملي لمشروعنا.

# كيف تستخدم هذا البحث

في الفصل الأول سنورد مبادئ رياضية يتوجب على القارئ معرفتها لفهم خوارزميات المشروع. و بعد ذلك سنبدأ بمجال التشفير اللامتناظر. ننصح القارئ بأن يبدأ بقراءة البحث من الفصل الثاني, و في حال وجد إبهاماً في الصيغ الرياضية للخوارزميات الواردة في الشرح, بإمكانه العودة إلى الفصل الأول للتعرف على المفاهيم الناقصة, و سنورد رقم المفهوم الرياضي بهدف تسهيل إيجاده.

أما بالنسبة لخوارزمية المشروع, فيجب قراءتها بعد قراءة الفصل الثاني الذي يشرح الخوارزميات اللامتناظرة, نظراً لأن العديد من المفاهيم في خوارزميتنا تعتمد على معلومات واردة في الفصل الثاني.

و بالنسبة لبرنامج المشروع, فسنشرحه مع كتابة الكود. لكن نذكر أن فهم هذا الكود يتطلب معرفة سابقة بلغة JAVA و C++.

نرجو أن تستمتعوا بقراءة هذا البحث بقدر ما استمتعنا بكتابته!

# الفصل الأول : خلفية رياضية

سنقوم في هذا الفصل بتقديم المفاهيم الرياضية التي يتوجب على القارئ معرفتها, حيث تعتمد الخوارزميات التي سنشرحها فيما بعد على هذه المفاهيم بشكل أو بآخر. بهدف السهولة و اختصار الوقت, سنتجاوز ذكر برهان العديد من الخواص التي سنوردها في هذا الفصل.

ننصح القارئ بأن بتجاوز هذا الفصل و يبدأ بقراءة البحث من الفصل الثاني. و في حال رأى معلومة جديدة, بإمكانه العودة إلى هذا الفصل للتعرف عليها.

## مبادئ في نظرية الأعداد

### قابلية القسمة

تعريف 1.1.1 : من أجل عددين صحيحين  نقول أن  يقسم  و نكتب  إذا فقط إذا وجد عدد صحيح  بحيث يكون.

مثال :  و .

خواص قابلية القسمة :

* خاصية 1 : .
* خاصية 2 :  (الانعكاسية) .
* خاصية 3 : .
* خاصية 4 : .
* خاصية 5 :  (التعدية).
* خاصية 6 : .

### الأعداد الأولية

تعريف 1.1.2.1 : يقال عدد صحيح موجب  أنه أولي إذا و فقط إذا كان  لا يقبل القسمة إلا على نفسه و على الواحد.

مثال :  هي مجموعة من الأعداد الأولية.

تعريف 1.1.2.2 : يقال عن عدد  أنه مركب إذا لم يكن أولياً, أي إذا كان يقبل قاسماً غير نفسه و الواحد.

مبرهنة 1.1.2.1 : أي عدد صحيح  يقبل على الأقل قاسماً أولياً.

مبرهنة 1.1.2.2 : أي عدد صحيح مركب  يقبل على الأقل قاسماً أولياً  لا يتعدى .

مبرهنة 1.1.2.3 : يوجد عدد لانهائي من الأعداد الأولية.

مبرهنة 1.1.2.4 (المبرهنة الأساسية في الحساب) : من أجل كل الأعداد الصحيحة الموجبة , توجد الأعداد الأولية الوحيدة  بحيث : .

على الرغم من عدم وجود صيغة لإيجاد الأعداد الأولية, إلا أن كثافتها (أي معدل ظهورها ضمن الأعداد الصحيحة) معروفة منذ حوالي المئة عام.

مبرهنة 1.1.2.5 (مبرهنة الأعداد الأولية) : ليكن  يعبر عن عدد الأعداد الأصغر من . فيكون :



### القاسم المشترك الأكبر و المضاعف المشترك الأصغر

تعريف 1.1.3.1 : ليكن لدينا عددين صحيحين . نقول عن عدد  أنه القاسم المشترك الأكبر و نكتب  إذا و فقط إذا كان  أكبر عدد صحيح يقسم كلا من  و .

مثال :  , .

خواص القاسم المشترك الأكبر :

* خاصية 1 : إذا كان  و  فيكون .
* خاصية 2 : .

تعريف 1.1.3.2 : ليكن لدينا عددين صحيحين . نقول عن عدد  أنه المضاعف المشترك الأصغر و نكتب  إذا و فقط إذا كان  أصغر عدد صحيح يقبل القسمة على كلا من  و .

مثال :  , .

مبرهنة 1.1.3.1 : ليكن لدينا العددين الصحيحين . فيكون لدينا .

تعريف 1.1.3.3 : نقول عن عددين  أنهما أوليان فيما بينهما إذا و فقط إذا كان .

### التطابق الخطي

تطابق الأعداد الصحيحة مفهوم أوجده الرياضي الشهير يوهان كارل فريدريك غاوس, و يهدف هذا المفهوم إلى تسهيل مفهوم قابلية القسمة و بواقي القسمة. و هو كالآتي :

تعريف 1.1.4.1 : يقال عن عددين صحيحين  أنهما متطابقان بالمقاس  و نكتب  إذا و فقط إذا كان .

مثال :  و  و .

* خاصية 1 :  (الانعكاسية).
* خاصية 2 :  (التعدية).
* خاصية 3 : .
* خاصية 4 : .
* خاصية 5 : .

### خوارزمية القسمة[[1]](#footnote-1)

مبرهنة 1.1.5.1 : من أجل كل عددين صحيحين  يوجد عددين صحيحين وحيدين  بحيث. و يدعى  ناتج القسمة (quotient) و  باقي القسمة (remainder-residue).

مثال : من أجل  لدينا . و من أجل  لدينا .

### مبرهنة أولر

تعريف .1.1.6.1: يعرف تابع أولر الذي رمزه  بأنه عدد الأعداد التي تحقق  و .

مثال :  و .

مبرهنة 1.1.6.1 : إذا كان لدينا فيكون:



مثال : لنكتب صيغة تابع أولر للمثال السابق:

بما أن فيكون:



بما أن عدد أولي فيكون :



مبرهنة 1.1.6.2 (مبرهنة أولر) : من أجل كل الأعداد الصحيحة الموجبة  بحيث, العلاقة الآتية محققة :



مثال : من أجل , نلاحظ أن:



### مبرهنة البواقي الصينية[[2]](#footnote-2)

مبرهنة 1.1.7.1 (مبرهنة البواقي الصينية) : لتكن  أعداد صحيحة موجبة غير  و أولية فيما بينها مثنى مثنى, و مهما تكن الأعداد الصحيحة  المختلفة عن تقبل جملة التطابقات الخطية:



حلولاً, و أي اثنين من هذه الحلول متطابقان بالمقاس .

مثال : لنجد حلول جملة التطابقات الخطية الآتية:

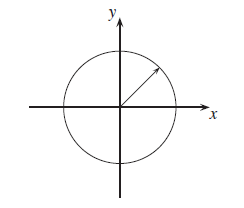
  

في هذه الحالة لدينا , أي يجب أن نجد حلاً لجملة التطابقات الخطية:

و هذا يكافئ إيجاد حل لجملة التطابقات الخطية:

نحصل على  فيكون:



و بمنا أن جميع الحلول متطابقة بالمقاس, يكفي أن نأخذ . و تعطى كل الحلول بالصيغة.

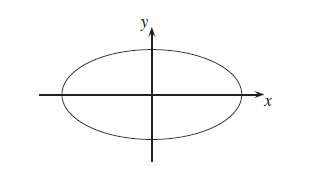
## رياضيات المنحنيات الناقصية[[3]](#footnote-3)

### معادلة منحني ناقصي

سنبدأ باعطاء مقدمة قصيرة حول المبدأ الرياضي للمنحني الناقصي بمعزل عن تطبيقاته التشفيرية. سنبدأ برسم الخط البياني لبعض كثيرات الحدود بمجهولين على الحقل .

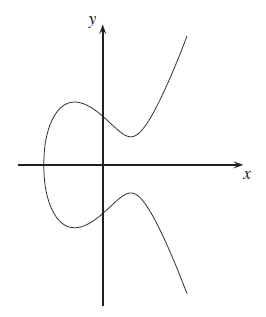
مثال : 

لنرسم المعادلة بتابعية , ستنتج لدينا دائرة كما في الشكل :

لنعمم معادلة الدائرة بحيث تصبح من الشكل، حيث  ثوابت, سينتج لدينا القطع الناقص كما في الشكل :

### تعريف المنحني الناقصي

كما وجدنا في المثالين أعلاه, يمكننا الحصول على أنواع مختلفة من الأشكال بتغيير شكل المعادلة (حيث نعني بالمنحني مجموعة النقاط التي تحقق احداثياتها معادلته) حيث يمكننا بوضوح ملاحظة ان النقطة  تنتمي للدائرة أما النقطة  لا تنتمي لها.

إن المنحني الناقصي هو شكل خاص تحدده معادلة سنذكرها في هذا الفصل, ولكن بسبب حاجتنا له في مجال التشفير لا نعرفه على حقل الأعداد الحقيقية, وإنما على حقل غالوا للأعداد الأولية , وهو الحقل الأكثر شيوعاً للاستخدام في مجال التشفير لأنه من الممكن اجراء العمليات الحسابية فيه بالمقاس.

تعريف 1.2.2.1 : المنحني الناقصي على  هو مجموعة النقاط  التي تحقق المعادلة



حيث .

هندسياً ، لا تعبر المعادلة الموجودة أعلاه عن منحني مستمر وإنما على مجموعة نقاط متناثرة, ولكن هذا لا يمنع أبداً من رسم المنحني على حقل الأعداد الحقيقية من أجل تنفيذ العمليات الهندسية عليه بيسر ثم تحويلها إلى عمليات جبرية تمهيداً لتعريف الزمرة الدائرية.

