

تحقيق وثوقية الرسالة وسريتها باستخدام خوارزمية الخصوصية المتفوقة PGP

ربا جاسم عيسى

جامعة الموصل - كلية علوم الحاسوب والرياضيات

الخلاصة

يهدف البحث إلى التحقق من وثوقية البيانات والمعلومات المتبادلة والتطبيق باستخدام خوارزمية PGP ضمن شبكة الحاسبات، وفيه يتم تقديم طريقة مقترحة لتحقيق الوثوقية باستخدام التواقيع الرقمية واستخدام خوارزمية مناسبة لكبس البيانات قبل إرسالها ومن خلال أمنية المعلومات وعلم التشفير تفترض الدراسة إمكانية تحقيق وثوقية وسرية المعلومات المتداولة إلكترونياً. وبالاستعانة بخوارزميات الكبس المختلفة والتواقيع الرقمية وإضفاء أسلوب للتشفير يجري تعزيز الوثوقية بخوارزمية الخصوصية المتفوقة والتحقق من سريتها تطبيقياً. وقد توصل البحث لعدة نتائج لعل أبرزها التقليل من المعلومات الإحصائية المقيدة لمحلل الشفرة وتقليل حجم النص لتسريع عملية التشفير فضلاً عن الحصول على مستوى أمنية أعلى بالتوليف التطبيقي للبرامج الجاهزة مع اللغات البرمجية المعتمدة في هذا المجال

كلمات مفتاحية : وثوقية ، سرية ، خوارزمية الخصوصية المتفوقة ، PGP

مقدمة

المعلومات (information theory)، أمنية الحاسوب (computer security) وكذلك الهندسة [5].

التغيير الرئيسي الذي أثر على أمنية المعلومات هو استخدام الأنظمة الموزعة Distributed Systems واستخدام تسهيلات الشبكات الاتصالات لنقل البيانات بين مستخدمي الطرفية User Terminal والحاسوب وبين الحاسوب والحاسوب الأخر. أصبحت إجراءات أمنية الشبكة مطلوبة لحماية البيانات خلال ترانسلمها.

هناك طريقتان رئيسيتان لتشفير المعلومات وهما: طريقة التشفير المتناظرة Symmetric Key Encryption وطريقة التشفير غير المتناظرة Asymmetric (SKE) وطريقة التشفير غير المتناظرة (Key Encryption (AKE) [3] [8] [4] [6].

الحاجة الى علم التشفير

تحتاج الأمنية إلى بقاء البيانات آمنة من الوصول غير المخول (unauthorized access)، ويمكن تقسيم المشكلة الأمنية إلى (5) متطلبات ومن الضروري الإشارة إليها وتحديدها بدقة وهذه المتطلبات هي:

الوثوقية (Confidentiality) والتحويل (Authorization) (Authentication) الصلاحية (Authorization)

نتيجة لتطور التقنيات الحديثة في الوقت الحاضر دخل الحاسوب في كل مجالات المجتمع ومنها مجال إرسال المعلومات التي تزداد بشكل هائل، فالمعلومات ترسل وتعالج أوتوماتيكياً على نطاق واسع، وهذا يتطلب الحرس على الخزن السري لنقل المعلومات. إذ يتم استخدام خوارزمية خاصة لتشفير البيانات، لزيادة حماية الرسائل وتقليل المساحة اللازمة ل تخزينها في ملف أو إرسالها على الشبكة استخدمت تقنية الكبس.

أن كبس النصوص يكون باستخدام رموز قصيرة للنصوص المتشابهة وطويلة للأخرى، وهي عملية تحويل بيانات نص إلى بيانات أخرى بحجم أقل من البيانات الأولى. وفي هذا البحث سوف نتناول الطريقتين السابقتين (الكبس، التشفير) إذ تم استخدام خوارزمية الخصوصية المتفوقة (PGP) للتشفير التي تستخدم خوارزمية الكبس (Lempel-Ziv (LZ77) 1977.

