مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (30) العدد (1) 2008
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (30) No. (1) 2008

تطوير تطبيق لتشفير البيانات بالاعتماد على خوارزميتي التشفير IDEA وDESX

الدكتورة كندة أبو قاسم*
الدكتور إبراهيم الشامي**
عبد الحميد قريعة ***

(تاريخ الإيداع 10 / 12 / 2007. قُبِل للنشر في 2008/3/4)

□ الملخّص □

تبحث هذه الورقة في دراسة تطبيقات المصادر المفتوحة التي تؤمن تشفير البيانات والرسائل وتركزعلى نقاط قوتها وضعفها ثم تصميم تطبيق جديد يقوم بحل هذه المشاكل وذلك من خلال النقاط الآتية:

- تصميم مولد مفاتيح تشفير عشوائية بالاعتماد على تقانات منيعة ضد الهجوم عليها ، والتأكد إحصائياً من تحقق العشوائية في هذه التقانات، والعمل على جعل توليد المفاتيح أكثر أماناً .
- توليد مفتاح تشفير جديد لكل جلسة عمل .
- بناء خوارزمية تشفير جديدة تعتمد على تعديل خوارزميتي التشفير IDEA و DESX للحصول على خوارزمية تشفير أبسط وأسرع وتحقق أمن البيانات ، وهذه الخوارزمية منيعة ضد الكسر بطريقة الهجوم المباشر Brute Force Attack .
- تطوير تطبيق لتشفير البيانات يعتمد على الخوارزمية الجديدة يعمل في بيئتي ويندوز ودوس Windows ,Dos.

الكلمات المفتاحية: خوارزميات التشفير ، التبييض ، الجمع الثنائي ، توليد المفاتيح ، سرية جيدة جداً.

*أستاذ مساعد – قسم هندسة الحاسبات والتحكم الآلي– كلية الهندسة الميكانيكية والكهربائية– جامعة تشرين– اللاذقية– سورية.
** أستاذ مساعد – قسم هندسة الحاسبات والتحكم الآلي– كلية الهندسة الميكانيكية والكهربائية– جامعة البعث– حمص– سورية.
*** طالب ماجستير– قسم هندسة الحاسبات والتحكم الآلي– كلية الهندسة الميكانيكية والكهربائية– جامعة تشرين– اللاذقية– سورية.

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (30) العدد (1) 2008
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (30) No. (1) 2008

Using IDEA and DESX Standards in Developing Security Applications

Dr. Kinda Abu Kassem*
Dr. Ibrahim Chami **
Abdel Hamid Kreaa ***

(Received 10 / 12 / 2007. Accepted 4/3/2008)

□ ABSTRACT □

This paper presents a new application for data encryption after studying the available open source applications, to try to find and overcome their weakness. We shall focus on these points:

- Design random key generating program, using a technique immune to physical attacks. A statistical study verifies randomness of the technique used and makes key generation even more secure.
- Generating a new key for each session.
- Develop a simpler, faster, and more secure encryption algorithm (Resistant to brute force attacks) through modifying IDEA & DESX standards.
- Developing an application using the new algorithm with Windows-DOS inter-portability.

Key Words: Encryption Algorithms, Whitening, XOR, Key Generation, PGP.

* **Associate Professor, Department of Computer Engineering and Automatic Control, Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia, Syria.**
****Associate Professor, Department of Computer Engineering and Automatic Control, Faculty of Mechanical and Electrical Engineering, Albaath University, Homs, Syria.**
*****Postgraduate Student, Department of Computer Engineering and Automatic Control, Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia, Syria.**

مقدمة:

تزداد الحاجة في أيامنا هذه لاستعمال خدمات التشفير، حيث يطلب من التشفير، فضلاً عن توفير السرية تحقيق مهام أخرى منها [1] :

- الإستيقان Authentication: يجب أن يكون ممكناً لمستقبل الرسالة التحقق من هوية منشئها، أي يجب أن لا يتمكن دخيل من نسبها إلى غير مرسلها.
- السلامة Integrity: يجب أن يكون ممكناً لمستقبل الرسالة التحقق أنها لم تتغير في أثناء النقل ، أي يجب ألا يتمكن دخيل من تعديل الرسالة الأصلية.
- عدم الإنكار Nonrepudiation: يجب ألا يكون المرسل قادراً فيما بعد على إنكار إرسال الرسالة التي أرسلها فعلاً .

والتشفير هو أداة مهمة جداً في مجال الدفاع عن أمن البيانات حيث من الصعب تحقيق المهام السابق ذكرها بأي وسيلة أخرى.

الشكل (1) تشفير وفك تشفير رسالة سرية.

وخوارزميات التشفير والتي تسمى أحياناً المشفرات هي التابع الرياضي المستخدم في التشفير وفك التشفير. وإذا كان أمن الخوارزمية قائماً على الاحتفاظ بطريقة عملها سرية تسمى بالخوارزمية المقيدة. وقد كان للخوارزميات المقيدة أهمية تاريخية ولكنها غير ملائمة للمعايير الحالية. إذ لا يمكن لمجموعة كبيرة ومتغيرة من المستخدمين استخدامها، لأنه في كل مرة يترك فيها مستخدم المجموعة على بقية أعضائها الانتقال إلى خوارزمية مختلفة. وإذا باح شخص من المجموعة بالسر مصادفة فعلى البقية تغيير خوارزميتهم .

الشكل (2) تشفير وفك تشفير رسالة باستخدام المفتاح.

حلت طرق التشفير الحديثة هذه المشكلة باستخدام المفتاح Key. ويكمن أمان هذه الخوارزميات برمته في المفتاح (أو المفتاحين)، ولا يستند شيء منه إلى تفاصيل الخوارزمية. وهذا يعني أنه يمكن نشر الخوارزمية وتحليلها،

ويمكن إنتاج المنتجات التي تستخدم الخوارزمية بكميات كبيرة. ولا يهم أن يعرف المتجسس خوارزميتك، لأنه لا يستطيع قراءة رسالتك إذا لم يعرف مفتاحك الخاص.

و سيتم في هذا البحث دراسة بعض خوارزميات التشفير وما هي التطبيقات المتداولة عالمياً والبحث عن الثغرات الأمنية ونقاط الضعف فيها ثم تطوير تطبيق يتجاوز هذه المشكلات .

تم إجراء هذا البحث في قسم هندسة الحاسبات والتحكم الآلي في كلية الهندسة الميكانيكية والكهربائية في جامعة تشرين خلال العام 2006-2007.