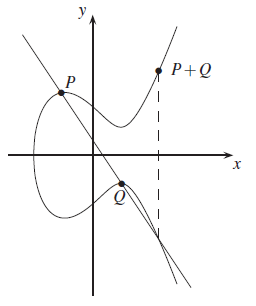
مثال : لنرسم المنحني الناقصي المعرف بالشكل :



يمكنننا ملاحظة الكثير من الأمور المتعلقة برسم المنحني الناقصي. أولاً إن تناظره بالنسبة للمحور لا يجعله تابعاً وإنما اجتماع تابعين هما :

 و 

ثانياً : يقطع المنحني المحور في نقطة واحدة فقط وهذا يدل أن للمعادلة حل حقيقي واحد وحلان عقديان, وهما بالطبع لن يظهرا في مستوي الأعداد الحقيقية, وهذا ما يقتضيه الشرط.

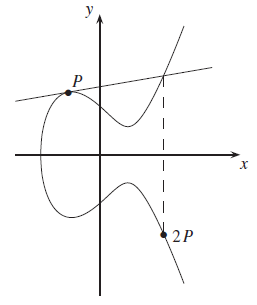
لنعد الآن إلى هدفنا الرئيسي وهو الحصول على زمرة دائرية من المنحنيات الناقصية, نلاحظ أننا في هذا القسم قد قمنا بالمهمة الأولى وهي تعريف عناصر المجموعة التي نبني عليها الزمرة. والسؤال التالي هو, "ما هي العمليات الزمرية المعرفة على هذه النقاط؟"

### العمليات الزمرية على المنحنيات الناقصية

نعرف عملية الجمع ونكتب الرمز  كالتالي : لتكن النقطتان  من المنحني الناقصي, تكون النقطة  من المنحني الناقصي نفسه, وهذا من شروط الزمرة.

بالطبع, إن هذا التعريف ليس عشوائيا. لذلك سنقوم بإعطاء التفسير الهندسي لهذه العمليات ثم تحويلها إلى عمليات جبرية.

جمع نقطتين مختلفتين :

عندما نجمع نقطتين مختلفتين من المنحني ولتكن  نرسم الوتر  فيقطع المنحني في نقطة ولتكن , ولتكن النقطة  صورة  وفق انعكاس بالنسبة للمحور, إن النقطة  هي ناتج الجمع .

مضاعفة نقطة :

عندها تكون , هندسياً يصبح الوتر بين النقطتين مماساً من نقطة واحدة, يقطع المنحني في نقطة, نأخذ النظيرة بالنسبة لـ ، وبالتالي لتكن هذه النقطة هي  بالتالي.

وكما هو معلوم, لا يمكننا التعامل مع العمليات الهندسية في النظام التشفيري, إنما يجب تحويلها إلى الشكل الجبري, ثم تعريفها على حقل غالوا للأعداد الأولية, وهو يعد رياضياً أمر سهل لذلك سنذكر الصيغة مباشرة وهي كالتالي :

الجمع والمضاعفة في زمرة المنحنيات الزائدية :

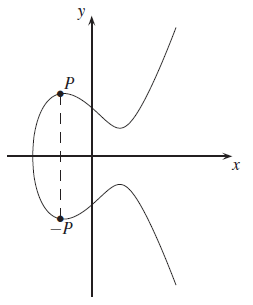
لتكن لدينا النقطتان . حاصل جمع النقطتين هو النقطة  حيث :





حيث  يعطى بالصيغة :



ويعبر عن ميل المستقيم الواصل بين النقطتين في حالة الجمع, أو المماس المرسوم من النقطة في حالة المضاعفة.

العنصر الحيادي : هو العنصر الذي يحقق :



ويعرف النظير كذلك وفق العلاقة :



يعرف العنصر الحيادي بأنه النقطة التي تنتمي لـ  وتقع في اللانهاية. كما تعرف نظيرة نقطة وفق العلاقة:



وهو كما في هذا الشكل:

مثال : ليكن  منحنياً زائدياً. ما هو ناتج مضاعفة النقطة ؟







و منه :



ويمكن ببساطة التأكد من صحة الحسابات بتعويضها في المعادلة الأساسية.

مبرهنة (مبرهنة هاس) : في أي منحنى ناقصي  على المجموعة , تتحقق العلاقة :



## مسائل أساسية في نظرية الأعداد

### مسألة التحليل

إذا كان لدينا عدد صحيح , أوجد تحليله إلى عوامله الأولية. أي, أوجد الأعداد الأولية  و  بحيث يكون  حيث  أعداد أولية مختلفة و .

### مسألة الجذور التربيعية بالمقاس

إذا كان لدينا عدد صحيح مركب و مقاس , أوجد الجذر التربيعي ل بالمقاس, أي أوجد العدد الصحيح  بحيث .

### مسألة اللوغاريتم المتقطع

إذا كان لدينا عدد أولي و العنصر المولد ل  و العنصر , أوجد العدد الصحيح  بحيث .

### مسألة اللوغاريتم المتقطع الموسعة

إذا كان لدينا زمرة دائرية منتهية من الرتبة  و العنصر المولد ل و العنصر , أوجد العدد الصحيح  بحيث.

### مسألة اللوغاريتم المتقطع في المنحنيات الناقصية :

إذا كان المنحني الناقصي. وليكن هو رتبة هذا المنحني. ولتكن نقطة من هذا المنحني. ولتكن كذلك  نقطة أخرى من المنحني. أوجد العدد الصحيح بحيث .

### مسألة ديفي-هيلمان

إذا لدينا عدد أولي و العنصر المولد ل  و العناصر  أوجد العنصر.

### مسألة ديفي-هيلمان الموسعة

إذا كان لدينا زمرة دائرية منتهية من الرتبة  و العنصر المولد ل والعناصر  أوجد العنصر .

### مسألة المجموعة الجزئية

إذا كان لدينا مجموعة من الأعداد الصحيحة  و العدد الصحيح, حدد ما إذا وجدت مجموعة جزئية  بحيث يكون مجموعة عناصر  يساوي.

## مبادئ الجبر المجرد

تلعب نظرية الزمر دوراً مهماً في الرياضيات البحتة والتطبيقية وتتصل كذلك مع الفيزياء والكيمياء اذ تعد الوسيلة التي يمكن من خلالها ترجمة أي فرع جديد في الرياضيات إلى موضوع مفهوم يمكن بسهولة ربطه مع بقية المواضيع ،تنسب نظرية الزمر إلى لاغرانج وفندرموند أما أول من استخدم مصطلح الزمرة أو group فهو الفرنسي ايفرست غالوا عام 1771.

ترتبط نظرية الزمر مع التشفير بشكل كبير إذ أن أي نظام تشفيري غير متناظر يبنى على زمرة دائرية خاصة ،لذلك ينبغي دراسة أسس نظرية الزمر من أجل فهم علاقتها بالتشفير وكيفية الحصول على نظام تشفيري .

### العلاقات الثنائية

تعريف 1.4.1.1 : العلاقة الثنائية هي أي وسيلة تستخدم للربط بين عنصرين لتكون النتيجة هي قضية رياضية. و يرمز لها عادة بالرموز.

مثال :  كلها علاقات ثنائية.

فيما يلي نورد الصفات الأساسية للعلاقات الثنائية.

تعريف 1.4.1.2 : العلاقة الانعكاسية هي علاقة تحقق.

مثال : علاقة قابلية القسمة هي علاقة انعكاسية, لأن .

تعريف 1.4.1.3 : العلاقة التناظرية هي علاقة تحقق.

مثال : علاقة المساواة هي علاقة تناظرية, لأن .

تعريف 1.4.1.4 : العلاقة المتعدية هي علاقة تحقق.

مثال : علاقة التراجح هي علاقة متعدية, لأن .

تعريف 1.4.1.5 : العلاقة التخالفية هي علاقة تحقق.

مثال : علاقة التراجح هي علاقة تخالفية, لأن .

و الآن نورد بعض أنواع العلاقات الثنائية.

تعريف 1.4.1.6 : علاقة التكافؤ هي علاقة انعكاسية و تناظرية و متعدية.

مثال : علاقة التكافؤ بالمقاس هي علاقة تكافؤ.

تعريف 1.4.1.6 : علاقة الترتيب هي علاقة انعكاسية و تخالفية و متعدية.

مثال : علاقة التراجح هي علاقة ترتيب.

تعريف 1.4.1.7 : علاقة الترتيب الجزئي هي علاقة ترتيب على مجموعة تحقق.

مثال : علاقة " مضاعف ل " هي علاقة ترتيب جزئي على المجموعة.

تعريف 1.4.1.8 : علاقة الترتيب الكلي هي علاقة ترتيب على مجموعة  تحقق.

مثال : علاقة التراجح هي علاقة ترتيب كلي على المجموعة.

### قوانين التشكيل (العمليات)

تعريف 1.4.2.1 : قانون التشكيل هو عملية على عنصر أو أكثر لنحصل على عنصر جديد. و يرمز لها عادة .

مثال : العمليات هي قوانين تشكيل.

تعريف 1.4.2.2 : قانون التشكيل الداخلي هو قانون تشكيل معرف على مجموعة بحيث.

مثال : قانون التشكيل  هو قانون تشكيل داخلي على المجموعة.

من الآن فصاعداً سندعو "قانون التشكيل" ب"العملية" للسهولة.

تعريف 1.4.2.3 : العملية التبديلية هي عملية معرفة على مجموعة على تحقق.

مثال : عملية  المعرفة على المجموعة هي عملية تبديلية.

تعريف 1.4.2.3 : العملية التجميعية هي عملية معرفة على مجموعة على تحقق.

مثال : عملية  المعرفة على المجموعة هي عملية تجميعية.

### البنى الجبرية

تعريف 1.4.3.1 : البنية الجبرية هي ثنائيةبحيثمجموعة و قانون تشكيل داخلي على.

مثال :  هي بنية جبرية.

تعريف 1.4.3.2 : لتكن لدينا البنية الجبرية, العنصر الحيادي اليميني بالنسبة للعملية هو عنصر يحقق . و كذلك العنصر الحيادي اليساري بالنسبة للعملية هو عنصر يحقق .

