P- ISSN 1991-8941 E-ISSN 2706-6703 (JUAPS) مجلة جامعة الانبار للعلوم الصرفة Open Access
2013,(7), (1) :33-39

تحقيق وثوقية الرسالة وسريتها باستخدام خوارزمية الخصوصية المتفوقة PGP

ريا جاسم عيسى

جامعة الموصل - كلية علوم الحاسوب والرياضيات

معلومات البحث:
تاريخ التسليم: ٠٠/٠٠/٠٠
تاريخ القبول: ٠٠/٠٠/٠٠
تاريخ النشر: ٩ / ١٢ / ٢٠١٢
DOI: 10.37652/juaps.2012.63375

الكلمات المفتاحية:
وثوقية ،
سرية ،
خوارزمية الخصوصية المتفوقة ،
PGP.

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 الكترونيا. وبالاستعانة بخوارزميات الكبس المختلفة والتواقيع الرقمية وإضفاء أسلوب للتشفير يجري تعزيز الوثوقية بخوارزمية الخصوصية المتفوقة والتحقق من سريتها تطبيقيا. وقد توصل البحث لعدة نتائج لعل أبرزها التقليل من المعلومات الإحصائية المقيدة لمحلل الشفرة وتقليل حجم النص لتسريع عملية التشفير فضلا عن الحصول على مستوى أمنية أعلى بالتوليف التطبيقي للبرامج الجاهزة مع اللغات البنيوية المعتمدة في هذا المجال.

مقدمة

نتيجة لتطور التقنيات الحديثة في الوقت الحاضر دخل الحاسوب في كل مجالات المجتمع ومنها مجال إرسال المعلومات التي تزداد بشكل هائل، فالمعلومات ترسل وتعالج أوتوماتيكيا على نطاق واسع، وهذا يتطلَّب الحرص على الخزن السري لنقل المعلومات. اذ يتم استخدام خوارزمية خاصة لتشفير البيانات، لزيادة حماية الرسائل وتقليل المساحة اللازمة لخزنها 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 اقل من البيانات الأولى. وفي هذا البحث سوف نتناول الطريقتين السابقتين (الكبس، التشفير) اذ تم استخدام خوارزمية الخصوصية المتفوقة (PGP) للتشفير التي تستخدم خوارزمية الكبس (LZ77) Lempel-Ziv 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 المعلومات (information theory)، أمنية الحاسوب (computer security) وكذلك الهندسة [5].

التغيير الرئيسي الذي اثر 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 تراسلها.

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

الحاجة الى علم التشفير

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

الوثوقية (Confidentiality) والتخويل (Authentication) الصلاحية (Authorization) وتكامل البيانات (Data Integrity) عدم الإنكار (Non-Repudiation) [3].

* Corresponding author at: University of Mosul - College of Computer Science and Mathematics;

كبس البيانات Data Compression

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

كبس البيانات باستخدام ZIP

تستخدم خوارزمية الخصوصية المتفوقة الكبس باستخدام Zip. خوارزمية ZIP و أنواعها من الخوارزميات الأخرى و تعتمد على تقنية النافذة المؤقتة (Sliding Window Buffer) التي يحتوي على النص المعالج حديثاً. و يشار إلى هذه الخوارزمية بـLZ77.

تستغل LZ77 و أنواعها الكلمات والمقاطع المكررة ضمن النص، وعند حدوث تكرار فان 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 الكبس تكون من خلال هذه الرموز و استنتاج النص الأصلي [13],[11],[1].

خوارزمية الكبس(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، فان الحرف الأول في مخزن التقدم يخرج و يزحف إلى المخزن المؤقت و الرف الأول في المخزن المؤقت يخرج من المخزن. أما إذا وجد تطابق فإنها تكمل إلى اعلي حد للتطابق ثم تمثل بشكل ثلاثي.

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

الكبس و التشفير Compression And Encryption

تستخدم خوارزميات الكبس مع خوارزميات التشفير للأسباب التالية:

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

٢- يحتاج التشفير إلى وقت كبير عادة ،وكبس الملف قبل التشفير سوف يسرع عملية التشفير.

٣- ويفترض أن تتم عملية الكبس قبل التشفير،وإذا كانت خوارزمية التشفير جيدة فان النص المشفر سوف يكون غير قابل للكبس وسوف يظهر كأنه بيانات عشوائية [9].

خوارزمية الخصوصية المتفوقة Pretty Good Privacy PGP

٤- 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 (Key Distribution) [2]. كما أنها تسمح بالتشفير/فك التضفير والتوقيع الالكتروني مما يجعل البريد الالكتروني يحضى بكثير من الأهمية [12].