هدف البحث وأهميته:

يهدف هذه البحث بشكل أساسي إلى تصميم نظام تشفير يضمن أمن البيانات من خلال بناء خوارزمية تشفير جديدة ترتكز على خوارزميات تشفير تناظري معتمدة ومصدقة وموثقة عالمياً (IDEA و DESX)، وتوليد مفاتيح تشفير عشوائية باستخدام تقنيات منيعة ضد الهجوم الفيزيائي عليها(قياس فعالية العمل على لوحة المفاتيح Keyboard Latency).

طريقة البحث ومواده:

يبدأ البحث بالتعرف على خوارزميات التشفير ومفاتيح التشفير ودراسة تطبيقات أمن البيانات المتداولة ثم طريقة توليد مفاتيح تضمن العشوائية وخطوات عمل نظام أمان مقترح وقراءة نتائجه وذلك ضمن الخطوات الآتية:

1- نظرة عامة عن خوارزميات التشفير وطرق اختيارها والمفاهيم المرافقة.

2- شرح مبدأ عمل خوارزميات التشفير التناظري IDEA وDESX.

3- نظرة على أهم التطبيقات المتداولة في مجال أمن البيانات PGP وGnuPG والتعرض لنقاط الضعف الموجودة فيها.

4- تصميم مولد مفاتيح عشوائية والتحقق من وثوقية نتائجه.

5- دراسة خوارزمية عمل النظام المقترح وخطوات تنفيذها.

6- تطوير تطبيق لتشفير البيانات يعتمد على الخوارزمية الجديدة يعمل في بيئتي ويندوز ودوس Windows ,Dos.

7- تحليل النتائج ومناقشتها والاستنتاجات والتوصيات المقترحة .

1- الأسس المعتمدة في اختيار خوارزمية التشفير :

توجد عدة خيارات لتقييم الخوارزميات واختيارها [1]:

1- يمكن اختيار خوارزمية منشورة على أساس الاعتقاد بأن الخوارزمية المعلنة على الملأ قد أمعن النظر فيها كثير ممن يعملون في مجال التشفير، وإذا لم يكسرها أحد حتى الآن فلا بد من أنها جيدة جداً .

2- و يمكن الوثوق بمنتج خوارزميات تشفير على أساس الاعتقاد بأن للمنتج الشهير سمعة يحافظ عليها، وبأنه من غير المحتمل أن يضحي بها ببيع أجهزة أو برامج ذات خوارزميات متدنية الأمان.

3- و يمكن الوثوق بمستشار مستقل على أساس الاعتقاد بأن المستشار النزيه مؤهلاً تأهيلاً جيداً للقيام بتقييم للخوارزميات المختلفة.

4- و يمكن الوثوق بالجهات الحكومية المختصة على أساس الاعتقاد بأن الحكومة جديرة بالثقة، وبأنها تختار الخوارزميات الأنسب.

5- و يمكن أن نقوم بكتابة الخوارزمية بأنفسنا على أساس الاعتقاد بأن مقدرتنا في هذا المجال لا تقل عن مقدرة أحد آخر، وأننا يجب أن لا نثق إلا بأنفسنا.

بعد أخذ ما ورد أعلاه بالحسبان وعلى اعتبار أن الخوارزمية DES(Digital Encryption Standard) والخوارزمية IDEA (International Data Encryption Standard) هما أكثر الخوارزميات المصدقة والمسجلة تداولاً حول العالم. فقد قمت ببناء خوارزمية جديدة أبسط وأسرع عبارة عن مزيج مما تقوم به هاتان الخوارزميتان بما في ذلك التبييض والتدوير وتشفير العلب (Block) والتشفير التسلسلي (Stream).

استخدام ضغط البيانات مع التشفير:

يوجد سببين منطقيين لاستخدام خوارزمية ضغط للبيانات مع التشفير وهما:

1- يعتمد تحليل التشفير على استغلال التكرار في حروف النص (بعض الخوارزميات)أو مايسمى فائض اللغة، وضغط ملف قبل تشفيره يخفض ذلك التكرار.

2- التشفير يستهلك وقت ليس بالقليل وضغط الملف قبل تشفيره يسرع العملية كلها.

لتوضيح الفكرة سنقوم هنا بالتجربة على النص المصدري للبرنامج والذي يتكون من حوالي 1000 سطر بما يحتويه من تكرار كبير لكلمات مثل cout, printf, get, read, write,… وهكذا ونظراً لأنه كلما زاد تكرار الكلمات زادت إمكانية ضغط النص.

و هذا يعني أنه لا يوجد حدود لمجال مجموعة الحروف المدخلة (المقصود بأنه لا يوجد حدود أن الدخل قد يكون أي من رموز ASCII (code 0-255) وهو هنا حوالي 200 رمز مختلف لأي دخل معتاد ولكن بعد الجولة الثانية من خوارزمية تشفير المقطع المصممة نقوم بتحديد النص الخرج إلى حوالي 100 رمز مختلف ASCII(Code 33-132) وهذا يدفعنا إلى استنتاج الملاحظة الآتية:

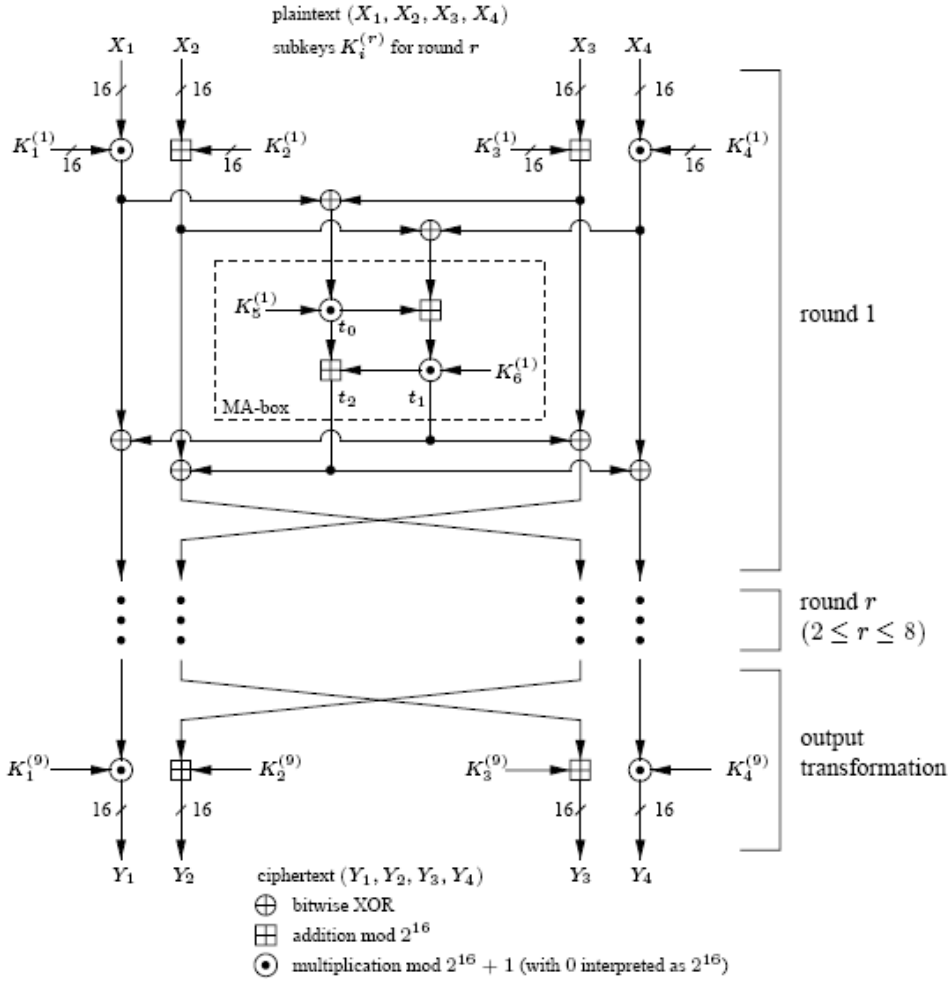
يوجد العديد من المحارف المتكررة في النص المشفر (تخيل تمثيل نص مؤلف من 17000 حرف (16747 حرف) يتركب من 200 رمز مختلف سنقوم بتمثيله (بتشفيره) باستخدام مجموعة محارف مركبة من 100 رمز، لهذا يكون حجم النص المشفر تقريباً ضعف حجم النص الصريح .

و يظهر الجدول (2) مشاهدات عند تشفير النص المصدري للبرنامج بدون ومع ضغط باستخدام التطبيق الذي طورناه.

2-1- الخوارزمية IDEA :

ظهر أول تجسيد للخوارزمية IDEA في عام 1990 وقد صممها كل من Xuejia Lai و James Massey وسميت حينئذ مقترح معيار تشفير (PES -Proposed Encryption Standard)، وبعد تقوية هذا المقترح تمت تسمية الخوارزمية IDEA (International Data Encryption Standard) في عام 1992.

تعمل الخوارزمية على علب من المعطيات بـ 64 خانة وتبدله إلى نص مشفر بـ 64 خانة .



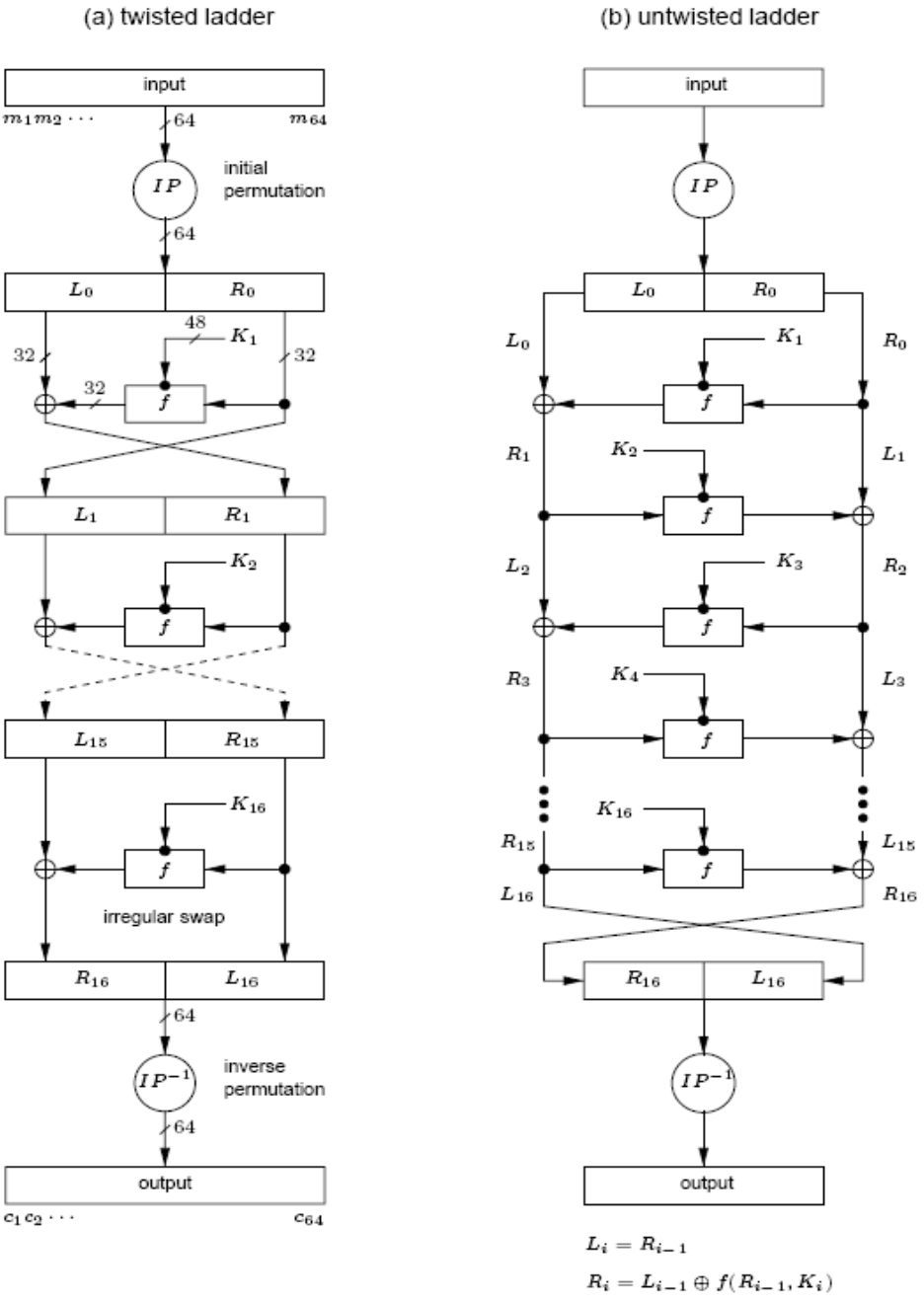
الشكل (3) يظهر مسار الخوارزمية IDEA [2].