تعريف 1.4.3.3 : لتكن لدينا البنية الجبرية, العنصر الحيادي هو عنصر و هو حيادي يميني و حيادي يساري.

مثال : العنصر هو حيادي بالنسبة للعملية  في البنية.

تعريف 1.4.3.4 : لتكن لدينا البنية الجبرية, نظير العنصر بالنسبة للعملية هو عنصر يحقق.

مثال : العنصر هو نظير العنصر في البنية.

تعريف 1.4.3.5 : الزمرة هي البنية جبرية التي تحقق :

*  عملية تبديلية.
*  عملية تجميعية.
* لكل عنصر نظير  بالنسبة للعملية.

مثال :  هي زمرة.

تعريف 1.4.3.6 : الزمرة التبديلية هي زمرة بحيث عملية تبديلية.

مثال :  هي زمرة تبديلية.

تعريف 1.4.3.7 : الزمرة المنتهية هي زمرة  فيها .

تعريف 1.4.3.8 : رتبة زمرة هي عدد صحيح موجب .

تعريف 1.4.3.9 : لتكن لدينا زمرة . رتبة العنصر  هي أصغر عدد صحيح  يحقق .

مثال : لتكن لدينا الزمرة . رتبة العنصر  هي  لأن .

مبرهنة 1.4.3.1 : رتبة زمرة ما  هي أصغر عدد صحيح  يحقق .

مبرهنة 1.4.3.2 (مبرهنة لاغرانج) : رتبة أي عنصر في زمرة تقسم رتبة الزمرة.

تعريف : لتكن لدينا الزمرتان  و . نقول عن الزمرة  أنها زمرة جزئية من الزمرة  و نكتب  إذا و فقط إذا .

خواص الزمر الجزئية :

1. حيادي الزمرة  هو نفسه حيادي الزمرة .
2. نظير أي عنصر من  في  هو ذاته في .

مثال :  هي زمرة جزئية من .

مبرهنة : الشرط اللازم و الكافي لتكون  زمرة جزئية من  هو :



مبرهنة : الشرط اللازم و الكافي لتكون  زمرة جزئية من  هو :



### المجموعة المولدة لزمرة

تعريف : نقول عن المجموعة  أنها مجموعة مولدة للزمرة  إذا :





### الزمرة الدائرية المولدة بمجموعة

لتكن لدينا الزمرة . إذا احتوت المجموعة  المولدة للزمرة  على عنصر واحد وهو أي أن :



حيث الزمرة  تبديلية بالضرورة .

مثال : ليكن  مولد الزمرة , بالتالي يمكن كتابة 

## خوارزميات أساسية

### الخوارزمية الإقليدية

الخوارزمية الإقليدية هي الطريقة الأمثل و الأكثر كفاءة لحساب القاسم المشترك الأكبر لعددين معلومين. و هي مبينة على الحقيقة الآتية :

مبرهنة : .

نكرر تطبيق هذه الخاصية لنحصل على :



طالما , أي بكلمات أخرى :



لندعو  ب, نكرر العملية مرة أخرى :



و بعد عدد نهائي من العمليات المماثلة نحصل على :



مثال : لنوجد القاسم المشترك الأكبر للعدين .

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

الخوارزمية 1.5.1 : الخوارزمية الإقليدية :

الدخل : العددين الصحيحين .

الخرج : .

الخوارزمية :



### الخوارزمية الإقليدية الممددة[[4]](#footnote-4)

الخوارزمية الإقليدية الممدة هي الخوارزيمة الأمثل إذا أردنا أن نحسب المعكوسات الضربية بالمقاس . فهي, إلى جانب حساب القاسم المشترك لعددين معلومين , تعبر عنه كتركيب خطي ل و بالصيغة:



السؤال هو, كيف نقوم بحساب الأمثال  ؟ الفكرة وراء الخوارزمية الممدة هي تنفيذ الخوارزمية الإقليدية, لكن, بكل مرحلة, نعبر عن الباقي  كتركيب خطي ل و  بالصيغة:



مثال : سننظر إلى تنفيذ الخوارزمية الإقليدية الممدة بنفس القيم السابقة.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| هنا انتهى تنفيذ الخوارزمية |  |  |  |  |  |  |

الخوارزمية 1.5.2 : الخوارزمية الإقليدية الممددة :

الدخل : الأعداد الصحيحة  حيث .

الخرج : الأعداد الصحيحة  بحيث .

الخوارزمية :



### حساب الأس[[5]](#footnote-5)

الطريقة الأبسط لحساب الأس هي كالتالي :



حيث تدل  على عملية ضرب العدد الحالي بالأساس . لكن تذكر أن الأعداد المستخدمة في الخوارزميات اللامتناظرة في الحالة العامة أعداد عملاقة, 1024 بت و أكثر. إذا أردنا حساب الأس السابق بالطريقة البسيطة, فسنحتاج إلى حوالي  عملية ضرب. يالها من فكرة مرعبة, لأن يوجد في الكون المعروف حوالي  ذرة! السؤال هو, هل هناك طريقة أسرع لحساب الأس بدون أن نضطر إلى انتظار الشمس لكي تنطفئ لننهي الحساب؟ لحسن الحظ, نعم. إنها تدعى خوارزمية الضرب و التربيع.

الخوارزمية 1.5.3 : خوارزمية الضرب و التربيع :

الدخل : عددين . عدد  بتمثيله الثنائي .

الخرج : .

الخوارزمية :



مثال : لنحسب .

لو أردنا أن نحسب هذا الأس بالطريقة "الساذجة" لتطلب الأمر 28 عملية ضرب. لنحسب الأس بطريقة الضرب و التربيع. أولا نلاحظ أن . بتطبيق الخوارزمية 2.1 نحصل على السلسة الآتية من عمليات الضرب و التربيع :



ما مجموعه عمليتا ضرب و أربع عمليات تربيع. في الواقع, تعقيد عملية التربيع يقارب تعقيد عملية الضرب, فبإمكاننا أن نقول أن كلفة هذا التنفيذ هي 6 عمليات ضرب فقط.

أما بالنسبة لتعقيد خوارزمية الضرب و التربيع. نلاحظ أنه في كل مرحلة من تنفيذ حلقة  يتم تربيع النتيجة الحالية. و بما أن حلقة  تتكرر  مرة حيث  هو طول الأس بالنظام الثنائي, فلدينا  عملية تربيع. إضاف إلى ذلك, يتم ضرب النتيجة الحالية بالأساس  مع كل  في التمثيل الثنائي للأس. بشكل عام, لدينا حوالي  رقم  في التمثيل الثنائي لعدد طوله  بت. فيكون تعقيد تنفيذ الخوارزمية هو . و كما قلنا سابقاً, تعقيد عملية التربيع يقارب تعقيد عملية الضرب, فبإمكاننا أن نقول أن تعقيد الخوارزمية هو . و هو أفضل بكثير من  لو أردنا تنفيذ الخوارزمية بالطريقة البدائية.

### حساب مجموع نقطة في منحني ناقصي

بالمقارنة مع نظام RSA, تم وضع خوارزمية للوصول إلى العدد المطلوب بسرعة وهي خوارزمية الضرب والتربيع. والآن نحن أيضاً بحاجة لخوارزمية تساعدنا في الوصول إلى النقطة المطلوبة في نظام ECC وهي خوارزمية المضاعفة والجمع. وسنعتمد كما قمنا في خوارزمية الضرب والتربيع على الكتابة الثنائية للعدد .

الخوارزمية 1.5.4 : خوارزمية الجمع و المضاعفة :

الدخل : نقطة  من منحني ناقصي . العدد .

الخرج : .

الخوارزمية :



مثال : لحساب النقطة، في البداية, نكتب العدد 26 ثنائياً . فيصبح تطبيق الخوارزمية كالتالي :



هذا يشابه خوارزمية الضرب و التربيع. و أيضاً, لولا هذه الخوارزمية لكنا قد احتجنا  عملية جمع.

### اختبارات الأولية

يتم توليد الأعداد الأولية بطريقة غريبة نوعاً ما, يتم اختيار عدد عشوائي بطول محدد (عادة 512 بت), ثم نختبر ما إذا كان أولياً أم لا. و كلنا نعلم أن كثافة الأعداد الأولية تتناقص كلما تزايد العدد. و هنا يأتي التساؤل, ألا يزال هناك احتمال منطقي لأن نجد أعداداً أولية بتلك الأطوال العملاقة عن طريق التجريب؟ حسب المبرهنة 1.1.2.5 احتمال أن يكون عدد ما بطول  بت أولياً هو :

.

أي نحن بحاجة لكي نختبر حوالي 177 عدد قبل أن نحصل على عدد أولي يطول  بت. لحسن الحظ, مقام الكسر أعلاه يتزايد تزايداً لوغارتمياً, أي حتى مع مفاتيح RSA الأكبر, مثل  بت, لا تزال هناك فرصة لنجد أعداداً أولية.

بعد اختبار العدد العشوائي, يأتي دور اختبارات الأولية. اختبار الأولية هو خوارزمية بمدخلين, عدد  يدعى "مرشح أولي". و معامل أمان . و تقوم الخوارزمية بالحكم على  ما إذا كان أولياً أم لا. يوجد نوعين لاختبارات الأولية :

1. اختبارات الأولية الصحيحة : و هي تعيد نتيجة قاطعة "لا" إذا لم يكن أولياً. أو نتيجة قاطعة "نعم" إذا كان أولياً.
2. اختبارات الأولية الاحتمالية : و هي تعيد نتيجة قاطعة "لا" إذا لم يكن أولياً. أو نتيجة "نعم" باحتمال كبير إذا كان أولياً.

سنتعامل الآن مع اختبارات الأولية الاحتمالية. في الواقع, اختبارات الأولية الاحتمالية هي التي يتم استخدامها في العالم الواقعي نظراً لكلفتها الحسابية المنخفضة مقارنة بالاختبارات الصحيحة.