علم التشفير

يمكن تعريف علم التشفير على انه ممارسة ودراسة إخفاء المعلومات. في العصر الحديث يعتبر التشفير احد فروع كلا من علم الرياضيات وعلم الحاسوب وله علاقة وثيقة بنظرية

1- محلل الشفرة يعتمد على المعلومات الزائدة (Redundancy) في النص الصريح، ويكسب الملف قبل التشفير يقلل هذه الكثرة.

2- يحتاج التشفير إلى وقت كبير عادة، ويكسب الملف قبل التشفير سوف يسرع عملية التشفير.

3- ويفترض أن تتم عملية الكسب قبل التشفير، وإذا كانت خوارزمية التشفير جيدة فإن النص المشفر سوف يكون غير قابل للكسب وسوف يظهر كأنه بيانات عشوائية [9].

خوارزمية الخصوصية المتفوقة Pretty Good Privacy PGP

4- إن خوارزمية الخصوصية المتفوقة التي استخدمت لتحقيق طور السرية والوثوقية باستخدامها لخوارزمية RSA والتي تعد من أفضل خوارزميات تشفير المفتاح العام وقد استخدمها الباحثون الثلاثة R-S-A لتشفير الرسالة للتخلص من مشكلة توزيع المفتاح [2] (Key Distribution). كما أنها تسمح بالتشفير/ فك التشفير والتوقيع الالكتروني مما يجعل البريد الالكتروني يحضى بكثير من الأهمية [12].

تحقيق الوثوقية والسرية في الخصوصية المتفوقة:

5- حيث يدمج الطورين على نفس الرسالة. في البداية يتم توليد التوقيع باستخدام المفتاح الخاص للمرسل للنص الصريح ودمج مع الرسالة. ثم النص الصريح مع الرسالة يشفران باستخدام خوارزمية RSA. ثم يكسب النص باستخدام خوارزمية LZ77.

6- ان خوارزمية الخصوصية المتفوقة (PGP) تستخدم الكسب قبل التشفير، وهذا له فائدة في حفظ مساحة خزن الملفات والاتصال عبر الشبكة. يشار إلى الكسب بZ و إلى فك الكسب Z^{-1} تشفير الرسالة يكون بعد تطبيق الكسب لتعزيز سرية التشفير ولأن الرسالة المكبوسة تكون اقل طولاً من النص الصريح، وبالتالي فإن تحليل الشفرة يكون أصعب. والشكل (1) يوضح وظائف ال PGP في طور الوثوقية.

الجانب العملي:

7- بعد التعرف على الجانب النظري سنتناول في هذا البحث الجانب العملي والتطبيقي للخوارزميات التي تم استخدامها وهي خوارزمية RSA و خوارزمية LZ77 والشكل (2) يوضح المخطط العام للنظام.

إضافة التوقيع الرقمي والتشفير باستخدام خوارزمية RSA

8- استخدمت هذه الخوارزمية لإضافة توقيع باستخدام المفتاح الخاص للمرسل (d)، وإتباع الخوارزمية الاتية لحساب $(x^d \text{ mod } n)$. بعد إضافة التوقيع الرقمي يتم تشفير الرسالة

وتكامل البيانات (Data Integrity) عدم الإنكار (Non-Repudiation) [3].

كسب البيانات Data Compression

والمقصود بكسب البيانات: هو ترميز بيانات ملف معين لإيجاد مساحة كفاءة بديلة لذلك الملف. ومع تطور شبكات الحاسوب نشأ إرسال جديد بكسب البيانات يزيد من كمية البيانات المرسل على الشبكة بتقليل عدد الأرقام الثنائية قل الإرسال. [13]، [2].

كسب البيانات باستخدام ZIP

تستخدم خوارزمية الخصوصية المتفوقة الكسب باستخدام Zip. خوارزمية ZIP و أنواعها من الخوارزميات الأخرى و تعتمد على تقنية النافذة المؤقتة (Sliding Window Buffer) التي تحتوي على النص المعالج حديثاً. و يشار إلى هذه الخوارزمية ب-LZ77.