إن طول مفتاح IDEA يساوي 128 خانة أي ما يزيد على ضعف طول مفتاح الخوارزمية DES [2] وبافتراض أن الهجوم المباشر (Brute Force Attack) هو الطريقة لمهاجمة الخوارزمية فإنه يتطلب 2^128 أي 10^38 عملية تشفير، صمم شريحة تستطيع اختبار مليار مفتاح في الثانية واستخدم مليار شريحة من هذا النوع فتجد أن إيجاد المفتاح يحتاج إلى 10^13 سنة وهي مدة أطول من عمر الكون.أما إذا استخدمنا مصفوفة تحوي 10^24 شريحة أمكن إيجاد المفتاح في يوم واحد، ولكن ليس ثمة ما يكفي من ذرات السيليكون في الكون لبناء تلك الآلة.

قام مصممو نظم التشفير بتحليل الخوارزمية IDEA لقياس قوتها ضد الهجوم بالتحليل التفاضلي (Differential Cryptanalysis) واستنتجوا مناعتها وقد صرح العالم بروس شناير أن هذه الخوارزمية في رأيه هي الأفضل والأكثر أماناً بين خوارزميات تشفير المقطع لغاية تاريخه [3].

2-2- الخوارزمية DESX :

و هي طراز معدل من الخوارزمية DES (الشكل (4) يظهر مخطط الخوارزمية DES [2]) ابتكرته شركة RSA Data Security Inc واستخدمته ضمن برنامج أمن البريد الإلكتروني منذ العام 1986. تستخدم هذه الخوارزمية تقانة تسمى التبييض(Whitening) وهو اسم تقنية الجمع الثنائي XOR لمشتقات مفتاح ما إلى مدخل خوارزمية علب (Block)، والجمع الثنائي لمشتقات مفتاح آخر إلى مخرجاتها، وقد استخدمت هذه التقنية أول مرة في الخوارزمية DESX ثم استخدمت في خوارزميات أخرى. $\text{DES-X}(M) = K_2 \oplus \text{DES}_K(M \oplus K_1)$



الشكل (4) يظهر مسار الخوارزمية DES [2].

و كان القصد من هذه التقانة منع محلل التشفير من الحصول على زوج من النص الواضح والنص المشفر بالخوارزمية المعنية. فهي لا ترغم المحلل على تخمين مفتاح الخوارزمية فحسب بل على تخمين واحدة من قيم التبييض أيضاً. هذه التقانة زادت قوة الخوارزمية الأصل DES ضد الهجوم بالتحليل التفاضلي والتحليل الخطي (Liner Cryptanalysis)، فضلاً عن زيادة قوتها أمام الهجوم المباشر [3] .و نظراً لوجود الجمع الثنائي قبل التشفير بخوارزمية العلب وبعدها، فإن هذه التقانة ليست عرضة لهجوم التلاقي في الوسط (Meet in the Middle Attack).

هذه الملاحظات المهمة عن هذه الخوارزميات، فضلاً عن ما تعاني منه التطبيقات المستخدمة حالياً من ثغرات أمنية [4] دفعتني لإعداد تطبيق برمجي بشكل مشابه ولكن يعمل بسرعة أكبر. حيث قمنا ببناء خوارزمية جديدة تعتمد جولة واحدة من جولات الخوارزمية IDEA مع استخدام مفتاحين فرعين مشتقين من مفتاح واحد طوله 10 خانات.(لذا فإن زيادة طول المفتاح إلى 32 خانة أو 64 لن تحدث أي تأثير ضار في الخوارزمية) إضافة إلى جولة من جولات الخوارزمية DESX مما يجعل الخوارزمية الجديدة تعمل بشكل أسرع وو تحقق أمان أعلى كما تبين نتائج البحث .

و بالتالي التطبيق الذي يعتمد هذه الخوارزمية سيكون أكثر أماناً من التطبيقات المستخدمة حالياً والتي أشارت الأبحاث إلى ضرورة التعديل في طريقة عملها لزيادة الأمان فيها [4].

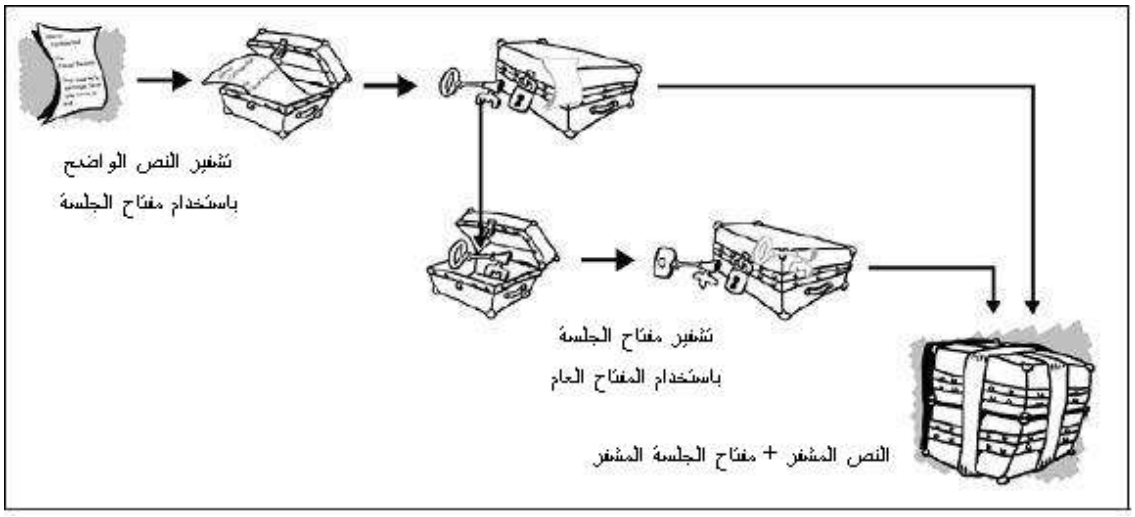
3- نظرة على أهم التطبيقات المتداولة في مجال أمان البيانات :

نظراً لأهمية الخصوصية فإنه من المطلوب تحقيق أمن المعلومات وبشكل قانوني. غالباً ما يكون الاتصال عبر الإنترنت لا يؤمن الخصوصية. ويقوم تشفير البريد الإلكتروني وتشفير الملفات بتأمين مستوى عالي من الأمن للبيانات عبر الإنترنت. ولكن غالباً ما تكون البرمجيات الخاصة بالشركات التقانية التي تقدم خدمات التشفير والخدمات المماثلة (مثل PGP Corporation) مكلفة وتحمل ميزانيات المؤسسات مبالغ طائلة [5] .

