Improvement of Qualification and Throughput of Hybrid Systems for Electronic Mail Security with Using RC6 – Elgamal – MD5 Encryption Algorithms

Shadi Fahid Alsaghir*

(Received 25 / 8 / 2019. Accepted 21 / 7 / 2020)

□ ABSTRACT □

The Electronic Mail is considered one of the greatest applications of communications and information in this epoch, where the scientific studies proved that the people now use E-Mail application more than other communication applications because it allows them to send and receive messages with a high speed, easy to use, and save money.

Because of this importance for the E-mail we have to adopt necessary steps to protect the information transferred with these electronic messages which may be very important, one of these steps is using Encryption .

In this paper, we have performed a comparison study to evaluate the performance and efficiency of many encryption algorithms used to design a general security scheme of hybrid encryption systems which achieve a secure transfer for E-Mail messages across the internet.

This study demonstrates that RC6-Elgamal-MD5 encryption algorithms have a good performance and high efficiency which help to improve qualification and throughput of this system, so we have used these algorithms to design and program a hybrid system for E-mail security, finally we used our designed system to get the test values, and we discussed the results .

We have used Net Beans IDE 6.9.1 software to design a graphical user interface of sender and receiver in our designed system to perform the tests and get the results.

Keywords: E-Mail Security, Hybrid Encryption System, Symmetric Encryption, Asymmetric Encryption, Authentication, Encryption Throughput, Digital Signature.

* **Academic Assistant, Department of Computers and Automatic Control Engineering, Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia, Syria.**
E-mail: alsaghir.shadi@yahoo.com .

تحسين كفاءة وإنتاجية أنظمة أمن البريد الإلكتروني الهجينة باستخدام خوارزميات التشفير RC6 – Elgamal – MD5

شادي فهد الصغير*

(تاريخ الإيداع 25 / 8 / 2019. قُبِل للنشر في 21/ 7 / 2020)

□ ملخّص □

يُعتبر البريد الإلكتروني من أعظم تطبيقات ثورة الاتصالات والمعلومات التي يشهدها العصر، فقد أثبتت الدراسات العلمية أن استخدامه بدأ يطغى على استخدام وسائل الاتصال الأخرى وذلك لكونه يُعتبر وسيلة مثالية لتبادل الرسائل بسرعة كبيرة ، إضافة إلى كونه يتميز بسهولة الإستخدام وبكُلفة إقتصادية مُنخفضة.

إن هذه الأهمية للبريد الإلكتروني دعت إلى إتخاذ إجراءات ضرورية تهدف إلى تأمين المعلومات المنقولة عبر الرسائل الإلكترونية، والتي قد تكون على درجة عالية من الأهمية، ومن هذه الإجراءات استخدام التشفير Encryption.

قُمنا في هذا البحث بإجراء دراسة مقارنة تحليلية لتقييم أداء وفعالية خوارزميات التشفير المُستخدمة في تصميم المُخطط الأمني العام لأنظمة التشفير الهجينة التي تقوم بتأمين رسائل البريد الإلكتروني المُتبادلة عبر شبكة الإنترنت، حيث أظهرت الدراسة أن الخوارزميات : RC6 - Elgamal - MD5 ، تتميز بأداء جيد وفعالية عالية تُسهم في تحسين كفاءة وإنتاجية هذه الأنظمة. بعد ذلك قُمنا بتصميم وبرمجة نظام تشفير هجين لأمن البريد الإلكتروني مُستخدمين الخوارزميات الأمثل لهذا الغرض، وفي النهاية قُمنا باختبار النظام المُصمم وحصلنا على النتائج المناسبة، ثم قُمنا بمناقشة هذه النتائج.

تم إستخدام البرنامج NetBeans IDE 6.9.1 ، في تصميم وبرمجة واجهتي الإرسال والإستقبال لمُستخدمي النظام المُصمم، بهدف إجراء الإختبارات اللازمة والحصول على النتائج المناسبة.

الكلمات المفتاحية: أمن البريد الإلكتروني، نظام تشفير هجين، تشفير مُتناظر، تشفير لا مُتناظر، المُصادقة إنتاجية التشفير، التوقيع الرقمي.