تحقيق الوثوقية والسرية في الخصوصية المتفوقة:

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

٦- ان خوارزمية الخصوصية المتفوقة (PGP) تستخدم الكبس قبل التشفير، وهذا له فائدة في حفظ مساحة خزن الملفات والاتصال عبر الشبكة. يشار إلى الكبس بـZ و إلى فك الكبس Z^{-1} تشفير الرسالة يكون بعد تطبيق الكبس لتعزيز سرية التشفير ولان الرسالة المكبوسة تكون اقل طولاً من النص الصريح، وبالتالي فان تحليل الشفرة يكون أصعب. والشكل (١) يوضح وظائف الـ PGP في طور الوثوقية.

الجانب العملي:

٧- بعد التعرف على الجانب النظري سنتناول في هذا البحث الجانب العملي والتطبيقي للخوارزميات التي تم استخدامها وهي خوارزمية RSA و خوارزمية LZ77 والشكل (٢) يوضح المخطط العام للنظام.

اضافة التوقيع الرقمي والتشفير باستخدام خوارزمية RSA

٨- استخدمت هذه الخوارزمية لإضافة توقيع باستخدام المفتاح الخاص للمرسل(d)، وإتباع الخوارزمية الاتية لحساب (x^d mod n).بعد

إضافة التوقيع الرقمي يتم تشفير الرسالة باستخدام المفتاح العام للمستلم ويمثل(e) وهو عدد أولي (prime number) والمفتاح الخاص (d) لفك الشفرة.

٩- أساس عمل هذه الخوارزمية أنها تأخذ القيمة وترفعها للقوة (power of)،للمفتاح العام أو الخاص ولتسريع التنفيذ تم إتباع الخوارزمية الآتية لحساب(x^e mod n)حيث تمثل xالعدد نفسه أما ei(لكل i=0.....m)فهي سلسلة الأرقام الثنائية للعددe كما موضح بالشكل (٣).

الكبس باستخدام خوارزمية LZ77

١٠-في البداية نضع في المخزن المؤقت عدداً من الأحرف بعدد Kالذي 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 المخزنين و عند حدوث تشابه نحسب موقع أول حرف من سلسلة الأحرف المتشابهة،ونستمر إلى أن نصل إلى اختلاف في الأحرف بين المخزنين.نخزن عدد الأحرف المتشابهة في المخزنين ثم نخزن مؤشر (Indicator) الذي يشير إلى وجود كبس 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 أخر حرف في مخزن التقدم يكون أول حرف من النص المتبقي الذي لم يحدث له معالجة. وهكذا نستمر إلى نهاية النص. وقد تم حساب نسبة الكبس وهي نسبة حجم الملف قبل الكبس على حجم الملف بعد الكبس كما في المعادلتين(1) و(2): حجم الملف قبل الكبس:

$$\text{نسبة الكبس (Compression Ratio)} = \frac{\text{حجم الملف قبل الكبس}}{\text{حجم الملف بعد الكبس}} \quad \ldots(1)$$

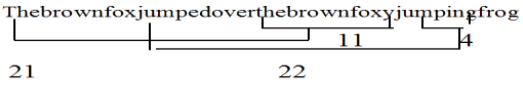
وحجم الملف بعد الكبس:

$$\text{نسبة الزيادة في حجم الملف} = \frac{\text{حجم الملف قبل الكبس} - \text{حجم الملف بعد الكبس}}{\text{حجم الملف قبل الكبس}} * 100 \quad \ldots(2)$$

ولتوضيح عمل الخوارزمية نفرض المثال الأتي[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 المتتابعة المتكررة. وفي حالة وجود تكرار نستمر إلى نهاية التكرار المتسلسل، بمعنى أخر، في كل مرة يحدث تكرار نحاول أن نأخذ اكبر 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 للتكرار هو قبله بـ 21 حرفاً و طول الأحرف المتكررة هو 11. الجزء المتبقي من الرسالة المكبوسة هو y، أما الأحرف الخمسة التي تليها فسوف تبدل بمؤشر22 وطول 4 وهو للفراغ و المقطع jump وتكمل باقي الرسالة. الناتج النهائي للرسالة المكبوسة يكون:



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

خوارزمية الكبس باستخدام LZ77

١-البداية

٢-قراءة النص المدخل في ملف من نوع TXT.

٣- تقطيع النص المدخل إلى حروف وخزنه في مصفوفة جديدة STR.