3-1- سرية جيدة جداً PGP (Pretty Good Privacy):

PGP هو برنامج مجاني لأمن البريد الإلكتروني صممه في الأصل Philip Zimmermann. يستخدم البرنامج الخوارزمية IDEA للتشفير والخوارزمية RSA لإدارة المفاتيح والبصمة الرقمية [6] .

يستخدم PGP كلمة مرور لتشفير مفتاح المستخدم الخاص على جهازه. وتستخدم كلمة المرور لفك تشفير المفتاح الخاص واستخدامه، وهي كلمة يجب أن لا ينساها المستخدم ولا يستطيع معرفتها الآخرون.و هذه النقطة مهمة بدرجة كبيرة؛ لأنه إذا نسي كلمة المرور لا يمكن الاستمرار في العملية؛ لأن المفتاح الخاص يصبح عديم الفائدة بدونها. فالأمان في PGP جيد إلى حد أنه يبعد الجميع عن ملفاتك، وأيضاً يبعدك أنت عنها إذا نسيت كلمة المرور.



الشكل (5) يظهر كيفية عمل PGP .

و هو تطبيق من المصادر المفتوحة (Open Source) ومجاني للاستعمال الشخصي فقط ومدفوع القيمة في حال الاستخدام التجاري وهو يعتبر مكلفاً.

3-2- التطبيق GnuPG :

تم بناء هذا التطبيق بشكل مشابه للبرنامج PGP [7] و هو من المصادر المفتوحة أي أنه بإمكان أي مستخدم تحميله وتعديله والعمل عليه وبشكل مجاني.[8]

3-3- الثغرات الأمنية في هذه التطبيقات :

يناقش موقع PGP FAQ على الإنترنت عدة أنواع من الهجوم على PGP بما في ذلك مسجلات المحارف المضغوطة في لوحة المفاتيح وقراءة الذاكرة وقراءة مسجلات القرص الصلب، إضافة إلى الفيروسات التي تسجل ضغطات لوحة المفاتيح ومنها مخصص لسرقة كلمة المرور والمفاتيح في PGP [9] و هو ما ينطبق حرفياً على التطبيق GnuPG .

1- خوارزميات التشفير المعتمدة :

معظم نسخ البرنامج PGP مجهزة باختيارات كثيرة من الخوارزميات حتى المعتمدة والمصدقة ويمكن إضافتها إليه، بينما البرنامج GnuPG على الرغم من كونه يأتي مع مكتبة كبيرة من الخوارزميات فإنه يفتقد إلى الخوارزميات المعتمدة والمصدقة حكومياً ومن أهمها IDEA [4] مما يفقده قدراً كبيراً من الوثوقية.

2- مولدات العشوائية :

لا توجد ملاحظة صريحة حول مولدات للأرقام للعشوائية في أنظمة التشغيل ويندوز نظراً لضعف هذه النقطة في هذه الأنظمة [5] مما يهدد أمان البيانات برمتها .

تنويه: مولدات الأرقام العشوائية في أي مترجم هي ليست مصادر حقيقية للعشوائية

3- النسخ المزورة :

بما أن هذا البرنامج PGP وGnuPG هما من المصادر المفتوحة فهناك نسخ مزورة منهما مطروحة عبر الإنترنت، فإذا لم تكن متأكداً من أن نسختك تأتي من مصدر موثوق، فلن يكون مستغرباً أن تلاحظ يوماً أن كلمة المرور تم إرسالها إلى مهاجم ما عبر البريد الإلكتروني في اللحظة التي تكون فيها مرتبطاً مع الشبكة العالمية.

4- مسجلات لوحة المفاتيح (Keyloggers) :

و هي عبارة عن برامج تركب على الحاسب بدون معرفة المستخدم وتقوم هذه المسجلات بتسجيل المحارف من لوحة المفاتيح وتمرير كلمات السر والبريد إلى متجسس ما [10] .

4- تصميم مولد مفاتيح تشفير يعتمد توليد قيم عشوائية :

يعتبر أسلوب عمل الأشخاص على لوحة المفاتيح عشوائياً وغير عشوائي في آن، فهو غير عشوائي لدرجة أنه يمكن عده وسيلة تميز الشخص، ولكنه يعتبر عشوائياً لدرجة يمكن استخدامه لتوليد أرقام عشوائية. وفيما يأتي شرح ما قمنا به في هذا الاتجاه:

1- قياس الوقت بين الضغطات المتتالية على لوحة المفاتيح ثم أخذ قيم الخانات الأصغر أهمية، وقيم هذه الخانات ستكون أرقام عشوائية بشكل مقبول.(هذه التقنية لا يمكن استعمالها مع نظام UNIX لأن ضربات المفاتيح تمر عبر فلترة وآليات أخرى قبل وصولها إلى البرنامج ولكن تعمل بصورة جيدة في بيئة Windows و Dos .)

2- إيجاد مجموع هذه القيم العشوائية وتقسيمها على رقم عشوائي آخر بين 1 و 11 يتم توليده من خلال تابع Randomize(x) (حيث x قيمة متغيرة متعلقة بزمن الحاسب) في لغة C++ فإذا كان هذا الرقم أكبر من 5 يتم تقسيم المجموع عليه وإلا ضرب المجموع به .

3- حساب الرقم الثنائي للنتيجة وأخذ 10 خانات ذات المرتبة الدنيا واستخدامها بوصفها مفتاحاً خاصاً.

<u>قياسات إحصائية:</u>

في مرحلة تجريب البرنامج سيتم حفظ نتائج مولدات المفاتيح عند كل تشغيل بالاعتماد على أساليب عمل مستخدمين مختلفين.حيث سيقوم 5 مستخدمين بتوليد 20 مفتاح لكل مستخدم وعدد المفاتيح الإجمالي 100 مفتاح، سيتم إدخالها إلى قاعدة بيانات لإجراء المزيد من التحليل.

كل مجموعة معطيات من مستخدم محدد يتم تصفيتها للحصول على عدد مرات التكرار في العشرين مفتاح المولدة، سيتم حساب فعالية هذه التقانة بتقسيم عدد المفاتيح الفريدة على إجمالي عدد المفاتيح المولدة. ويتم حساب فعالية معطيات كل مستخدم مع وسطي الفعالية لكل خوارزمية توليد المفاتيح. علماً أنه تم الطلب من بعض المستخدمين أن يقوم بطباعة كلمة مختلفة في كل مرة بينما طلب من آخرين أن يقوم بتكرار الكلمة نفسها في كل مرة.
نقوم بحساب الفعالية لكل مستخدم والفعالية الإجمالية للطريقة :

الفعالية لكل مستخدم = عدد المفاتيح المكررة \ إجمالي عدد المفاتيح المولدة للمستخدم.