تستغل LZ77 و أنواعها الكلمات والمقاطع المكررة ضمن النص. وعند حدوث تكرار فإن المقطع المكرر يبدل برموز قصيرة. تعمل هذه الخوارزمية على مسح مثل هذه التكرارات و إبدال النص المتكرر برموز تشير إلى مواقع التكرار و عدد الأحرف المكررة. عملية فك الكسب تكون من خلال هذه الرموز و استنتاج النص الأصلي [13],[11],[1].

خوارزمية الكسب (compression Algorithm)

في خوارزمية LZ77 و أنواعها تستخدم مخزنان (Buffers) هما المخزن المؤقت Sliding Window (Buffer) (و يحتوي على أخر مجموعة من الأحرف في النص الأصلي التي سوف تعالج، ومخزن التقدم (Look Ahead Buffer) الذي يحتوي على عدد من الأحرف التي تليها و التي سوف تعالج. [7] الخوارزمية تحاول أن تجد تطابقاً لحرفين أو أكثر من بداية مخزن التقدم مع سلسلة في المخزن المؤقت. وحالة عدم وجود تطابق، فإن الحرف الأول في مخزن التقدم يخرج و يزحف إلى المخزن المؤقت و الرفع الأول في المخزن المؤقت يخرج من المخزن. أما إذا وجد تطابق فإنها تكمل إلى اعلى حد للتطابق ثم تمثل بشكل ثلاثي.

(دليل Indicator, مؤشر Pointer, الطول Length) من سلسلة الأحرف، فأن K من الأحرف في النافذة المؤقتة تخرج، و K من الأحرف التي سوف ترمز تدخل إلى النافذة. خوارزمية LZ77 فعالة و متكيفة مع الإدخال الحالي [1] [10].

الكسب و التشفير Compression And Encryption

تستخدم خوارزميات الكسب مع خوارزميات التشفير للأسباب التالية:

للتكرار هو قبله ب 21 حرفاً و طول الأحرف المتكررة هو 11. الجزء المتبقي من الرسالة المكبوسة هو y, أما الأحرف الخمسة التي تليها فسوف تبدل بمؤشر 22 وطول 4 وهو الفراغ و المقطع jump وتكمل باقي الرسالة. الناتج النهائي للرسالة المكبوسة يكون:

1 the brown fox jumped over the brown foxy jumping frog
21 22 4

وسيكون ناتج الكيس مخزون في ملف كما موضح في الشكل (4) وحسب الخوارزمية التالية:

خوارزمية الكيس باستخدام LZ77

- 1-البداية
- 2-قراءة النص المدخل في ملف من نوع .TXT.
- 3- تقطيع النص المدخل إلى حروف وخزنه في مصفوفة جديدة STR.
- 4- تعريف مصفوفة لخزن موقع التكرار السابق offset(0) = 0 (لان اول حرف لا يتكرر) ، ، offset(1) = 0 (لان LZ77 لا تشفر حرف وإنما حرفين فأكثر)
- 5- تعريف مصفوفة لخزن عدد الحروف المكررة 0 = offset(0) Length (1)= 0 ، Length
- 6- تعريف مصفوفة لخزن الحروف Char(0)=0 ، Char(1)=Str(1)
- 7- تعريف مؤشر (id) يشير الى الموقع الثاني في النص الأصلي
- 8- كرر الخطوات من 9- 16 حتى نهاية النص
- 9- قراءة النص ابتداء من الموقع الثاني،
- 10- تعريف متغير 0 = Index يشير الى الموقع الثاني في النص المراد كيبسه، والمتغير N=0 ليمثل عدد تكرار الكلمات
- 11- كرر الخطوة 12 طالما $i-1 < j$
- 12- اذا كانت Str(j)=Str(i) و index=0 فان index=i-1 (j لإيجاد offset)، $j=j+1$
- 13- $N=N+1$
- 14- اذا كانت $N \leq 2$ اذهب إلى الخطوة 16
- 15- $offset(id)=index$ و $length(id)=N$ و $id=id+1$ و $Char(id)=Str(i)$
- 16- $i=i-1$ و $offset(id)=0$ و $length(id)=0$ و $Char(id)=Str(i)$ و $id=id+1$ يعني لا يوجد تكرار للحرف.
- 17- $i=i+1$
- 18- كرر الخوة 19 طالما $i-1$
- 19- اطبع offset(i) و length(i) و Char(i)
- 20- النهاية