* **قائم بالأعمال – قسم هندسة الحاسبات والتحكم الآلي – كلية الهندسة الميكانيكية والكهربائية – جامعة تشرين– اللاذقية – سورية.**
البريد الإلكتروني : alsaghir.shadi@yahoo.com .

مُقدمة:

تعتمد فعاليات الجنس البشري في مختلف المجالات وخاصة الشؤون المهنية على استخدام الوثائق وتبادلها مع ضرورة تأمين سريّة وتكاملية معلومات هذه الوثائق التي قد تتطلب في أغلب الأحيان نوعاً خاصاً من الحماية ضد عمليات نشرها أو تزويرها أو تخريبها، فالباب مفتوح على تفاصيل العقود بين الشركات المُتنافسة والمعلومات الأمنية والعسكرية والتجارة الإلكترونية.

أدى التغلغل الكبير لنظم المعلومات في مختلف مجالات الحياة إلى استبدال الوثائق الورقية الإعتيادية بوثائق إلكترونية يتم تبادلها عبر رسائل البريد الإلكتروني الذي أصبح يُمثّل الوسيلة الأساسية لقطاع الأعمال والإتصالات، وبالتالي يجب أن تؤدي الوثائق الإلكترونية نفس الوظائف التي كانت تقوم بها الوثائق الورقية إلا أن السمات الخاصة بالوثائق الإلكترونية جعلت تحقيق هذه الوظائف أمراً صعباً ، ومنها [1] :

1. من المُمكن عادة التمييز بين الوثيقة الورقية الأصلية وأية نسخة عنها ، إلا أنه من المُستحيل التمييز بين الوثيقة الإلكترونية وأية نسخة منها ، وذلك لأن الوثيقة الإلكترونية هي بشكل أو بآخر عبارة عن تسلسل محدد للخانات الرقمية فحسب.
2. يمكن أن يترك التغيير في الوثيقة الورقية أثراً فيزيائياً يكشف هذا التغيير أو التعديل ، أما في حالة الوثائق الإلكترونية فإن تغيير مُحتوى خانة ما أو إشارة ما ، لن يترك أي أثر فيزيائي .
3. تُبنى أيةُ عملية تحقق أو إثبات مُرتبطة بوثائق ورقية على مجموعة من الخصائص الفيزيائية لهذه الوثيقة على سبيل المثال شكل التوقيع اليدوي، أو ختم الكاتب بالعدل. أما التّحقق من أصولية الوثيقة الإلكترونية فيجب أن يُبنى على أثر داخلي موجود ضمن المعلومات.

دفعت هذه الخصائص للوثائق الإلكترونية إلى العمل على تأمين أنظمة تشفير تُؤمّن الخَدمات الأمنية الأساسية لرسائل البريد الإلكتروني المُتبادلة عبر شبكة الإنترنت ، وهذه الخدمات هي [2] :

1. السّرية : تعني حماية المعلومات المنقولة عبر الرسائل من عمليات التنصّت أو المُراقبة .
2. التكاملية : تعني منع عمليات الحذف أو الإضافة أو التعديل في معلومات الرسائل المُتبادلة .
3. عدم الإنكار: تتيح هذه الخدمة التعرف على هوية المُرسل، وبالتالي فهولا يستطيع إنكار مسؤوليته عن إرسال رسالة مُعينة.
4. المُصادقة : تتيح هذه الخدمة التّحقق من تكاملية الرسالة ، ومن هوية الجهة المُرسلة لهذه الرسالة.

أهمية البحث و أهدافه:

تكمن أهمية البحث في الحاجة المُلحّة لتوفير الخدمات الأمنية الأساسية لرسائل البريد الإلكتروني المُتبادلة عبر شبكة الإنترنت في ظل الإنتشار الواسع لإستخدامها في كافة المجالات ، أمّا الهدف من هذا البحث فهو العمل على تحسين كفاءة وإنتاجية أنظمة التشفير الهجينة من خلال تصميم وبرمجة نظام تشفير هجين Hybrid Encryption System، يُحقق الخدمات الأمنية الأساسية للرسائل المُتبادلة عبر البريد الإلكتروني باستخدام خوارزميات التشفير الأمثل لعمل هذه الأنظمة ، وذلك وفقاً لدراسة المقارنة التحليلية التي سنُجريها لتقييم أداء وفعالية خوارزميات التشفير المُستخدمة في بناء المُخطط الأمني العام لأنظمة التشفير الهجينة ، بحيث يتمتع النظام المُصمم بالخصائص التالية :