الفعالية الإجمالية = عدد المفاتيح المكررة \ إجمالي عدد المفاتيح المولدة.

و يمكن أن نلخص النتائج في الجدول **(1)**.

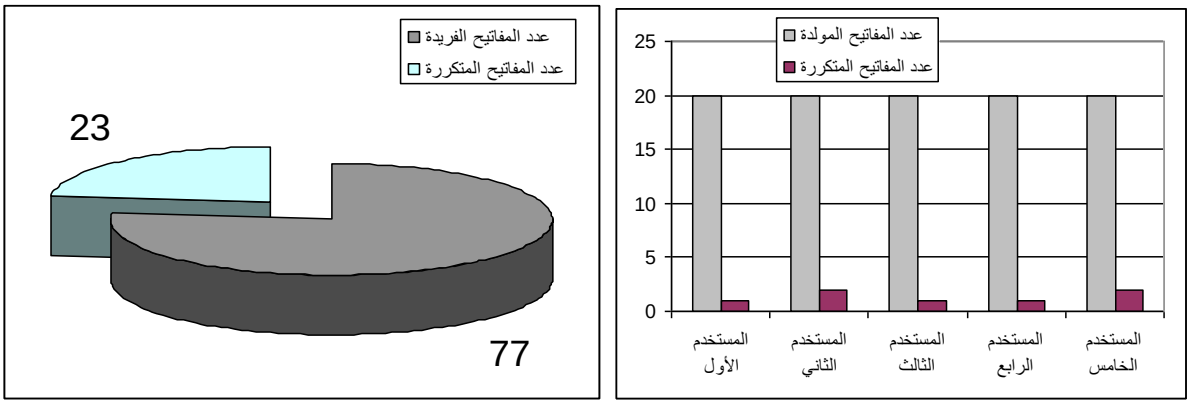
الجدول (1) يظهر نتائج فعالية توليد المفاتيح من قبل المستخدمين.

	عدد المفاتيح	عدد المفاتيح	الفعالية %

	المولدة	المتكررة	
المستخدم الأول	**20**	**1**	**95**
المستخدم الثاني	**20**	**2**	**90**
المستخدم الثالث	**20**	**1**	**95**
المستخدم الرابع	**20**	**1**	**95**
المستخدم الخامس	**20**	**2**	**90**

و بالحساب ظهرت أن الفعالية الإجمالية لتوليد المفاتيح هي 77% حيث وجد 23 تكرار.

و تظهر المخططات البيانية في الشكل (6)عدد المفاتيح المولدة والفعالية الإجمالية



الشكل (6) يظهر عدد المفاتيح المولدة والمفاتيح المتكررة لكل مستخدم والفعالية الإجمالية

5- خوارزمية عمل النظام المصمم :

5-1-معالجة النص الواضح :

قراءة ملف الدخل "PLAIN.TXT" بشكل تتابعي (كل خانة على حدى) وتوليد الخرج الموافق بالنظام الثنائي. بما أن مجموعة رموز ASCII تتألف من 256 رمز (0-255)، مما يستتبع أننا نحتاج إلى 8 خانات لتمثيل كل رمز في النظام الثنائي (8^2=256) .

5-2- إضافة متممات إلى النص الثنائي :

إن ترميز المحرف (ثمان خانات مقابل كل رمز من النص الواضح) يتم إتمامه إلى عشر خانات بإضافة أصفار بعد الخانة ذات المرتبة العليا (MSB Most Significant Bit). نظراً لكون تشفير المطيات الثنائية يتم بمفتاح خاص مكون من عشر خانات. ثم يتم حفظ الخرج الناتج في ملف "BINARY.TXT".

5-3- توليد المفتاح بقياس فعالية العمل على لوحة المفاتيح :

الغاية هنا الحصول على معطيات دخل عشوائية من لوحة المفاتيح. يعتبر الفارق الزمني بين كبس أزرار لوحة المفاتيح مصدر ممتاز للعشوائية. وتعتمد طريقتنا في توليد المفاتيح على تابع يقوم بجمع قيم أجزاء حقول الميلي ثانية للفارق الزمني بين ضغطتين متتاليتين على لوحة المفاتيح ثم ضرب أو تقسيم الناتج برقم عشوائي آخر يتم توليده حسب قيمة هذا الرقم العشوائي(إذا كان هذا الرقم أكبر من 5 يتم تقسيم المجموع عليه وإلا ضرب المجموع به)، ثم تحويل

الناتج النهائي إلى رقم ثنائي من عشر خانات، ومن ثم استخدامه بوصفه مفتاحاً خاصاً. هذا المفتاح يخزن في ملف آخر "PVTKEY.TXT" بصيغة ثنائية للرجوع إليه لاحقاً.

5-4- تشفير النص (المرحلة الأولى من التشفير تشفير تسلسلي):

يتم قراءة كل مقطع مكون من عشر خانات ثنائية تسلسلياً من الملف "BINARY.TXT" ويتم جمعه ثنائياً (XORed) مع المفتاح المكون من عشر خانات والذي تتم قراءته من الملف "PVTKEY.TXT". والعشر خانات الناتجة يتم تخزينها في الملف "XOR.TXT". هذه العملية غالباً ما يطلق عليها اسم التبييض (Whitening).

5-5- تطعيم (Addition of Salt) النص المشفر:

في الجولة الثانية من التشفير سنقوم باستخدام علبة من المعطيات بطول 16 خانة. لذا فإن عدد الخانات في الملف " XOR.TXT " يجب أن يكون عدد صحيح من مضاعفات العدد 16. من أجل هذه الغاية يتم إضافة n خانة أصفار إلى نهاية "XOR.TXT " حيث n يتم حسابها بالمعادلة التالية :

$$n = 16-\{ [(count-1)*10]\%16\}$$

حيث (count-1) : يمثل عدد الأحرف في النص الصريح.

* : تمثل عملية الضرب العادي.

% : تمثل الباقي بعد القسمة العادية.

و لجعل الأمر أكثر تعقيداً يمكن إضافة تطعيم بقيم أخرى غير سلسلة الأصفار. ويمكن ذلك بإضافة تابع آخر يحدد قيمة التطعيم التي يجب استخدامها بالاعتماد على قيم n المحسوبة من المعادلة السابقة.

5-6- توليد المفاتيح الفرعية من أجل تشفيرالعلب :

يتم توليد مفتاحين فرعيين K1 و K2 باستخدام المفتاح الخاص. حيث يتم إسقاط أول أعلى خانتين والخانات الثمان الباقية يتم إتمامها وحفظها بوصفها مفتاحاً K1. ثم يتم عكس قيمة K1 للحصول على K2 ، ورغم سهولة هذه الطريقة في توليد المفاتيح الفرعية إلا أنه يمكن بسهولة إضافة المزيد من التعقيد عليها.