#### اختبار فيرما

يعتمد اختبار فيرما على المبرهنة 1.1.6.2 و هو كالتالي :

لكن يوجد عيب في اختبار فيرما. توجد بعض الأعداد الصحيحة  التي تحقق  من أجل العديد من قيم  لكنها ليست أعداد أولية, و تدعى "أعداد كاذبة". عدد الأعداد الكاذبة على اختبار فيرما كبير نسبياً. في الواقع, توجد أعداد صحيحة  تدعى "أعداد كارمايكل" تحقق العلاقة  من أجل جميع قيم  لكنها ليست أولية. يوجد حوالي  عدد كارمايكل أصغر من . لذلك, لا يتم اعتماد اختبار فيرما.

الخوارزمية 1.5.5.1 : اختبار فيرما :

الدخل : مرشح أولي . معامل أمان .

الخرج : " عدد مركب" أو " غالباً عدد أولي".

الخوارزمية :



#### اختبار ميلر-رابين

يوجد العديد من الاختبارات الاحتمالية الأخرى, مثل اختبار سولوفاي-ستراسين الشهير. لكن سنكتفي باختبار ميلر-رابين نظراً لكونه الاختبار الأكثر كفاءة و فاعلية.

يعتمد اختبار ميلر-رابين على الحقيقة الآتية :

مبرهنة 1.5.5.2 : ليكن  عدد أولي فردي و  حيث  عدد فردي. و ليكن  أي عدد صحيح بحيث . عندئذ إما  أو  من أجل عدد صحيح

بالاعتماد على المبرهن السابقة, نستطيع صياغة اختبار ميلر-رابين.

الخوارزمية 1.5.5.2 : اختبار ميلر رابين :

الدخل : مرشح أولي . معامل أمان .

الخرج : " عدد مركب" أو " غالباً عدد أولي".

الخوارزمية :



# الفصل الثاني : بعض الخوارزميات اللامتناظرة

## خوارزمية RSA

حالياً, أوسع نظم التشفير استخداماً هو نظام RSA. فقد تم اعتماد خوارزمية RSA في الولايات المتحدة الأمريكية كالخوارزمية اللامتناظرة المعيارية حتى عام 2000. لكن في الوقت الحالي, بدأت الخوارزميات الأخرى, مثل خوارزمية المنحنيات القطعية باكتساب الأفضلية على خوارزمية RSA.

توجد العديد من التطبيقات لخوارزمية RSA, لكنها تستعمل عادة من أجل :

تشفير كتل صغيرة من البيانات, خاصة في تبادل المفاتيح.

البصمات الإلكترونية, و هي موضوع لن نتطرق له في مشروعنا هذا.

في قلب خوارزمية RSA توجد المسألة 1.3.1 تحليل الأعداد الصحيحة و هي تمثل تابع الجهة الواحدة في هذه الخوارزمية. و تلعب مبرهنة أولر و تابع أولر دوراً هاماً في تنفيذ هذه الخوارزمية.

في هذا الفصل, سنشرح آلية عمل تشفير و فك تشفير و توليد مفاتيح RSA, ثم سنتحدث عن بعض الجوانب التطبيقية لتنفيذ الخوارزمية في الحياة الواقعية.

### توليد المفاتيح

توجد خاصية مميزة لجميع خوارزميات التشفير اللامتناظرة, و هي وجود مرحلة تحضير, يتم أثنائها حساب المفتاحين الخاص و العام. و قد يكون في بعض الخوارزميات شديد التعقيد. نذكر هنا أن توليد المفتاح ليس مسألة هامة في الخوارزميات المتناظرة, و هذا أحد أسباب كونها أسرع في التنفيذ من الخوارزميات اللامتناظرة.

توليد المفاتيح لخوارزمية RSA :

1. اختر عددين أوليين كبيرين . (باستخدام الخوارزمية 1.4.4 أو 1.4.5)
2. احسب .
3. احسب .
4. اختر المفتاح العام . بحيث .
5. احسب المفتاح الخاص بحيث .

سنقوم بشرح كيفية توليد العددين الأوليين فيما بعد. الهدف من الشرط  في الخطوة الرابعة ضمان وجود نظير ل بالنسبة لعملية الضرب بالمقاس. في الواقع, بإمكاننا حساب الخطوتين الرابعة و الخامسة في الوقت نفسه, نقوم باختيار عدد  و ننفذ الخوارزمية 1.5.2 من أجل الأعداد . فنحصل على الصيغة :

.

فإذا كان فعلا  نكون قد حصلنا على المفتاح العام. بعد أن نأخذ طرفي العلاقة السابقة بالمقاس  نحصل على العلاقة :



فتكون قيمة  هي قيمة  التي حصلنا عليها من خرج الخوارزمية الإقليدية الممدة.

أما في حال , ببساطة نعيد تنفيذ الخوارزمية من أجل قيم أخرى ل. نلاحظ أن قيمة المعامل  الذي حصلنا عليه من خرج الخوارزمية ليست ذات قيمة, و بالتالي بإمكاننا أن نتجنب حسابه.

الآن سنقدم مثالاً بسيطاً عن تنفيذ خوارزمية RSA بأعداد صغيرة.

### التشفير و فك التشفير

يحدث تشفير و فك تشفير RSA في الحلقة (الملحق ب) و تلعب التطابقات الخطية دوراً هاماً فيها (1.1.4). تشفر RSA البيانات الأصلية , حيث  هي القيمة العددية لسلسلة الأصفار و الواحدات إذا عاملناها معاملة عدد مكتوب بالنظام الثنائي, فيكون. نستنتج أنه يتوجب على  أن يكون محصوراً في المجال  و كذلك الأمر بالنسبة للبيانات المشفرة .

سنورد الآن آلية تشفير RSA بواسطة المفتاح العام, و آلية فك التشفير بالمفتاح الخاص.

تشفير RSA :

ليكن لدينا المفتاح العام  و النص الأصلي . يعطى تابع التشفير  بالصيغة :



باستخدام الخوارزمية 1.5.3 حيث .

أما آلية فك التشفير :

فك تشفير RSA :

ليكن لدينا المفتاح الخاص  و النص المشفر . يعطى تابع فك التشفير  بالصيغة :



باستخدام الخوارزمية 1.5.3 حيث .

تطبيقياً, الأعداد  كبيرة جداً, عادة أكثر من 1024 بت. و عادة ما يدعى  ب"الأس العام" أو "أس التشفير", و يدعى  ب"الأس الخاص" أو "أس فك التشفير".

إذا أراد مصطفى أن يرسل رسالة  إلى محمود, ينبغي على مصطفى الحصول على مفتاح محمود العام , ثم يقوم بحساب , هي الرسالة المشفرة التي سيرسلها في الشبكة. و على الجهة الأخرى يقوم محمود بفك التشفير عن طريق حساب.

بمعلوماتنا المحدودة هذه, نستطيع أن نستنتج بعض متطلبات تنفيذ الخوارزمية :

1. بما أن بإمكان المهاجم الحصول على المفتاح العام من الشبكة, يجب أن يكون من المستحيل حسابياً تحديد قيمة المفتاح الخاص من المفتاح العام.
2. بما أن النص الأصلي مميز حتى الوصول إلى حجم المقاس  , فنحن لا نستطيع أن نشفر رسالة أطول من بت, حيث هو عدد منازل العدد إذا كتب في النظام الثنائي.
3. من أجل قيمة محددة ل, يلزم وجود عدد كبير من أزواج المفاتيح العامةو الخاصة. و إلا سيقوم المهاجم ببساطة بشن هجوم بالقوة الغاشمة و يجرب جميع القيم الممكنة للمفاتيح (سنرى فيما بعد أنه يسهل تحقيق هذا الشرط).

الآن سنقدم مثالاً بسيطاً عن تنفيذ خوارزمية RSA بأعداد صغيرة.

مثال : يريد مصطفى أن يرسل رسالة مشفرة إلى محمود.

أولاً : يقوم محمود بحساب معاملات RSA عن طريق خطوات خوارزمية توليد المفتاح من 1 إلى 5, ثم ينشر في الشبكة المفتاح العام الذي يحصل عليه مصطفى. يقوم مصطفى بتشفير رسالته , و يرسل النص المشفر إلى محمود عبر الشبكة. أخيراً, يقوم محمود بفك تشفير رسالة مصطفى بعد أن تصله بواسطة مفتاحه الخاص.

مصطفى : محمود :









في الواقع, الأعداد التي أوردناها في المثال السابق حتى لا تقترب من الوصول إلى حجم المفاتيح المستخدمة في الواقع. على سبيل المثال, نورد هذه المعاملات التي هي بالطول الكامل :

المثير للاهتمام في تنفيذ الخوارزمية, هو أن في عملية التشفير, رفعنا الرسالة  إلى الأس  لتظهر معنا الرسالة المشفرة . و بعد ذلك, في عملية فك التشفير, رفعنا الرسالة المشفرة  إلى الأس  فكانت النتيجة هي الرسالة الأصلية . لنكتب ذلك بالمعادلات :



هذه العلاقة هي جوهر RSA. لنبرهن صحة هذه العلاقة.

### برهان صحة خوارزمية RSA

سنبرهن أن تابع فك التشفير هو التابع العكسي لتابع التشفير. أي سنبرهن صحة العلاقة :



لبرهان صحة هذه العلاقة, نعود إلى تعريف المفتاح الخاص , التي تقول أن . من تعريف التطابق الخطي (الملحق أ) هذه يكافئ :



من أجل عدد صحيح . نعوض في العلاقة الأساسية :



من هنا سنناقش حالتين :

الحالة الأولى : . نطبق مبرهنة أولر (الملحق أ) لنحصل على العلاقة :



و هو المطلوب.

الحالة الثانية : . بما أن , و كلا من  أعداد أولية, فلابد أن يكون هناك واحد منهم يقسم . لنفرض بدون فقدان العمومية أن , فيكون . بالاعتماد على مبرهنة أولر :



لدينا :



بالاعتماد على تعريف التطابق الخطي, هذا يكافئ :











و هو المطلوب.