1. تحقيق مُستوى أمنٍ عالٍ للمعلومات المُتبادلة عبر رسائل البريد الإلكتروني .
2. تحقيق سرعة كبيرة في إنجاز عمليتي تشفير وفك تشفير الرسائل المُرسلة والمُستقبلة .
3. تحقيق إنتاجية عالية في مُعالجة الرسائل المُرسلة والمُستقبلة .

4. تحقيق مُتطلبات ذاكرة مُنخفضة .

5. تحقيق مُرونة عالية ، وبساطة في التصميم قدر الإمكان .

طرائق البحث و مواده:

يُمكننا تقسيم البحث إلى أربعة أجزاء رئيسة :

1. إجراء دراسة مرجعية تتضمن استعراض الخوارزميات المُستخدمة في بناء المُخطط الأمني لأنظمة التشفير الهجينة المعروفة والمُستَخدمة لضمان أمن الرسائل المُتبادلة عبر البريد الإلكتروني .

2. إجراء دراسة مقارنة تحليلية وفق عدة معايير لأهم الخوارزميات المُستخدمة في بناء المُخطط الأمني العام لأنظمة التشفير الهجينة المُبيّن في الشّكل (1) .

3. تصميم وبرمجة نظام تشفير هجين يُحقق الخدمات الأمنية الأساسية للرسائل المُتبادلة عبر البريد الإلكتروني وفقاً للأهداف الموضوعة ، وبناءً على الدراسة السابقة .

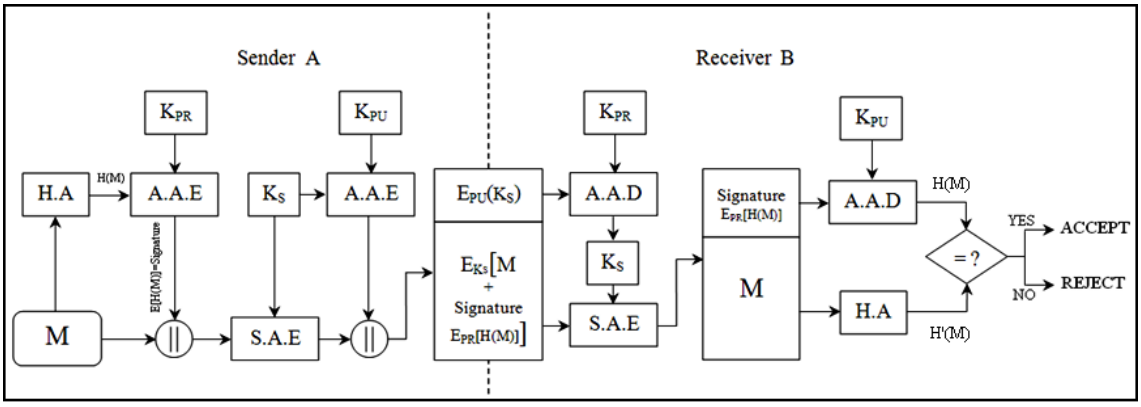
4. إختبار النظام المُصمم ، والحصول على النتائج المُناسبة ومُناقشتها .

تُصَمّم أنظمة التشفير الهجينة المُستخدمة لضمان أمن الرسائل المُتبادلة عبر البريد الإلكتروني وفقاً لعدّة مُخططات تُؤمّن خصوصية وتكاملية المعلومات المنقولة عبر هذه الرسائل، بالإضافة إلى خدمة التوقيع الرقمي. يُبنَى المُخطط الأمني لهذه الأنظمة باستخدام خوارزميات التشفير (المُتناظر منها واللامُتناظر) بالإضافة إلى خوارزميات البعثرة كما هو مُوضح في الجدول (1) [2,3,4]، حيث تتباين هذه المُخططات فيما بينها من حيث مُستوى الأمن الذي تُوفره، والكفاءة في الأداء، والبساطة في التصميم، وذلك تبعاً للخوارزميات التي تدعمها.