5-7- تقسيم النص المشفر تسلسلياً إلى علب من أجل مضاعفة تشفيرها :

عدد الخانات في الملف "XOR.TXT " قابل للقسمة على 16. يتم معالجة كل مقطع مكون من 16 خانة بالتتالي. حيث يقسم كل مقطع إلى نصفين متساويين X1 و X2

المعطيات بالنظام الثنائي	1	0	0	1	0	1	0	1	0	1	1	0	0	1	0	0
أرقام الخانات الفرعية	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
الأجزاء X1,X2	X1								X2							
المقطع رقم 1	1															

الشكل (7) جدول يظهر مقطع واحد من 16 خانة مقسمة إلى x1 وx2 كل منها طوله 8 خانات.

5-8- تشفير النص الثنائي (المرحلة الثانية من التشفير علب):

يـتم إجـراء العمليـات الآتيـة بالإعتمـاد علـى X1,X2,K1,K2 ويـتم تخـزين النتيجـة فـي الملـف "CIPHER2.TXT" :

$$Y1 = X1 \oplus K1$$

$$Y2 = X2 \oplus K2$$

حيث

$\oplus$: تمثل عملية الجمع الثنائي XOR

Y1 وY2 يتم وصلها مع بعضها للحصول على مقطع ثنائي مؤلف من 16 خانة .و يتم تطبيق هذه العملية على كامل النص المشفر تسلسلياً .

5-9- المرحلة الأولى من معالجة النص المشفر :

طول الملف الثنائي "CIPHER2.TXT " هو بشكل أساسي من مضاعفات 16، وهو ما يعني أنه أيضاً من مضاعفات العدد 8. سنقوم في هذه المرحلة بقراءة 8 خانات بالتسلسل وتحويلها إلى ما يكافئها بالنظام العشري، قيمة العدد الناتج ستكون من المجال 0-255. ويتم إتمام كل مكافئ عشري إلى ثلاث خانات عشرية (فمثلاً يتم إتمام العدد 3 إلى 003 والعدد 52 إلى 052 ويبقى العدد 145 كما هو) ويتم كتابة الناتج في الملف "CIPHER3.TXT ".

5-10- إجراء تطعيم إضافي للنص :

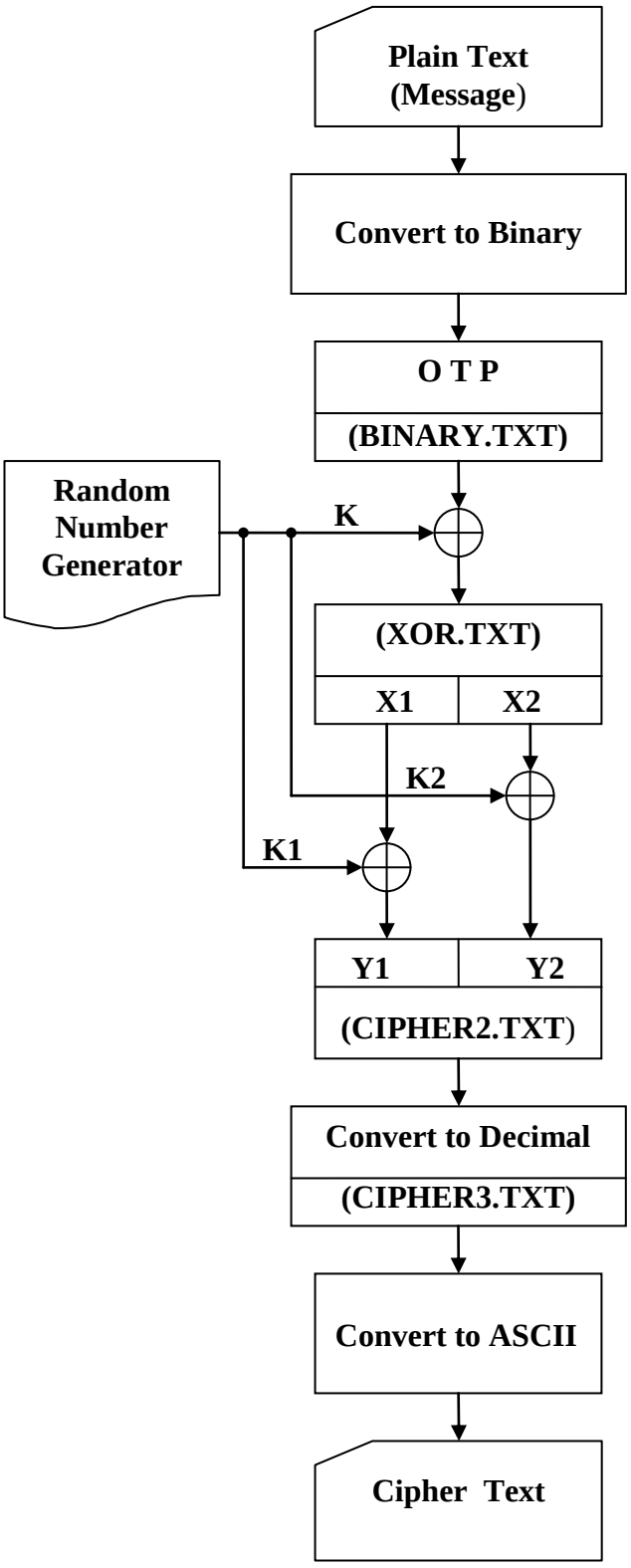
عدد المحارف الذي نحصل عليه في الملف "CIPHER3.TXT " يحتاج إلى ضربه بالعدد 2 من أجل المرحلة الثانية من تشفير المقطع. نظراً لكون كل رمز من الملف "CIPHER2.TXT " تم تمثيله بثلاثة أعداد في الملف "CIPHER3.TXT " ، فإن طول الملف "CIPHER3.TXT " هو من مضاعفات العدد 3. نقوم بضربه بالرقم 2 .

5-11- المرحلة الثانية من معالجة النص المشفر :

هنا يتم قراءة الملف "CIPHER3.TXT " تسلسلياً وهذه المرة كل خانتين معاً. وهكذا في كل قراءة نحصل على عدد صحيح بين 00 و99 ، نضيف 33 إلى كل عدد ونحصل عدد رمز الـ ASCII الموافق للعدد الناتج يتم كتابته في الملف "CIPHER.TXT ". والغاية الأساسية من إضافة العدد 33 هي تجنب الرموز ذات الأرقام القليلة كونها تحوي رموز غير قابلة للطباعة وكودات Esc وفواصل أسطر وفراغات ....إلخ .