### تقنيات تسريع تنفيذ خوارزمية RSA

#### التشفير السريع باستخدام المفاتيح العامة الصغيرة

و هي إحدى الطرق الفعالة لتسريع تنفيذ RSA. و هي تستخدم في تشفير RSA و تأكيد البصمات الإلكترونية. و ببساطة, في هذه التقنية, نقوم باختيار المفتاح العام  ليكون قيمة صغيرة. تطبيقياً, غالباً ما نستخدم القيم  فتكون كلفة عمليات التشفير كالآتي :

|  |  |  |
| --- | --- | --- |
| المفتاح العام | التمثيل الثنائي ل |  |
|  |  |  |
|  |  |  |
|  |  |  |

لاحظ أن المفاتيح الواردة أعلاه ذات عدد قليل من الواحدات في التمثيل الثنائي, و هذا يفيد في تقليل عدد عمليات الضرب المنفذة. في الواقع, اختيار المفاتيح العامة الصغيرة لا يؤثر على أمان خوارزمية RSA, نظراً لكونه عاماً و لا داعي لجعله مكلف حسابياً. لكن, في هذه الحالات, لا يزال المفتاح الخاص  بطوله الكامل.

نتيجة اتباع هذه التقنية هي أن عمليات التشفير و تأكيد البصمات الإلكترونية تكون أسرع بشكل ملحوظ. في الواقع, هي تجعل خوارزمية RSA الخوارزمية اللامتناظرة الأسرع في هاتين العمليتين. لسوء الحظ, لا يمكننا اتباع هذه الطريقة من أجل المفتاح الخاص , أي من أجل فك التشفير و توليد البصمات الإلكترونية, بدون تعريض أمن الخوارزمية للخطر. و نتيجة لذلك توجد بعض الخوارزميات الأخرى الأسرع من RSA في هاتين العمليتين (مثل خوارزميات المنحنيات القطعية).

#### فك التشفير السريع بمبرهنة البواقي الصينية

تعتمد هذه التقنية على المبرهنة 1.1.7. الفكرة الأساسية هنا, هي أن الطرف الذي يملك المفتاح الخاص , أيضاً يملك الأعداد . فبإمكانه أن يقوم بحساب الأس بالمقاس بمقاسين صغيرين بدلاً من مقاس واحد كبير.

الخطوة 1 :

نقوم بإرجاع الرسالة  بالمقاسين  لنحصل على :





و كذلك بالنسبة للأس  :





الخطوة 2 :

نقوم بحساب الأسين بالنسخ المرجعة من الرسالة  لنحصل على





نلاحظ أن كلا من الأعداد  محدودة بالأرقام . و بما أننا نختار الأعداد الأولية  ليكون لها حوالي نفس الطول, فلكل من الأعداد السابقة حوالي نصف طول .

الخطوة 3 :

نقوم لحساب النظائر الآتية :





نقوم بتركيب النتيجة  من نسخاتها المرجعة . حسب مبرهنة البواقي الصينية, تعطى النتيجة  بالعلاقة :



مثال : لتكن معاملات RSA كالآتي :





سنقوم الآن بتنفيذ فك التشفير بتقنية البواقي الصينية من أجل النص المشفر .

الخطوة 1 :









الخطوة 2 :





الخطوة 3 :







أما بالنسبة لتعقيد عملية فك التشفير بتقنية البواقي الصينية. يكفي لحساب كلفة العملية أن نحسب كلفة الخطوة الثانية, نظراً لسهولة العمليات الحسابية في الخطوات الأخرى. لنفرض أن طول  بالتمثيل الثنائي هو , فيكون طول كل من  هو . فتكون كلفة كل حساب أس في الخطوة الثانية هي  من أجل ثابت ما . و تكون كلفة الخطوة الثانية, و بالتالي تقريباً كل التنفيذ هي . لكن, تذكر أن تعقيد عملية الضرب يتزايد تربيعياً مع عدد منازل الأعداد المضروبة. و بما أن كل من الأعداد السابقة بحوالي نصف طول , فأن كلفة تنفيذ عملية فك التشفير بتقنية البواقي الصينية أقل بأربع مرات من تنفيذها بالطريقة العادية.

### أمن RSA

ما شرحناه إلى حد الآن عن خوارزمية RSA هو ما يدعى "الخوارزمية التعليمية". لكن يوجد في الخوارزمية التعليمية العديد من النقاط الضعف التي استخدمها علماء التشفير لشن هجومات متعددة على RSA. سنقوم في هذه القسم بشرح بعض هذه الهجومات و طرق تفاديها.

#### علاقة الأمن بالتحليل

أراد مهاجم أن يقوم بحساب المفتاح الخاص  من المفتاح العام  و المقاس . اعتماداً على تعريف المفتاح الخاص , عليه أن يقوم بحساب نظير  الضربي بالمقاس . و هو بحاجة لمعرفة  لحسابه. و لمعرفة  يجب عليه أن يملك العددين . فيبدو أن بإمكانه أن يشن هجوم على RSA بمحاولة تحليل المقاس . بعد أن يقوم بذلك, يقوم بحساب , و بالتالي بإمكانه حساب  بنفس الطريقة التي حسبه بها الطرف الأوسط. هذا ما يدعى مسألة RSA.

مبرهنة 2.1.5.1 : مسألة الحصول على المفتاح الخاص  من المفتاح العام  و المقاس  (مسألة RSA) و مسألة تحليل المقاس  إلى عوامله الأولية متكافئتان حسابياً.

برهان هذه المبرهنة يتطلب معلومات متقدمة في دراسة تعقيد الخوارزميات, سنتجاوز ذكره نظراً لطوله.

بما أننا اخترنا العددين  كبيرين بحيث أنه من المستحيل حسابياً تحليل  إلى عوامله الأولية, فمن المستحيل حسابياً نجاح هذه الهجوم. لكن, لم يتم إثبات أن كسر RSA و تحليل  عمليتان متكافئتان حسابياً حتى اللحظة هذه.

#### المفاتيح العامة و الرسائل الصغيرة

كما ناقشنا سابقاً في 1.5.2, بإمكاننا أن نسرع عملية التشفير عن طريق اختيار المفتاح العام  ليكون صغيراً. و من المتعارف عليه اختيار . ليكن لدينا طرف يريد إرسال الرسالة  إلى ثلاثة أطراف مختلفة, و يستخدم المفتاح  من أجل عمليات التشفير الثلاث. فيقوم هذا الطرق بإرسال  (يجب أن تكون المقاسات الثلاث مختلفة). و لكن, بإمكان متنصت أن يكتب جملة التطابقات الخطية الآتية :







و بإمكانه أن يحل الجملة السابقة اعتماداً على مبرهنة البواقي الصينية ليحصل على . و بما أن , فلابد أن يكون . و بالتالي بإمكانه أن يحصل على  بحساب الجذر التكعيبي ل.

لكي نتجنب حدوث هذا الهجوم, بإمكاننا أن نزيد على  سلسة شبه عشوائية من البتات قبل عملية التشفير. و من أجل كل طرف نريد إن نرسل رسالة له, نولد سلسة مختلفة من البتات, و هو ما يدعى ب"تمليح" الرسالة.

و كذلك, لا يجب أن نستخدم مفاتيح التشفير الصغيرة من أجل الرسائل القصيرة مقارنة مع المفتاح العام. فإذا كان , فبإمكان المهاجم أن يسترجع الرسالة الأصلية عن طريق حساب . بإمكاننا أن نتجنب هذه المشكلة عن طريق "تمليح" الرسالة.

الرسائل القصيرة بغض النظر عن المفتاح العام تشكل خطراً. فإذا كان عدد الرسائل الممكنة من طول محدد قصير نسبياً أو حتى متوقع, بإمكان مهاجم ببساطة أن يشفر كل الاحتمالات الممكنة حتى يحصل على الرسالة المشفير التي رآها بالشبكة. أيضاً, تمليح الرسالة أحد طرق تفادي هذه المشكلة.

#### المفاتيح الخاصة الصغيرة

بنفس الطريقة التي سرعنا فيها تنفيذ الخوارزمية من أجل عملية التشفير, بإمكاننا أن نسرع عملية فك التشفير باختيار أس فك التشفير صغيراً.[[6]](#footnote-6) لكن, في حال كان  صغيرا -و هذه هي الحالة العامة- , و كان  أقصر من ربع  بالتمثيل الثنائي, فتوجد خوارزمية لحساب المفتاح الخاص  قام واينر بدراستها في رسالته. لذلك ينصح بأن يكون طول المفتاح الخاص يقارب طول المقاس .

#### المقاسات المشتركة

عادة ما يتم اقتراح وجود طرف موثوق في شبكة ما, و يقوم هذا الطرف بحساب المقاس , و هذا المقاس واحد لجميع الأطراف في هذه الشبكة. و يتم توزيع ثنائيات  لكل طرفين في هذه الشبكة.

لكن, توجد طريقة تمكننا من تحليل  عن طريق معرفة الثنائية . و بالتالي, إذا عرف طرف واحد تحليل , فقد عرفه من أجل جميع الأطراف في هذه الشبكة.

الهدف من هذه الطريقة هو جعل المراسلات عبر شبكة ما أكثر كفاءة و سرعة, لكنها غير آمنة. لذا يجب على كل طرفين اتخاذ مقاس  خاص بهما.

#### الرسائل اللامخفية

يقال عن رسالة أنها لامخفية إذا كانت تشفر لتعطي الرسالة نفسها, أي . دائماً توجد بعض الرسائل اللامخفية في نظام RSA مهما كانت معملاته. على سبيل المثال  هي رسائل لا مخفية. بإمكاننا أن نعرف عدد الرسائل اللامخفية في نظام RSA بمعملات ما بالصيغة :



فدائماً يكون عدد الرسائل اللامخفية هو على الأقل . فإذا كانت الأعداد  عشوائية, و المفتاح العام صغير , فيكون عدد الرسائل اللامخفية صغير و مهمل أمام عدد الرسائل الممكنة. و بالتالي لا تشكل الرسائل اللامخفية خطر على أمن RSA.