الجدول (1) : الخوارزميات المُستخدمة في بناء المُخطط الأمني لأنظمة التشفير الهجينة

Schemes	Encryption Algorithm	Key Management	Message Integrity	Digital Signature
PGP	DES, IDEA CAST	RSA Elgamal	MD5 SHA-1	RSA Elgamal
S/MIME	AES, 3DES RC2 .	RSA Elgamal	MD5 SHA-1	RSA Elgamal
PEM	DES 3DES	RSA	MD5 MD2	RSA
MOSS	DES 3DES	RSA	MD5	RSA

تُستَخدم خوارزمية التشفير المُتناظر DES في بناء المُخطط الأمني لأنظمة التشفير الهجينة، كما هو مُوضّح في الجدول (1)، حيث تُعالج هذه الخوارزمية كُتل مُعطيات بحجم 64 bits من النص الأصلي، وتدعم مفتاح تشفير بطول 56 bits، مما يُحقق سريّة المعلومات المُتبادلة عبر رسائل البريد الإلكتروني. إلا أنه في العام 1998 تعرضّت هذه الخوارزمية للإختراق نظراً لطول المفتاح القصير الذي تدعمه [5]، الأمر الذي حدَّ بشكل كبير من استخدامها لاعتبارها غير آمنة. مما دعا إلى استخدام خوارزميات تشفير بديلة ذات خصائص تُحقق مُتطلبات الأمن اللازمة.



الشّكل (1) : المُخطط الأمني العام لأنظمة التشفير الهجينة

يُوضح الشّكل (1) : المُخطط الأمني العام لأنظمة التشفير الهجينة [3,4] ، حيثُ أنَّ :

A.A.E : **A**symmetric **A**lgorithm **E**ncryption
A.A.D : **A**symmetric **A**lgorithm **D**ecryption.
S.A.E : **S**ymmetric **A**lgorithm **E**ncryption.
S.A.D : **S**ymmetric **A**lgorithm **D**ecryption.
H.A : **H**ash **A**lgorithm.
Ks : Secret key.
KPR : Private key.
KPU : Public key.

يتألف هذا المُخطط من المراحل التالية :

1. يُطبّق على الرسالة الإلكترونية M التي يُريد الطرف A إرسالها إلى الطرف B خوارزمية بعثرة H.A للحصول على مُلخّص الرسالة M ، أو ما يُسمّى برمز البعثرة H(M) .
2. يُشفّر رمز البعثرة H(M) ، باستخدام المفتاح الخاص K_{PR} لخوارزمية التشفير اللامُتناظر A.A.E فينتج لدينا $E_{PR}[H(M)]$ ، وهو ما يُكافئ التوقيع الرقمي Digital Signature الذي يُلحق بالرسالة M.
3. تُشفّر جُملة الرسالة M المُذيّلة بالتوقيع الرقمي باستخدام المفتاح السّري K_S لخوارزمية التشفير المُتناظر S.A.E .
4. يُشفّر المفتاح السّري K_S باستخدام المفتاح العام K_{PU} لخوارزمية التشفير اللامُتناظر A.A.E وذلك لتأمين عملية تبادل المفتاح K_S بشكل آمن بين المُرسل والمُستقبل .
5. تُرسل كتلة المُعطيات الكاملة التي تضم جُملة الرسالة المُذيّلة بالتوقيع الرقمي والمُشفّرة باستخدام المفتاح السّري K_S ، بالإضافة إلى المفتاح السّري المُشفّر باستخدام المفتاح العام $E_{PU}(K_S)$.
6. يقوم المُستقبل B باستقبال كتلة المُعطيات الكاملة المُشفّرة ، ثم يبدأ بفك تشفير المفتاح السّري K_S ، وذلك باستخدام المفتاح الخاص K_{PR} لخوارزمية فك التشفير اللامُتناظر A.A.D .
7. بعد استخلاص المفتاح السّري K_S ، يتم فك تشفير جُملة الرسالة M المُذيّلة بالتوقيع الرقمي وذلك بتطبيق خوارزمية فك التشفير المُتناظر S.A.D باستخدام هذا المفتاح .
8. تُطبّق خوارزمية فك التشفير اللامُتناظر A.A.D على التوقيع الرقمي، لفك تشفيره باستخدام المفتاح العام لهذه الخوارزمية K_{PU} ، فنحصل على رمز البعثرة للرسالة H(M) .
9. يُطبّق على الرسالة المُستقبلة نفس خوارزمية البعثرة H.A المُطبّقة على الرسالة المُرسلة ، فينتج لدينا رمز بعثرة H'(M) ، يتم بعد ذلك مُقارنة H'(M) مع H (M) ، وهنا نكون أمام إحتمالين:

a) H(M) = H'(M) : في هذه الحالة يُمكن للمُستقبل B أن يضمن تكاملية الرسالة، أي أنه استقبل هذه الرسالة بحالتها الأصلية الصحيحة دون أي يطرأ عليها أي تعديل أثناء النقل ، كما يضمن بأن الطرف A هو الطرف المُرسل

لهذه الرسالة ، حيث أن الطرف A لا يستطيع إنكار مسؤوليته عن إرسال هذه الرسالة كونه الطرف الوحيد الذي يملك المفتاح الخاص K_{PR} الذي استُخدم في إنجاز التوقيع الرقمي ، وبالنتيجة هذه الرسالة مُصادقة ويتم قبولها .

b) $H(M) \neq H'(M)$: في هذه الحالة : إمّا أن يكون الطرف المُرسل للرسالة منتحلاً شخصية أخرى ، لعدم امتلاكه المفتاح الخاص المُناسب . أو أن يكون قد حدث تعديل على معلومات الرسالة أثناء نقلها بطريقة ما وفي كلتا الحالتين تُرفض الرسالة لاعتبارها غير مُصادقة .

نستنتج أنّ المُخطط السابق يعمل على تأمين خدمات السريّة والتكاملية وعدم الإنكار والمُصادقة للرسائل المُتبادلة عبر البريد الإلكتروني بالإضافة إلى خدمة تبادل المفتاح السّري بشكل آمن بين المُرسل والمُستقبل، وذلك من خلال استخدام عدة أنواع من الخوارزميات تعمل بالتكامل مع بعضها البعض لتوفير الخدمات الأمنية الأساسية لرسائل البريد الإلكتروني المُتبادلة عبر الشبكة ، ويمكن تصنيف هذه الخوارزميات إلى:

1- خوارزميات التشفير Encryption Algorithms [6] : وتشمل:

- خوارزميات التشفير المُتناظر Symmetric Key Encryption Algorithms :

يتم في هذا النمط من الخوارزميات استخدام مفتاح سرّي واحد Secret Key ، لإنجاز عملية تشفير البيانات في طرف الإرسال ومن ثم فك تشفيرها في طرف الاستقبال ، وتجدر الإشارة إلى أنه يجب أن يبقى هذا المفتاح مُحتفظاً بسريّته من قبل المُرسل والمُستقبل فقط . تُقدّم هذه الخوارزميات خدمة السريّة وتتميز بسرعة أدائها في إنجاز عمليتي تشفير وفك تشفير البيانات ، إلا أنها تعاني من مشكلة التوزيع غير الآمن للمفتاح السّري بين المُرسل والمُستقبل . وأشهر خوارزميات هذا النمط :

. AES [10] –RC6 [6] – Towfish [9] – MARS [8] – Serpent [7]

- خوارزميات التشفير اللامُتناظر Asymmetric Key Encryption Algorithms:

يُستخدم في هذا النمط من الخوارزميات مفتاحين مختلفين تربط بينهما علاقة لإنجاز عمليتي تشفير و فك تشفير البيانات وهما : المفتاح العام Public Key ، والمفتاح الخاص Private Key . يكون المفتاح الخاص معروفاً أو خاصاً بطرف واحد فقط من الأطراف الشّرعيّة المُتصلة عبر الشبكة ، ويُستخدم لإنجاز خدمة التوقيع الرقمي . أمّا المفتاح العام فيكون معروفاً ومعمماً لدى جميع الأطراف الشّرعيّة على الشبكة ويُستخدم لفك تشفير البيانات التي شفّرها المفتاح الخاص المُرتبط به فقط (التّحقق من التوقيع الرقمي) ، كما يُمكن استخدامه لتشفير البيانات ، إذ أن مالك المفتاح الخاص المُرتبط بمفتاح عام معين ، هو الوحيد الذي يستطيع فك تشفير البيانات التي شفّرها هذا المفتاح العام . تُعتبر هذه الخوارزميات حلاً لمُشكلة التوزيع غير الآمن للمفاتيح في خوارزميات التشفير المُتناظر إلا أنها تُعاني من مُشكلة البُطئ في الأداء حيث تجدر الإشارة إلى أن سرعة تنفيذ هذه الخوارزميات أقل بحوالي (100−1000) مرّة من سرعة تنفيذ خوارزميات التشفير المُتناظر في مُعالجة المُعطيات [3]. وأشهر خوارزميات هذا النمط:Elgamal [12]–RSA [11]

2- خوارزميات البعثرة Hash Algorithms [1] :

هي توابع تقبل رسائل ذات طول متغير M bits كدخل ، وتُنتج خرج ذو طول ثابت، يُشار إلى الخرج الناتج بمُلخّص الرسالة M أو ما يسمى برمز البعثرة H(M) . إن رمز البعثرة لا يستخدم أي مفتاح بل هو تابع لرسالة الدخل فقط ويُزود قابلية كشف الخطأ: حيث أن أي تغيير في أي بت من بتات الرسالة، يُنتج رمز بعثرة مُختلف. وأشهر هذه الخوارزميات هي: [4] SHA-1 – [13] MD5 .

النتائج والمناقشة:

1- دراسة خوارزميات التشفير المُتناظر:

سنقوم بدراسة الخصائص المُميزة لخوارزميات التشفير المُتناظر ، وفق المعايير التالية [6,7] :

1- مُرونة الخوارزمية : تعني إمكانية تطبيقها بشكل آمن و فعّال على مجال واسع من التطبيقات البرمجية.

2- البساطة : هي أن تكون الخوارزمية سهلة التصميم ، ويُعتبر حجم البرنامج الكلي مقدراً بعدد أسطر الكود أحد المعايير المُتبعة لتحديد درجة تعقيد الخوارزمية ، وهو المعيار الذي اعتمدناه في هذا البحث .

3- سرعة التشفير/فك التشفير : ترتبط سرعة الخوارزمية في إنجاز عمليتي التشفير وفك التشفير للبيانات بإنتاجيتها Throughput والتي يُمكن تعريفها بأنها: كمية البيانات التي يُمكن مُعالجتها (تشفيرها - فك تشفيرها) في وحدة الزمن.

4- مُستوى الأمن : يرتبط بعاملين هما : بنية الخوارزمية، و طول المفتاح المُستخدم في الخوارزمية .

5- مُتطلبات الذاكرة : تأخذ بالحُسبان حجم الكود، و كذلك مُتطلبات الذاكرة RAM.

1-1 دراسة خوارزميات التشفير المُتناظر من حيث سرعة التشفير/فك التشفير :

تختلف إنتاجية خوارزميات التشفير المُتناظر عند إنجاز تشفير وفك تشفير البيانات تبعاً للخوارزمية المُستخدمة و طول المفتاح المُستخدم أيضاً ، كما هو مُبين في الجدول (2) ، حيث يُشير الحقل الأول إلى الخوارزمية المدروسة في حين يُشير الحقل الثاني إلى إنتاجية كل خوارزمية مُقدّرة بواحدة (Kbits/sec) عند إنجاز التشفير باستخدام مفتاح بطول 128 bits ، ويُبين الحقل الثالث إنتاجية هذه الخوارزميات عند استخدام مفتاح بطول192 bits ، أما الحقل الرابع فيُشير إلى الإنتاجية عندما طول المفتاح 256 bits . في حين يُشير الحقل الخامس و السادس و السابع إلى إنتاجية هذه الخوارزميات مُقدرة بواحدة (Kbits/sec) عند إنجاز فك التشفير بأطوال مفاتيح 128 -192-256 bits على التوالي [6,7].