باستخدام المفتاح العام للمستلم ويمثل (e) وهو عدد أولي (prime number) والمفتاح الخاص (d) لفك الشفرة. 9- أساس عمل هذه الخوارزمية أنها تأخذ القيمة وترفعها للقوة (power of)، بالمفتاح العام أو الخاص ولتسريع التنفيذ تم إتباع الخوارزمية الآتية لحساب $(x^e \text{ mod } n)$ حيث تمثل x العدد نفسه أما e_i (لكل $i=0, \dots, m$) فهي سلسلة الأرقام الثنائية للعدد e كما موضح بالشكل (3).

الكيس باستخدام خوارزمية LZ77

10- في البداية نضع في المخزن المؤقت عدداً من الأحرف بعدد K الذي يمثل طول المخزن، وفي مخزن التقدم عدد الأحرف يكون بطول L اختيار K, L يكون غير محدد حسب حجم النص، بعد ذلك نبدأ بالمقارنة بين المخزين و عند حدوث تشابه نحسب موقع أول حرف من سلسلة الأحرف المتشابهة، ونستمر إلى أن نصل إلى اختلاف في الأحرف بين المخزين. نخزن عدد الأحرف المتشابهة في المخزين ثم نخزن مؤشر (Indicator) الذي يشير إلى وجود كيس ثم نخزن الموقع وطول التشابه، وهكذا تعاد العملية عند كل تشابه إلى نهاية مخزن التقدم. في كل مقارنة مع مخزن التقدم نقارن مع نهاية المخزن (وصل إلى L) إذا كانت نهاية مخزن التقدم سوف تزحف الأحرف بمقدار واحد و آخر حرف في مخزن التقدم يكون أول حرف من النص المتبقي الذي لم يحدث له معالجة. وهكذا نستمر إلى نهاية النص. وقد تم حساب نسبة الكيس وهي نسبة حجم الملف قبل الكيس على حجم الملف بعد الكيس كما في المعادلتين (1) و (2): حجم الملف قبل الكيس:

$$\text{حجم الملف بعد الكيس} = \text{حجم الملف قبل الكيس} - \text{حجم الملف بعد الكيس} \quad (1)$$

$$\text{نسبة الكيس} = \frac{\text{حجم الملف بعد الكيس}}{\text{حجم الملف قبل الكيس}} \quad (2)$$

ولتوضيح عمل الخوارزمية نفرض المثال الآتي [2]:

The brown fox jumped over the brown foxy jumping frog

يتكون هذا النص من 44 حرفاً = 352 رقماً ثنائياً (44*8=352 رقماً ثنائياً). تعالج الخوارزمية هذا النص من اليسار إلى اليمين. قبل البدء بالمعالجة يجب البحث عن الأحرف المتتالية المتكررة. وفي حالة وجود تكرار نستمر إلى نهاية التكرار المتسلسل، بمعنى آخر، في كل مرة يحدث تكرار نحاول أن نأخذ أكبر عدد ممكن من الأحرف المتسلسلة. في هذا المثال فان أول تكرار هو the brown fox وهذه الأحرف تبدل بمؤشر يشير إلى التسلسل السابق للأحرف المتكررة و طول التكرار. و في هذه الحالة فان التسلسل السابق

تم التعامل في هذا البحث مع نص صريح وإجراء عمليات معينة للحفاظ على السرية والوثوقية وضمان صعوبة دخول غير المخول لمعرفته ومن خلال ذلك تم التوصل إلى الاستنتاجات التالية:

1. عند إدخال النص إلى خوارزمية الكبس LZ77 وهي من الخوارزميات المتكيفة (Adaptive Algorithms) والتي تعمل على كبس سلاسل من الأحرف المتمثلة بشكل كلمات أو عبارات وهذا يفيد في التقليل من المعلومات الإحصائية التي تفيد محلل الشفرة وتمكنه من كسر الشفرة إضافة إلى ذلك فإن الكبس يقلل من حجم النص ليسرع عملية التشفير التي تلي الكبس.
2. وعند تحليل نتائج التطبيق اتضحت ضرورة الأخذ بنظر الاعتبار عدم وجود الفراغ، أي إدخال الكلمات المراد تشفيرها وكبسها بدون فراغات.
3. باستخدام خوارزمية الخصوصية المتوقعة تم الحصول على مستوى أمنية أعلى.
4. تم تطبيق خوارزمية مقترحة لتحقيق التوقيع الرقمي والحصول على درجة عالية من العشوائية للنص الناتج وبالتالي أعطت هذه الطريقة مستوى عالياً من السرية و الوثوقية.
5. ان استخدام لغة ++ C أعطى البرنامج صورة متماسكة وسهلة التداول مقارنة بجمل الانتقال بين السطور المستخدمة في لغتي بيبيك و فورتران.

المصادر

1. إسماعيل، ياسين حكمت (2004). دراسة الوثوقية وتحققها باستخدام التوقيع الرقمية. 34. رسالة ماجستير، كلية علوم الحاسبات والرياضيات، جامعة الموصل.
2. الحسو، شهد عبد الرحمن (2004). تصميم نظام هجين بالاعتماد على النصوص. 54. رسالة ماجستير، كلية علوم الحاسبات والرياضيات، جامعة الموصل.
3. الحمادي، علاء حسين ومحمد علاء الحمادي (2008). إخفاء المعلومات. 18. إثراء للنشر والتوزيع، عمان.
4. الشاهين، ناهي يوسف ونديم شاهين (2004). دراسة وتصميم خوارزمية تشفير البيانات المنقولة عبر الشبكة. مجلة جامعة دمشق للعلوم الهندسية، المجلد 20: العدد الثاني. 12-22.
5. حسين، عبد الأمير خلف (2010). طرق التشفير للمبتدئين. 33. دار وائل للنشر والتوزيع، عمان.
6. Chandra, P. (2005). *Bullet Proof Wireless Security*. 32 Elsevier Inc.
7. <http://encyclopedia2.thefreedictionary.com/PGP>.
8. Lucas, M. W. (2006). *PGP and GPG: Email for the Practical Paranoid*. 55 Starch Press, July 10,.

فك الكبس (Decompression) لخوارزمية LZ77

عند فك الكبس فإننا نبحث عن المؤشر (indicator) الذي يشير إلى وجود كبس ثم نخزن الرقمين اللذين يليه على إنهما يمثلان موقع أول الحروف في السلسلة المتشابهة وطول التشابه .

خوارزمية فك النص المكبوس بطريقة LZ77

1. اقرأ النص المكبوس Doc.txt
2. كرر الخطوات 3-15 حتى نهاية الملف
3. اطبع (i) و Length(i) و Char(i)
4. $i=i+1$
5. نعرف مصفوفة $Str = 0$ لخزن النص بعد فك الكبس
6. $J=0$
7. كرر الخطوات 8-13 طالما $(j < i-1)$
8. إذا كان $offset(j)=0$ (أي لا يوجد كبس) ، اذهب إلى الخطوة 10
9. نعرف المتغير K ، $K=j-offset(j)$ (لنحدد بداية الكلمة المكبوسة) ، اذهب إلى الخطوة 11
10. $Str=Chr(j)$
11. نعرف متغير K ، $K1=K$
12. إذا كان $K1 < length(j)$ فان $Str=Char(K1)$
13. $K1=K1+1$
14. $j=j+1$
15. $Str=Str&Char(i-1)$
16. النهاية

واجهة النظام:

في هذه الواجهة يتم إدخال النص الصريح الذي نريد تشفيره أو النص المشفر الذي نريد فك شفرته. الناتج سوف يخزن داخل ملف.