## تبادل مفتاح ديفي-هيلمان

جاءت هذه الخوارزمية من ويتفيلد ديفي و مارتن هيلمان, و هما مؤسسا التشفير اللامتناظر. و اقترحا هذه الخوارزمية التي تعتبر أول خوارزمية لامتناظرة.

في وقتنا الحالي, تبادل مفتاح ديفي-هيلمان هو الخوارزمية المعيارية في العديد من بروتوكولات الانترنت. مثل (Internet Protocol Security) و (Secure Shell). يستخدم تبادل مفتاح ديفي-هيلمان كحل لمشكلة تبادل المفاتيح. و يعتبره البعض الحل الأكثر كفاءة, لكن لا يزال البعض الآخر يفضل خوارزميات أخرى نظراً لكونها أوسع استخداما منه, مثل RSA التي تستخدم إلى جانب تبادل المفاتيح, من أجل البصمات الإلكترونية.

في قلب تبادل مفتاح ديفي-هيلمان توجد المسألة 1.3.3 و هي تابع الجهة الواحدة فيه. سنقوم الآن بشرح آلية عمل هذا التبادل. و سنناقش أمن هذه الخوارزمية.

### آلية عمل تبادل مفتاح ديفي-هيلمان

يقوم تبادل مفتاح ديفي هيلمان على مرحلتين. المرحلة التحضيرية و المرحلة الأساسية.

المرحلة التحضيرية :

1. اختر عدداً أولياً كبيراً .
2. اختر عدداً صحيحاً.
3. انشر  و .

عادة ما يقوم بالمرحلة التحضيرية طرف وسطي موثوق. في حال حصل طرفان (محمود و مصطفى) على المعاملات السابقة, بإمكانهما أن يتبادلاً مفتاحاً كما يلي :

المرحلة الأساسية :

مصطفى محمود

**

* *

### برهان صحة تبادل مفتاح ديفي هيلمان

قيمة  التي حسبها محمود هي :



و القيمة التي حسبها مصطفى هي :



إذا هما متفقان على المفتاح نفسه. فيما بعد, سيستخدمان هذا المفتاح من أجل التشفير المتناظر ب AES أو أي خوارزمية متناظرة أخرى. و لكن, إذا أراد مهاجم ما أن يحسب المفتاح من المعطيات الموجودة في الشبكة, و هي  و , فعليه أن يحسب  من  و . و لكن هذه هي مسألة اللوغاريتم, و حسابه سيكون شديد الكلفة الحسابية.

مثال : ليكن .

مصطفى محمود

**

* *

## خوارزميات المنحنيات الناقصية

التشفير بالمنحنيات الناقصية هو العضو الأحدث في عائلة الخوارزميات التي تستخدم المفتاح العام, وقد ظهرت في منتصف الثمانينات.

تزودنا ECC بنفس درجة الأمان التي تزودنا بها RSA أو أنظمة اللوغاريتمات المتقطعة, بالنسبة للسلاسل التي يتراوح طولها بين (160-3072)بت. تبنى الـ ECC أساساً على مسألة لوغاريتمات متقطعة تنجز تبادل مفتاح ديفي - هيلمان وغيره عن طريق المنحنيات الناقصية.

وفي حالات كثيرة يكون للـ ECC فوائد عديدة مثل قلة الحسابات وصغر طول سلسلة المفاتيح العامة والخاصة مقارنة مع غيرها.

إن الرياضيات المتعلقة بالـECC تعد أكثر تعقيداً إذا ما قورنت بـ RSA و DL, حيث تتضمن بعض المواضيع المتقدمة كعد نقاط قطع ناقص تحقق خاصة معينة. ولكننا سنبتعد عن الرياضيات المعقدة إذ تركنا للقارئ فرصة المحاولة بالرجوع إلى (الخلفية الرياضية للمشروع).

### آلية تبادل مفتاح ديفي ـ هيلمان في المنحنيات الناقصية

كما في تبادل مفتاح ديفي-هيلمان العادي, ينفذ تبادل مفتاح ديفي-هيلمان على مرحلتين, رحلة التحضير و المرحلة الأساسية.

المرحلة التحضيرية :

1. اختر عدداً أولياً كبيراً .
2. اختر منحنياً ناقصياً .
3. اختر نقطة ابتدائية .
4. انشر .

وليكن بالعلم أن عملية اختيار منحن ناقصي هي عملية صعبة للغاية, إذ يجب ان يتمتع المنحني ببعض الخواص لكي يكون أكثر اماناً، وسنتكلم أكثر في هذا الموضوع. والآن سنشرح عملية تبادل المفتاح .

المرحلة الأساسية :

مصطفى محمود

**

* *

### برهان صحة تبادل مفتاح ديفي-هيلمان في المنحنيان الناقصية

ببساطة, القيمة التي حسبها محمود هي :



و القيمة التي حسبها مصطفى :



و من خواص الزمرة الدائرية :  و هو المطلوب.

مثال : ليكن لدينا  و .

مصطفى محمود

**

* *

إن السبب الذي جعلنا نستخدم ECC كخوارزمية تشفير هو أن ECDLP تعد مسألة جيدة من حيث البارامترات, فإذا أراد المهاجم أن يكسر ECDH من خلال المعطيات المتوفرة, و هي  فإن عليه أن بحل إحدى المسألتين :



فإذا كان المنحني قد اختير بعناية, ستكون أفضل طريقة لكسر ECDLP ضعيفة أمامها, وهي بشكل خاص تصنف تحت اسم خوارزميات الهجوم التحليلي للدليل, ولكن تم تطوير بعض الطرق مثل طريقة شانك وطريقة بولارد.

إن عدد الخطوات التي يجب اتباعها في هجوم ما يساوي تقريباً الجذر التربيعي لعدد عناصر الزمرة. فإذا كان عدد عناصر الزمرة  عنصر بالتالي بناءً على نظرية هاس, نحن نحتاج إلى عدد أولي طوله  بت, ومنه يحتاج المهاجم إلى  خطوة لإتمام الهجوم.

في الحقيقة ، المنحني الأكثر استخداماً طول عدده الأولي  بت, وهو يعطي درجة من الأمان تقدر بـ 128بت. ويوجد معايير لاختيار هذه الخوارزميات وضعها المعهد الوطني للمعايير و التكنولوجيا NISTحيث وضع العديد من المنحنيات المعيارية على هذا الأساس.

### ملاحظة على التعقيد

بما أن ضمن حسابات تبادل مفتاح ديفي-هيلمان في المنحنيات الناقصية توجد خوارزمية الجمع و التضعيف, التي هي مماثلة لخوارزمية الضرب و التربيع في RSA و تبادل مفتاح ديفي هيلمان العادي, فإن تعقيد هذه الخوارزمية يشابه تعقيد RSA و DHKE. لكن, تذكر أن معاملات ECDHKE أصغر من معاملاتRSA و DHKE. و هذا يلعب دوراً في نقليل تعقيد هذه الخوارزمية.

# الفصل الثالث نتاج المشروع

## الخوارزمية

سوف نبني خوارزميتنا على مسألة لوغاريتم متقطع. أساسها الرياضي هو التحاكي الذي نطبقه على نقطة ما في المستوي القطبي وفق نسبة تحاكي محددة وبزاوية محددة ثم سنطابق المستوي مع حقل غالوا للأعداد الأولية وسيظهر ذلك في تعريفنا للزاوية  ولنصف القطر . سندعو هذه الزمرة زمرة الدوائر القطبية, و فيما يلي سنورد شرحاً مفصلاً عنها.

### الزمرة الدائرية

نعرف المجموعة  على أنها مجموعة نقاط الدوائر المتمركزة في المبدأ  وأنصاف أقطارها هي مجموعة الأعداد الصحيحة الموجبة.

نعرف العملية الداخلية  كما يلي :

لتكن النقطتان  تنتميان للمجموعة . فيكون :



حيث يعطى  و  بالصيغ :





و بما أن الجمع داخلي على مجموعة الأعداد الصحيحة الموجبة, فأن  هي بنية.

سوف نثبت الآن أن  هي زمرة.

1. شرط التجميعية: كذلك يعد واضحاً لأن الجمع تجميعي على مجموعة الأعداد الصحيحة الموجبة.
2. شرط الحيادي: الحيادي هو النقطة .
3. شرط النظير : وفقاً لتعريف النظير الضربي بالمقاس  لدينا :



وبذلك نكون قد أثبتنا الشرط الأول للنظام التشفيري. سوف نثبت الآن أن  زمرة دائرية.

يمكننا أن نلاحظ بوضوح أن :





وبذلك نحصل على زمرة دائرية رتبتها هي  وهو ما يعد تفوقاً على ECC التي تبلغ رتبتها فيه حدها الأقصى عند  وفقاً لنظرية هاس .

### تبادل مفتاح ديفي-هيلمان في الدوائر القطبية

سنستخدم زمرتنا الدائرية من أجل إنجاز تبادل مفتاح ديفي-هيلمان. و سنقوم بذلك بطريقة مشابهة للتبادل العادي و التبادل بالمنحنيات الناقصية.

المرحلة التحضيرية :

1. اختر عدداً أولياً كبيراً .
2. اختر نقطة .
3. انشر .

نلاحظ في المرحلة التحضيرية أنه يوجد متحول وحيد فقط هو العدد الأولي , و هذا يعتبر أفضلية على ECC التي تقوم بنشر ثلاث متحولات  في الشبكة.

المرحلة الأساسية :

مصطفى محمود

**

* *

### برهان صحة تبادل مفتاح ديفي هيلمان في الدوائر القطبية

القيمة التي حسبها محمود هي :



و القيمة التي حسبها مصطفى هي :



و ببساطة, حسب خواص زمرة الدوائر القطبية :



و هو المطلوب.

### الحساب السريع

كما في RSA و ECC, سنستخدم خوارزمية خاصة من أجل حساب  بكفاءة عالية.

الخوارزمية 3.1.4 :

الدخل : النقطة . العدد.