الجدول (2) : إنتاجية خوارزميات التشفير المُتناظر عند مُعالجة البيانات بأطوال مفاتيح مختلفة

	Throughput (Kbits/sec)					
Algorithm	Encrypt 128 bits	Encrypt 192 bits	Encrypt 256 bits	Decrypt 128 bits	Decrypt 192 bits	Decrypt 256 bits
RC6	4698	4740	4733	4733	4698	4740
AES	4855	4664	4481	4819	4624	4444
MARS	3738	3707	3733	3965	3965	3936
Serpent	1843	1855	1861	1873	1897	1896
Twofish	1749	1749	1744	1781	1775	1781

تُوضّح النتائج المُبيّنة في الجدول (2) ما يلي:

تتمتع الخوارزمية RC6 بإنتاجية عالية تفوق إنتاجية باقي الخوارزميات عند إنجاز عمليتي التشفير وفك التشفير باستخدام مفاتيح بأطوال192 -256 bits . فعند التشفير باستخدام مفتاح بطول 256 bits تكون إنتاجية هذه الخوارزمية أكبر بنسبة 6% من إنتاجية الخوارزمية AES ، وأكبر بنسبة 27% من إنتاجية الخوارزمية MARS ، كما أنها تكون أكبر بنسبة 154% من إنتاجية الخوارزمية Serpent، وأكبر بنسبة 171% من إنتاجية الخوارزمية Twofish .

وعند التشفير باستخدام مفتاح بطول 192 bits تكون إنتاجية الخوارزمية RC6 أكبر بنسبة 2% من إنتاجية الخوارزمية AES ، و أكبر بنسبة 28% من إنتاجية الخوارزمية MARS ، كما أنها تكون أكبر بنسبة 156% من إنتاجية

الخوارزمية Serpent، وأكبر بنسبة 171% من إنتاجية الخوارزمية Twofish . بينما تَفوق إنتاجية الخوارزمية AES إنتاجية باقي الخوارزميات عند التشفير باستخدام مفتاح بطول 128 bits، حيث تكون إنتاجية هذه الخوارزمية أكبر بنسبة 3% من إنتاجية الخوارزمية RC6 عند هذا الطول للمفتاح.

وبالنسبة لعملية فك التشفير وعند إستخدام مفتاح بطول 256 bits ، تكون إنتاجية الخوارزمية RC6 أكبر بنسبة 7% من إنتاجية الخوارزمية AES ، و أكبر بنسبة 20% من إنتاجية الخوارزمية MARS ، كما أنها تكون أكبر بنسبة 150% من إنتاجية الخوارزمية Serpent ، وأكبر بنسبة 166% من إنتاجية الخوارزمية Twofish وعند فك التشفير باستخدام مفتاح بطول 192 bits ، تكون إنتاجية الخوارزمية RC6 أكبر بنسبة 2% من إنتاجية الخوارزمية AES ، وأكبر بنسبة 18% من إنتاجية الخوارزمية MARS ، كما أنها تكون أكبر بنسبة 148% من إنتاجية الخوارزمية Serpent ، وأكبر بنسبة 165% من إنتاجية الخوارزمية Twofish . بينما تَفوق إنتاجية الخوارزمية AES إنتاجية باقي الخوارزميات عند فك التشفير باستخدام مفتاح بطول 128 bits حيث تكون إنتاجية هذه الخوارزمية أكبر بنسبة 2% من إنتاجية الخوارزمية RC6 عند هذا الطول للمفتاح.

2-1 دراسة خوارزميات التشفير المُتناظر من حيث مُستوى الأمن :

يرتبط عامل مُستوى أمن الخوارزمية بعاملين اثنين هما [4] :

1. بنية الخوارزمية المُستخدمة .
2. طول المفتاح المُستخدم عند إنجاز عمليتي التشفير و فك التشفير للبيانات .