يتم بعدها ضغط النص المشفر. وهكذا ينتج لدينا الملف "CIPHER.TXT " الذي يحوي تشفير النص الواضح الموجود في الملف "PLAIN.TXT".

و الشكل الآتي (8) يظهر مسار الخوارزمية التي تم تصميمها.



الشكل (8) مسار الخوارزمية المصممة .

6- النتائج:

حصلنا من خلال هذا البحث على النتائج الآتية:

- تصميم مولد مفاتيح تشفير عشوائية بالاعتماد على قياس فعالية عمل المستخدم على لوحة المفاتيح وتم التأكد إحصائياً من عشوائية هذه التقنية، فضلاً عن زيادة أمان توليد المفتاح بالاعتماد على بعض التوابع الرياضية .

- توليد خوارزمية تشفير جديدة تعتمد على خواص الخوارزميتين IDEA و DESX من حيث المناعة ضد الهجوم بالتحليل التفاضلي والهجوم بالتحليل الخطي، فضلاً عن قوتها أمام الهجوم المباشر (و هو ما أشرنا إليه سابقاً)، مما يكسب الخوارزمية الجديدة هذه خواص.

- تطوير تطبيقين برمجين لتشفير وفك تشفير البيانات بلغة C++ بالإعتماد على الخوارزمية الجديدة ومولد المفاتيح يعملان في بيئتي دوس وويندوز وتم تسجيل نتائج هذه التطبيق والتأكد من صحتها من خلال النجاح في تشفير نص واضح ثم النجاح في استرجاع النص الواضح الأصلي من النص المشفر.

- قمت بضغط النص بعد تشفيره، مما يؤدي إلى تحديد مجموعة محارف الخرج إلى حد أدنى،بعكس الطرق الأخرى حيث لا تقوم خوارزميات تشفير العلب التقليدية بهذه العملية. فضلاً عن ذلك قمت بمعالجة البيانات المضغوطة بتابع تغيير ترتيب الخانات الذي يستخدم المفتاح كمعامل له، وهكذا بدون هذا المفتاح يصبح مستحيلاً فك ضغط البيانات المشفرة. فقد يقول البعض إنه في حال نشر الخوارزمية على العلن فإن ضغط المعطيات لن يضيف الأمان إليها لأن المتجسس سيقوم بفك ضغط النص للحصول على النص المشفر مع كل ما فيه من تكرار .و هكذا تم التغلب على هذه المشكلة. ويظهر الجدول التالي (2) مشاهدات عن تأثير الضغط في حجوم الملفات قبل التشفير وبعده.

الجدول (2) مشاهدات عند تشفير نصين (النص المصدري للتطبيق) مع وبدون ضغط باستخدام التطبيق الذي طورناه.

	نص 1	**نص 2**
حجم النص الواضح	16,947	43,843
حجم النص الواضح بعد الضغط	5,045	14,485
*** النسبة المئوية لضغط النص الواضح**	**71%**	**67%**
حجم النص المشفر	29,923	80,143
حجم النص المشفر بعد الضغط	11,833	29,000
*** النسبة المئوية لضغط النص المشفر**	**61%**	**64%**
نسبة النص المشفر والنص الواضح (قبل الضغط)	**1.76**	**1.82**
نسبة النص المشفر والنص الواضح (بعد الضغط)	**2.34**	**2.00**
* تمت عملية الضغط باستخدام البرنامج :WinZip 8.1		

7-الاستنتاجات والتوصيات :

تمكنا خلال هذا البحث من تطوير تطبيق يتغلب على الثغرات الأمنية الموجودة في التطبيقات المتداولة PGP وGnuPG من خلال الأمان في توليد المفاتيح واعتماد التطبيق على خوارزميات تشفير مصدقة ومعتمدة عالمياً:

1- أمان توليد المفاتيح: إن توليد المفاتيح بالاعتماد على قياس الفرق الزمني بين ضغطات المفاتيح ومن ثم معالجة هذا الفارق بتوابع رياضية عشوائية يعطي أماناً أفضل للمفاتيح ضد أنواع الهجوم التي تتعرض لها التطبيقات المتداولة.

2- خوارزميات التشفير:يعتبر التطبيق أكثر أماناً من التطبيق المتداول GnuPG لأن هذا الأخير يفتقد إلى الخوارزميات المعتمدة بينما اعتمد التطبيق المقترح على الخوارزمية المقترحة والتي هي دمج للخوارزميتين DESX وIDEA وهما أكثر الخوارزميات المصدقة والمسجلة تداولاً حول العالم.

وبهدف زيادة الأمان في الخوارزمية الجديدة يمكن اقتراح التوصيات الآتية:

1- زيادة طول المفتاح حتى يصبح الهجوم المباشر على الخوارزمية يحتاج إلى مدة كبيرة جداً .

2- إضافة المزيد من توابع العشوائية على تابع توليد المفاتيح لزيادة أمان توليد المفتاح.

3- إضافة مراحل تشفير وتوابع إزاحة إلى الخوارزمية الجديدة.

المراجع:

1- شناير،بروس ؛النجدي،حاتم؛ الدكاك،أميمة.التعمية التطبيقية- موافيق ورماز مصدري باللغة C ،الطبعة الثانية – الجمعية العلمية السورية للمعلوماتية سورية، 2006 .

2- A. MENEZES, P. van Oorschot and S. Vanstone. *Handbook of Applied Cryptography* by CRC Press, 1997 (780 Pages).

3- Oct 24,2007< www.answers.com>.

4 -JALLAD, K., KATZ, J., and SCHNEIER, B. "*Implementation of Chosen-Ciphertext Attacks against PGP and GnuPG"*. 2002.

5- EDWARD C. DONAHUE *Roll Your Own Crypto Services (Using Open Source and Free Cryptography)*,January 24, 2002.

6-..ZIMMERMANN, PHILIP. "*PGP User's Guide, Volume I: Essential .Topics*" Revised 11 Oct 94 for PGP version 2.6.2, 11 Oct 1994.

7- Jan 15,2007. <www.gnupg.org>.

8- RYAN THOMAS, *Using GPL Software For Email and File Encryption* Version 1.4b Option 2, May 12, 2003.

9- RYAN THOMAS ,*Attacks on PGP: A User's Perspective* GSEC Practical Assignment, Version 1.4b, 2003.

10- JEFFREY BUTTACCIO, and SAM HEALD. *The Evolution of Cryptography 12 Oct 2003*.