Encryption: عند الضغط على هذا الاختيار يتم

إجراء عملية تشفير للنص المدخل.

Decryption: عند الضغط على هذا الاختيار يتم

إجراء عملية فك تشفير للنص المدخل.

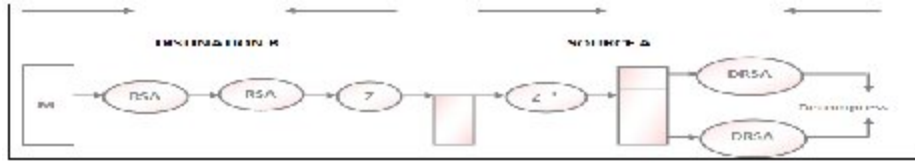
فعلى سبيل المثال لو أدخلنا النص السابق: The brown fox jumped over the brown foxy jumping frog مخزونة في ملف كما في الشكل (6). وعند استلام هذا الملف من قبل الشخص المخول سيقوم بفك الكبس والتشفير في الشكل (7). يلاحظ انه قد تم استرجاع النص الأصلي.

<pre> c2 = 1 For j = 1 To Val(Text3.Text) c2 = (c2 * c1) Mod (Val(Text4.Text) * Val(Text5.Text)) Next j str1 = str1 & Chr(c2 + 96) Next i For i = 0 To Len(str1) str(i) = Mid(str1, i + 1, 1) Next i off(0) = 0, off(1) = 0, lenth(0) = 0 lenth(1) = 0 char(0) = str(0) char(1) = str(1) id = 2 For i = 2 To Len(str1) - 1 Index = 0 n = 0 For j = 0 To i - 1 If (str(i) = str(j)) Then If (Index = 0) Then Index = i - j End If n = n + 1 i = i + 1 End If Next j If (n > 2) Then off(id) = Index lenth(id) = n char(id) = str(i) id = id + 1 Else i = i - n off(id) = 0 </pre>	<pre> str = str & char(j) Else k = j - off(j) For k1 = k To lenth(j) - 1 str = str & char(k1) Next k1 End If Next j str = str & char(i - 1) For i = 1 To Len(str) Number = Mid(str, i, 1) p1 = 1 For j = 1 To Val(Text9.Text) p1 = (p1 * (Asc(Number) - 96)) Mod (Val(Text6.Text) * Val(Text7.Text)) Next j Text10.Text = Val(Text10.Text) Mod (Val(Text6.Text) * Val(Text7.Text)) c2 = 1 For j = 1 To Val(Text8.Text) c2 = c2 * p1 Next j c2 = c2 Mod (Val(Text6.Text) * Val(Text7.Text)) es = Chr(c2 + 96) Text1.Text = Text1.Text & es Next i End Sub Private Sub Form_Load() Load Form3 Form3.Show End Sub </pre>
---	--

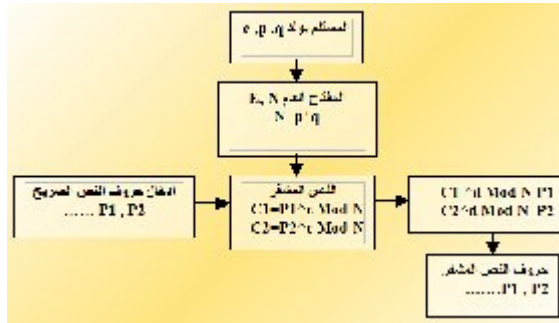
9. Salomon, D. (1998). *Data Compression: the Complete Reference*.43.Springer-Verlag, New York.
10. Sayood, Kh. (2006). *Introduction to Data Compression*. 48. 3rd ed., Elsevier Inc.
11. Schneier, B. (1996). *Applied Cryptography*. 23.2nd ed., John Wiley & Sons Inc.
12. Stalling, W. (1999). *Cryptography and Network Security: Principles and Practice*,66. 2nd ed., Prentice –Hall, Inc.
13. Umbaugh, S. E. (1998). *Computer Vision and Image*.22. Prentice –Hall, Inc,1998.
14. *Processing: A Practical Approach Using CVIP Tools*".32Prentice –Hall, Inc,1998.