٤- 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)

٥- تعريف مصفوفة لخزن عدد الحروف المكررة 0 = (0)

Length (1)= 0 ،Length

٦-تعريف مصفوفة لخزن الحروف Char(0)=o، Char(1)=Str(1)،

٧- تعريف مؤشر (id) يشير الى الموقع الثاني في النص الأصلي

٨- كرر الخطوات من ٩- ١٦حتى نهاية النص

٩- قراءة النص ابتدا من الموقع الثاني،

١٠- تعريف متغير Index= 0 يشير الى الموقع الثاني في النص المراد كبسه، والمتغير N=0 ليمثل عدد تكرار الكلمات

١١- كرر الخطوة ١٢ طالما j < i-1

١٢- اذا كانت Str(j)=Str(i) و ndex=0 فان index=i-j(لإيجاد offset)،j=j+1

١٣- N=N+1

١٤- اذا كانت N<=2 اذهب إلى الخطوة ١٦

١٥- offset(id)= index و length(id)=N و Char(id)=Str(i) و id=id+1

١٦- i=i-1 و offset(id)=0 و

length(id)=0 و Char(id)=Str(i) و id=id+1 يعني لايوجد تكرار للحرف.

١٧- i=i+1

١٨- كرر الخوة ١٩ طالما i=id-1

١٩- اطبع offset(i) و length(i) و Char(i)

٢٠- النهاية

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

١. اقرأ النص المكبوس Doc.txt
٢. كرر الخطوات ٣-١٥ حتى نهاية الملف
٣. اطبع Offset(i) و Length(i) و Char(i)
٤. i=i+1
٥. نعرف مصفوفة Str =0 لخزن النص بعد فك الكبس
٦. J=0
٧. كرر الخطوات ٨-١٣ طالما(j<i-1)
٨. إذا كان offset(j)=0 (أي لا يوجد كبس) ،اذهب إلى الخطوة ١٠
٩. نعرف المتغير K ، K=j-offset(j) (لنحدد بداية الكلمة المكبوسة) ،اذهب الى الخطوة ١١
١٠. Str=Chr(j)
١١. نعرف متغير K ، K1=K
١٢. اذا كان K1<length(j) فان Str=Char(K1)
١٣. K1=K1+1
١٤. j=j+1
١٥. Str=Str&Char(i-1)
١٦. 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 ستكون نتيجة التشفير والكبس والتوقيع مخزونة في ملف كما في الشكل (6). وعند استلا م هذا الملف من قبل الشخص المخول سيقوم بفك الكبس والتشفير في الشكل (٧). يلاحظ انه قد تم استرجاع النص الأصلي.

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:

١. 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 فان الكبس يقلل من حجم النص ليسرع عملية التشفير التي تلي الكبس.

٢. وعند تحليل نتائج التطبيق اتضحت ضرورة الأخذ بنظر الاعتبار عدم وجود الفراغ، أي إدخال الكلمات المراد تشفيرها وكبسها بدون فراغات.

٣. باستخدام خوارزمية الخصوصية المتفوقة تم الحصول على مستوى أمنية أعلى.

٤. 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.

٥. ان استخدام لغة C ++ أعطى البرنامج صورة متماسكة وسهلة التداول مقارنة بجمل الانتقال بين السطور المستخدمة في لغتي بيسك و فورتران.

المصادر

١. إسماعيل، ياسين حكمت (2004). دراسة الوثوقية وتحقيقها باستخدام التواقيع الرقمية .٣٤. رسالة ماجستير، كلية علوم الحاسبات والرياضيات ، جامعة الموصل.

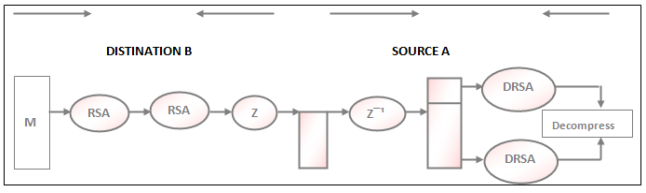
٢. الحسو، شهد عبد الرحمن (2004). تصميم *نظام هجين بالاعتماد على النصوص*.٥٤. *رسالة ماجستير*، كلية علوم الحاسبات والرياضيات، جامعة الموصل.

٣. الحمامي، علاء حسين ومحمد علاء الحمامي (2008). *إخفاء المعلومات*.١٨ إثراء للنشر والتوزيع، عمان.

٤. الشاهين، ناهي يوسف ونديم شاهين (2004). *دراسة وتصميم خوارزمية تشفير البيانات المنقولة عبر الشبكة*. مجلة جامعة دمشق للعلوم الهندسية، المجلد ٢٠: العدد الثاني.١٢-٢٢.

٥. حسين، عبد الأمير خلف(2010). *طرق التشفير للمبتدئين*.٣٣. 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,.
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.