الخرج : .

الخوارزمية :



## البرنامج

### شيفرة RSA

قمنا بكتابة برنامج بلغة JAVA يحاكي تنفيذ خوارزمية RSA. فيه ثلاث صفوف. اللأول عن الخوارزمية الإقليدية الممددة, و الثاني عن مولد الأرقام العشوائية و الأخير هو التنفيذ الفعلي للبرنامج.

صف الخوارزمية الإقليدية الممددة :

package rsa;

import java.math.BigInteger;

public class ExtendedEuclideanAlgorithm

{

BigInteger alpha, beta;

public void getAlphaBeta(BigInteger a, BigInteger b)

{

BigInteger d = a.gcd(b);

BigInteger x, y;

if(a == BigInteger.ZERO)

{

y = BigInteger.ONE;

this.beta = y;

x = BigInteger.ZERO;

this.alpha = x;

}

else if(b == BigInteger.ZERO)

{

y = BigInteger.ZERO;

this.beta = y;

x = BigInteger.ONE;

this.alpha = x;

}

else

{

BigInteger x1 = BigInteger.ZERO;

BigInteger x2 = BigInteger.ONE;

BigInteger y1 = BigInteger.ONE;

BigInteger y2 = BigInteger.ZERO;

while(b.compareTo(BigInteger.ZERO) != 0)

{

BigInteger q = a.divide(b);

BigInteger r = a.subtract(q.multiply(b));

x = x2.subtract(q.multiply(x1));

y = y2.subtract(q.multiply(y1));

a = b;

b = r;

x2 = x1;

x1 = x;

y2 = y1;

y1 = y;

}

d = a;

x = x2;

y = y2;

this.alpha = x;

this.beta = y;

}

}

}

صف مولد الأرقام العشوائية :

package rsa;

import java.util.Random;

import java.math.BigInteger;

import java.security.SecureRandom;

public class RNG

{

public static BigInteger generateBigPrime ()

{

SecureRandom sr = new SecureRandom();

Random primeSource = new Random(sr.nextLong());

return BigInteger.probablePrime(1024, primeSource);

}

public static BigInteger generateBigNumber ()

{

SecureRandom sr = new SecureRandom();

Random numberSource = new Random(sr.nextLong());

return new BigInteger(1025, numberSource);

}

}

صف البرنامج التنفيذي :

package rsa;

import rsa.RNG;

import rsa.ExtendedEuclideanAlgorithm;

import java.math.BigInteger;

import java.util.Random;

import java.util.Scanner;

public class RSA

{

public static void main(String[] args)

{

// defining vars...

BigInteger p = RNG.generateBigPrime();

BigInteger q = RNG.generateBigPrime();

// BigInteger p = new BigInteger("5");

// BigInteger q = new BigInteger("7");

BigInteger n = p.multiply(q);

BigInteger fn = (p.subtract(BigInteger.ONE)).multiply(q.subtract(BigInteger.ONE));

// outputting p, q, n and fn...

System.out.println("p = " + p + "\n" + "q = " + q + "\n" + "n = "

+ n + "\n" + "fn = " + fn);

// computing e...

Random eSource = new Random();

BigInteger e = new BigInteger(1024, eSource);

while(e.gcd(fn).compareTo(BigInteger.ONE) != 0 || e.compareTo(fn.subtract(BigInteger.ONE)) == 1)

{

e = new BigInteger(5, eSource);

}

// computing alpha and beta...

ExtendedEuclideanAlgorithm eeaImplementation = new ExtendedEuclideanAlgorithm();

eeaImplementation.getAlphaBeta(e, fn);

BigInteger alpha = eeaImplementation.alpha;

BigInteger beta = eeaImplementation.beta;

// outputting e, alpha and beta...

System.out.println("your random exponent is: " + "\n" + e + "\n"

+ "alpha = " + alpha + "\n" + "beta = " + beta);

// inputting integer message...

BigInteger message = new Scanner(System.in).nextBigInteger();

if(message.compareTo(n) == 1)

{

System.out.println("your message is too big");

}

BigInteger constMessage = message;

String eBin = e.toString(2);

for (int i = 1; i < eBin.length(); i++)

{

if(eBin.charAt(i) == '1')

{

message = message.pow(2).mod(n);

message = message.multiply(constMessage).mod(n);

}

if(eBin.charAt(i) == '0')

{

message = message.pow(2).mod(n);

}

}

// outputting encrypted message...

System.out.println("your encrypted message is: " + "\n" + message);

// decrypting message...

BigInteger ConstMessage = message;

if(alpha.compareTo(BigInteger.ZERO) == -1)

{

alpha = alpha.add(fn);

}

String alphaBin = alpha.toString(2);

for (int i = 1; i < alphaBin.length(); i++)

{

if(alphaBin.charAt(i) == '1')

{

message = message.pow(2).mod(n);

message = message.multiply(ConstMessage).mod(n);

}

if(alphaBin.charAt(i) == '0')

{

message = message.pow(2).mod(n);

}

}

// outputting decrypted message...

System.out.println("your decrypted message is: " + "\n" + message);

}

}

### شيفرة الدوائر القطبية

و كذلك قمنا بحاكاة تبادل مفتاح ديفي هيلمان في الدوائر القطبية. فيه ثلاث صفوف. الأول هو مولد الأرقام العشوائي, الثاني و صف النقطة و الأخير هو التنفيذ الفعلي.

صف مولد الأرقام العشوائي :

package cryptography.prototype;

import java.util.Random;

import java.math.BigInteger;

import java.security.SecureRandom;

public class RNG

{

public static BigInteger generateBigPrime ()

{

SecureRandom sr = new SecureRandom();

Random primeSource = new Random(sr.nextLong());

return BigInteger.probablePrime(1024, primeSource);

}

public static BigInteger generateBigNumber ()

{

SecureRandom sr = new SecureRandom();

Random numberSource = new Random(sr.nextLong());

return new BigInteger(1025, numberSource);

}

}

صف النقطة :

package cryptography.prototype;

import java.math.BigDecimal;

import java.math.BigInteger;

import java.math.MathContext;

public class Point

{

private BigDecimal radius;

private BigDecimal angle;

private static final BigDecimal bigPi = new BigDecimal("3.141592653589793");

private static final BigDecimal doubleBigPi = new BigDecimal("6.283185307179586476925286766559");

private final BigDecimal one = new BigDecimal("1");

private final BigDecimal two = new BigDecimal("2");

public Point(BigDecimal q)

{

this.radius = new BigDecimal("1");

this.angle = bigPi.divide(q, MathContext.DECIMAL128);

}

public BigDecimal getAngle() {

return angle;

}

public BigDecimal getRadius() {

return radius;

}

public void add()

{

this.angle = this.angle.add(one);

this.radius = this.radius.add(one);

while(this.angle.compareTo(doubleBigPi) == 1)

{

this.angle = this.angle.subtract(doubleBigPi);

}

}

public void addDouble()

{

this.angle = this.angle.add(one);

this.radius = this.radius.add(one);

this.angle = this.angle.multiply(two);

this.radius = this.radius.multiply(two);

while(this.angle.compareTo(doubleBigPi) == 1)

{

this.angle = this.angle.subtract(doubleBigPi);

}

}

public void pow(BigInteger power)

{

String pow = power.toString(2);

for (int i = 0; i < pow.length(); i++)

{

if(pow.charAt(i) == '1')

{

this.addDouble();

}

else if(pow.charAt(i) == '0')

{

this.add();

}

}

}

}

صف تنفيذ البرنامج :

package cryptography.prototype;

import cryptography.prototype.RNG;

import cryptography.prototype.Point;

import java.util.Scanner;

import java.util.Random;

import java.math.BigInteger;

import java.math.BigDecimal;

public class CryptographyPrototype

{

public static void main(String[] args)

{

BigDecimal q = new BigDecimal(RNG.generateBigPrime());

Point p = new Point(q);

BigInteger a = RNG.generateBigNumber();

p.pow(a);

BigInteger b = RNG.generateBigNumber();

p.pow(b);

System.out.println(p.getRadius() + "\n" + p.getAngle());

}

### الشيفرة المتناظرة Trivium

فيما يخص الشيفرات المتناظرة، قمنا بكتابة برنامج بلغة C++ يحاكي شيفرة Trivium. حيث قمنا بتعريف ثلاث مصفوفات ثنائية البعد، على أن كل مصفوفة هي عبارة عن Shift Register.  . ثم نقوم بإدخال قيمة لكل flip-flop. ثم تبدأ عملية توليد . أولا: يقوم كل flip-flop بتخزين قيمته بالذي يليه. ثم نقوم بحساب خرج كل flip-flop. حيث:







ثم نقوم بحساب القيمة الأولى من كل flip-flop :





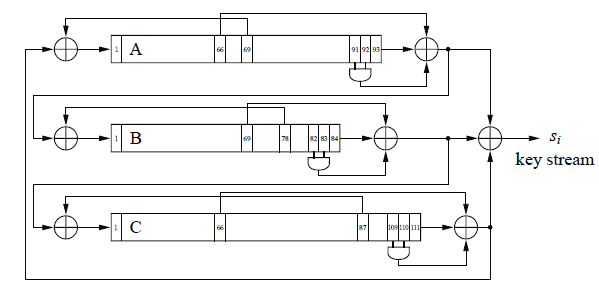


وتكرر تلك العمليات بعدد محارف النص المدخل. (فرضنا في البرنامج أن النص المدخل يحوي 1000 محرف على الأكثر).ثم نقوم بإدخال النص المراد تشفيره  ثم نقوم بتخزين ترميز كل محرف حسب جدول ASCII  ثم نقوم بتحويل  إلى النظام الثنائي .

ثم نقوم بحساب البتات المشفرة  حيث:



و بعد ذلك نقوم بتحويل البتات المشفرة إلى أعداد عشرية  ، و نقوم بإيجاد المحرف الموافق لكل عدد عشري مشفر.