ملحق-1: برمجة البحث

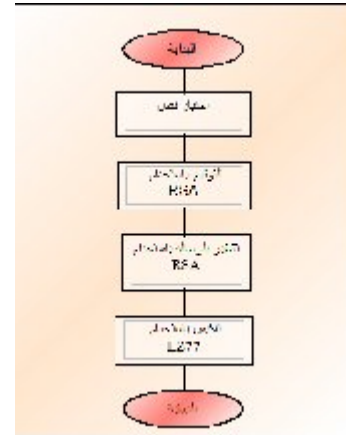
<pre> Private Sub Command1_Click() End End Sub Private Sub Command2_Click() Dim str(100) As String Dim off(100), lenth(100) As Integer Dim char(100) As String Dim p1, c1 As Integer Dim str1 As String str1 = "" Open "encrypt.txt" For Output As #1 chars_in_file% = Len(Text1.Text) For i = 1 To chars_in_file% letters\$ = Mid(Text1.Text, i, 1) p1 = Asc(letters\$) - 96 p2 = 1 For j = 1 To Val(Text2.Text) p2 = p2 * p1 Next j c1 = p2 Mod (Val(Text4.Text) * Val(Text5.Text)) </pre>	<pre> lenth(id) = 0 char(id) = str(i) id = id + 1 End If Next i For i = 0 To id - 1 Print #1, off(i) & ", " & lenth(i) & ", " & char(i) Next i End Sub Private Sub Command3_Click() Dim str As String Dim off(100), lenth(100) As Integer Dim char(100) As String Dim p1, c1 As Integer i = 0 Open "encrypt.txt" For Input As #1 Do Until EOF(1) Input #1, off(i) Input #1, lenth(i) Input #1, char(i) i = i + 1 Loop str = "" For j = 0 To i - 1 If (off(j) = 0) Then </pre>
--	---



الشكل (1): وظائف الـ PGP في طور الوثوقية



الشكل (3): إضافة التوقيع والتشفير باستخدام خوارزمية RSA



الشكل (2): المخطط العام للنظام



Other	Length	Character
0	1	A
0	2	B
0	3	C
0	4	D
0	5	E
0	6	F
0	7	G
0	8	H
0	9	I
0	0	J
0	1	K
0	2	L
0	3	M
0	4	N
0	5	O
0	6	P
0	7	Q
0	8	R
0	9	S
0	0	T
0	1	U
0	2	V
0	3	W
0	4	X
0	5	Y
0	6	Z
0	7	[
0	8]
0	9	^
0	0	_
0	1	`
0	2	{
0	3	}
0	4	~
0	5	
0	6	
0	7	
0	8	
0	9	
0	0	
0	1	
0	2	
0	3	
0	4	
0	5	
0	6	
0	7	
0	8	
0	9	
0	0	

الشكل (4): ملف ناتج عملية الكبس



الشكل (6): نتيجة التشفير والكبس

الشكل (7): فك الكبس والتشفير

Achieve reliable and confidential message using Pretty Good Privacy (PGP) algorithm

Raya Jassim Essa

E.mail: mortadha61@yahoo.com

Abstract:The paper aims at verifying the reliability of the exchanged data and information. The study conducted using the Pretty Good Privacy (PGP) algorithm application within a network of computers. It proposed to provide a method to achieve reliability, using digital signatures and a suitable algorithm for pressing the data before sending it. By information security and cryptography, the study assumes the possibility of achieving reliable and confidential information handled electronically. Algorithms using different pressing and digital signatures as well as giving a method for encryption. Enhance reliability algorithm being superior privacy and verification of confidentiality applied. The research has come to several conclusions, most notably: reduction of statistical information limiting the code analyst, reduce the size of the text to speed up the encryption process, access to the highest level of security; and synthesis applied to programs with ready-made structural language adopted in this area.