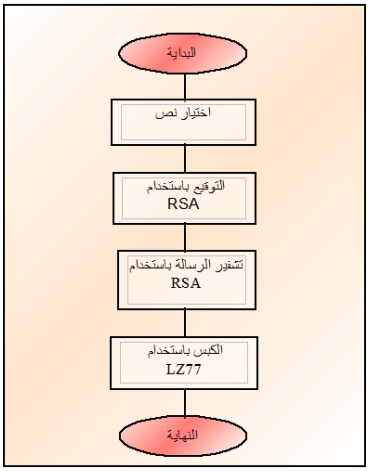


الشكل (١): وظائف الـ PGP في طور الوثوقية

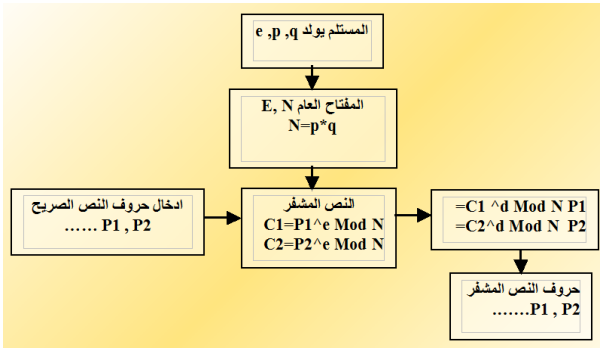
ملحق-١: برمجية البحث

```
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))
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
  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
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
```

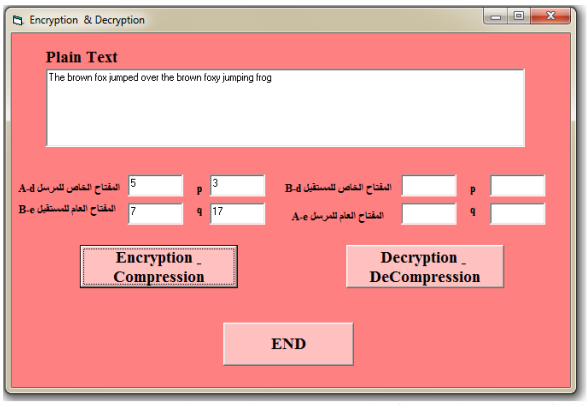
P- ISSN 1991-8941 E-ISSN 2706-6703 (JUAPS) مجلة جامعة الانبار للعلوم الصرفة Open Access
2013,(7), (1) :33-39



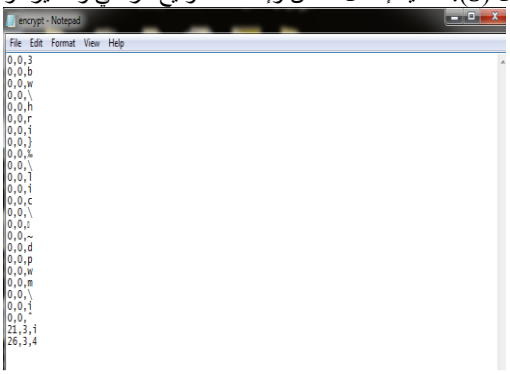
الشكل (٢): المخطط العام للنظام



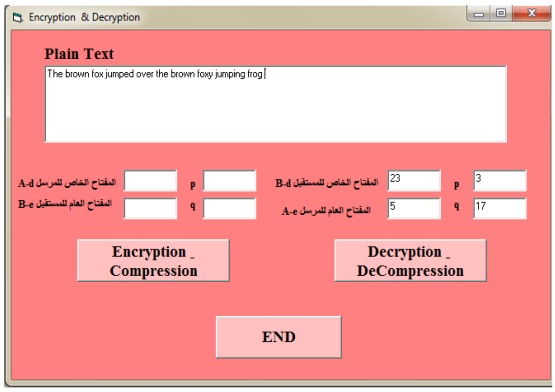
الشكل (3): إضافة التوقيع والتشفير باستخدام خوارزمية RSA



الشكل (5): عملية إدخال النص وإضافة التوقيع الرقمي وتشفيره وكبسه



الشكل (٦): نتيجة التشفير والكبس



الشكل (7): فك الكبس والتشفير

Offset	Length	Character
		T
0	0	h
0	0	e
0	0	b
0	0	r
0	0	o
0	0	w
0	0	n
0	0	f
0	0	o
0	0	x
0	0	j
0	0	u
0	0	m
0	0	p
0	0	e
0	0	d
0	0	o
0	0	v
0	0	E
0	0	R
21	11	y
22	4	i
0	0	n
0	0	g
0	0	f
0	0	r
0	0	o

الشكل (٤): ملف ناتج عملية الكبس

P- ISSN 1991-8941 E-ISSN 2706-6703 (*JUAPS*) مجلة جامعة الانبار للعلوم الصرفة Open Access
2013,(7), (1) :33-39

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