#### الكود

#include<iostream>

#include<cmath>

#include<string>

#include<cstring>

bool a1[8000],b1[8000];

int a[1000],b[1000];

char x[1000];

using namespace std;

void bin(int a[]){

int p,e=0;

bool q[8];

for(int i=0;i<strlen(x);i++){

int t=a[i];

p=0;

while(1){

q[p]=t%2;

t/=2;

if(t==0)

break;

else

p++;

}

while(p<7){

q[p+1]=0;

p++;

}

for(int j=e;j<e+8;j++){

a1[j]=q[p];

p--;

}

e+=8;

}

}

void dec(bool b1[]){

int r=0,t=0;

for(int c=0;c<strlen(x)\*8;c+=8){

int sum=0;

int d=7;

for(int i=0;i<8;i++){

sum+=b1[r]\*pow(2,(double)d);

d--;

r++;

}

b[t]=sum;

t++;

}

}

void main(){

bool f1[2][93],f2[2][84],f3[2][111];

bool s1,s2,s3,s[1000];

cout<<"Enter your 288-bit key" << endl;

for(int i=0;i<93;i++)

cin>>f1[0][i];

for(int i=0;i<84;i++)

cin>>f2[0][i];

for(int i=0;i<111;i++)

cin>>f3[0][i];

for(int j=0;j<1000;j++){

for(int i=0;i<92;i++)

f1[1][i+1]=f1[0][i];

for(int i=0;i<83;i++)

f2[1][i+1]=f2[0][i];

for(int i=0;i<110;i++)

f3[1][i+1]=f3[0][i];

s1=(f1[1][91]\*f1[1][90]+f1[1][92]+f1[1][65])%2;

s2=(f2[1][83]+f2[1][82]\*f2[1][81]+f2[1][68])%2;

s3=(f3[1][110]+f3[1][109]\*f3[1][108]+f3[1][65])%2;

f1[1][0]=(f1[1][68]+s3)%2;

f2[1][0]=(f2[1][77]+s1)%2;

f3[1][0]=(f3[1][86]+s2)%2;

s[j]=(s1+s2+s3)%2;

for(int i=0;i<93;i++)

f1[0][i]=f1[1][i];

for(int i=0;i<84;i++)

f2[0][i]=f2[1][i];

for(int i=0;i<111;i++)

f3[0][i]=f3[1][i];

}

cout <<"Enter your text : " << endl;

cin>>x;

for(int i=0;i<strlen(x);i++){

a[i]=x[i];

if(a[i]<0) a[i]+=256;

}

bin(a);

for(int i=0;i<strlen(x)\*8;i++)

b1[i]=(s[i]+a1[i])%2;

dec(b1);

cout << "Your result is : " << endl;

for(int i=0;i<strlen(x);i++){

if(b[i]<0) b[i]+=256;

cout<<(char)b[i];

}

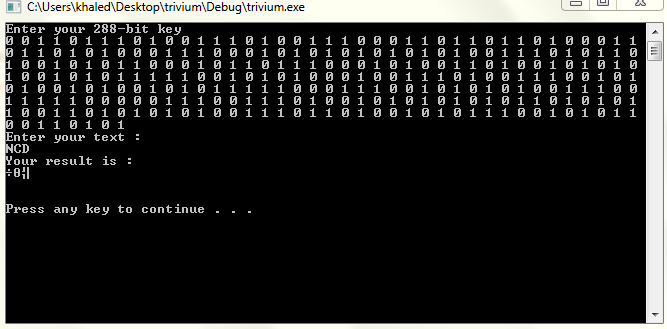
cout<<endl<<endl<<endl;

system("pause");

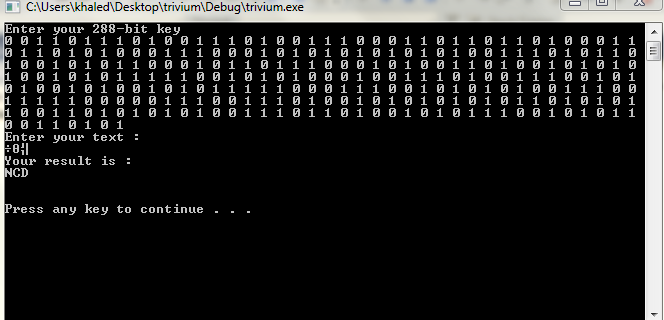
}

#### تنفيذ البرنامج

هذه هي عملية تشفير النص “NCD”.



أما عملية فك تشفير النص المشفر, فهي مماثلة لتنفيذ عملية التشفير.



# الخاتمة

في الختام, ننوه إلى أهمية دراسة علم التشفير لما له من دور كبير في حفظ المعلومات والأسرار المتعلقة بجميع الجوانب العلمية والسياسية والعسكرية, كما يجب الإشارة الى جميع المواضيع التي تم ادراجها في قسم (الواجهة الرياضية للمشروع) في المناهج التعليمية من أجل فهم أكبر للتشفير غير المتناظر.

قمنا في مشروع بحثنا هذا بالتوصل إلى زمرة دائرية جديدة لم تكن مكتشفة من قبل. و تتمتع هذه الزمرة ببعض الخاص التي ترفع من كفاءتها مقارنة بالزمر الأخرى. كما كتبتنا برنامجاً ليحاكي هذه الشيفرة, لنبرهن أن علم التشفير ليس ببعيد المنال عن أمتنا. و نأم أن يكون هذا المشروع كحجر الأساس لبداية ازدهار علم التشفير في سوريا.

نورد **النتائج** التالية :

1. يمكن الحصول على زمرة دائرية من حقل تبديلي.
2. لا يؤثر تعقيد العمليات على قوة النظام التشفيري.
3. الحجم الأفضل للمفتاح في جميع الخوارزميات غير المتناظرة هو 256بت.
4. استخدام خوارزميات جديدة لاختبار أولية عدد (وهي لا تعني الحكم على عدد أنه أولي تماماً وإنما "شبه" أولي) وهو ما توفره معايير ميلر-رابين ومعيار فيرما.

وبناءً على هذه النتائج نوصي **بالمقترحات** التالية:

1. زيادة الاهتمام بعلم التشفير في الجامعات السورية.
2. دراسة خوارزمية ElGamal اللامتناظرة, و هي من اختراع عالم التشفير المصري الأصل, طاهر الجمال.
3. دراسة إمكانية بناء نظم تشفيرية على توابع جهة واحدة أخرى. مثل مسألة كثيرات الحدود و خوارزميات كشف الأخطاء. فقد نجد في تلك التوابع إمكانيات لبناء نظام تشفيري متفوق على الأنظمة الحالية.
4. الاطلاع على الدراسات الجديدة حول الخوارزميات RSAوECC ودراسة نقاط قوتها وضعفها.

وفي الختام نقدم الشكر لإدارة المركز الوطني للمتميزين وكادره التدريسي ولمشرف المشروع الأستاذ حبيب عيسى، وإلى

**قائد الوطن السيد الرئيس بشار الأسد والسيدة عقيلته**

واللذان قدما لنا الدعم الأكبر متمثلاً بالمركز الوطني للمتميزين وعسى أن نكون قد قدمنا ما نستطيع في هذا المجال لتصبح سوريا من الدول المتقدمة في هذا المجال.

أعضاء المشروع :

جابر أديب ـ مصطفى خليل ـ محمود قره فلاح ـ خالد إسماعيل

|  |  |  |
| --- | --- | --- |
|  | English | عربي |
| CRT | Chinese Remainders Theorem | مبرهنة البواقي الصينية |
| ECC | Elliptic Curves Cryptography | التشفير بالمنحنيات الناقصية |
| ECDHKE | Elliptic Curves Diffie-Hellman Key Exchange | تبادل مفتاح ديفي-هيلمان بالمنحنيات الناقصية |
| DHKE | Diffie-Hellman Key Exchange | تبادل مفتاح |
| RNG | Random Number Generator | مولد أرقام عشوائية |
| RSA | Rivest Shamir Adleman | أسماء العلماء رايفت شامير و أديلمان |

# المصطلحات

# المراجع :

1. A. Menezes, P. VanOorschot, S. Vanstone. *Handbook of Applied Cryptography*. CRC press, Boca Raton, Florida, USA, 1997.
2. Christof Paar, Jan Penzel. *Understanding Cryptography*. Springer, Heidelberg, Berlin, Germany, 2010.
3. Titu Andrescu, Dorin Andrica. *Number Theory structures, examples and problems.* Springer, London, United Kingdom, 2010.
4. W. Diffie, M.Hellman. *New directions in cryptography*. IEEE Transactions on Information Theory, IT-22:644-644, 1976.
5. T.ElGamal. *Apublic-key cryptosystem and a digital signature scheme based on discrete logarithms*, IEEE Transactions on Information Theory, IT-31(4):469-472, 1985.
6. Dan Boneh, Ron Rivest, Adi Shamir, and Len Adelman. *Twenty Years of Attacks on the RSA Cryptosystem*. Notices of the AMS, 46:203-213, 1999.

1. Titu Andrescu, Dorin Andrica. *Number Theory structures, examples and problems.* Springer, London, United Kingdom, 2010. [↑](#footnote-ref-1)
2. Titu Andrescu, Dorin Andrica. *Number Theory structures, examples and problems.* Springer, London, United Kingdom, 2010. Page 47. [↑](#footnote-ref-2)
3. Christof Paar, Jan Penzel. *Understanding Cryptography*. Springer, Heidelberg, Berlin, Germany, 2010. [↑](#footnote-ref-3)
4. A. Menezes, P. VanOorschot, S. Vanstone. *Handbook of Applied Cryptography*. CRC press, Boca Raton, Florida, USA, 1997. Page 67. [↑](#footnote-ref-4)
5. Christof Paar, Jan Penzel. *Understanding Cryptography*. Springer, Heidelberg, Berlin, Germany, 2010. Page 182. [↑](#footnote-ref-5)
6. هنا نقوم بحساب الأس الخاص أولاً, ثم نحسب نظيره الضربي بالمقاس  و ليس بالعكس. [↑](#footnote-ref-6)