يختلف تصميم البنية الداخلية لخوارزميات التشفير المُتناظر من حيث : طول المفتاح المُستخدم ، وحجم كتلة المُعطيات المُعالَجة ، وبنية التشفير ، وعدد الدورات اللازمة لمُعالجة المُعطيات ، والعمليات الحسابية والمنطقية المُستخدمة فيها ، حيث يُبيّن الجدول (3)، والجدول (4)، الإختلاف بين خوارزميات التشفير المُتناظر المدروسة وفق هذه المعايير [2,3,6].

الجدول (3) : الإختلاف بين خوارزميات التشفير المُتناظر من حيث طول المفتاح و حجم كتلة المُعطيات وبنية التشفير و عدد الدورات

Algorithm	Key Length (bits)	Block Size (bits)	Cipher Structures	Number of Round
AES	128 -192- 256	128	Substitution Permutation (SPN)	10 – 12 – 14
RC6	128 -192- 256	128	Modified Feistel (MFN)	20
Twofish	128 -192- 256	128	Feistel Network (FN)	16
MARS	128 -192- 256	128	Modified Feistel (MFN)	32
Serpent	128 -192- 256	128	Substitution Permutation (SPN)	32

الجدول (4): العمليات الحسابية و المنطقية المستخدمة في كل خوارزمية

Algorithm	العمليات الحسابية و المنطقية								
	⊕	⊞	⊟	⊙	>>>	<<<	S-box	$\log_x$	exp.
RC6	✓	✓		✓	✓	✓		✓	
AES	✓			✓	✓		✓		
Twofish	✓	✓					✓		
Mars	✓	✓	✓	✓	✓	✓	✓		
Serpent	✓						✓		

يتبيّن لنا من هذه الجداول ما يلي:

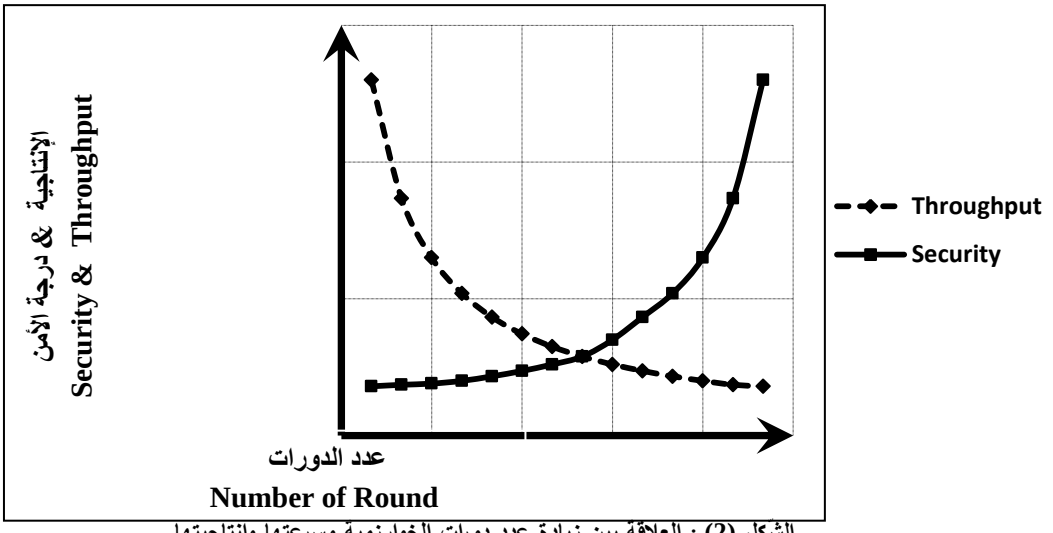
1- تُوفر الخوارزميات : RC6, MARS, Twofish, AES, Serpent ، سويّات أمنية مُختلفة للمعلومات بحسب طول المفتاح المُستخدم فيها ، وذلك من خلال دعمها لمفاتيح بأطوال 256 – 192 – 128 bits .

2- تُعالج الخوارزميات: Serpent, AES, Twofish, MARS, RC6، كتل مُعطيات بحجم 128 bits من النص الأصلي.

3- تتم مُعالجة المُعطيات في الخوارزميتين Serpent , AES ، عبر شبكة الإستعاضة والإستبدال SPN ، في حين تُعالَج المُعطيات في الخوارزمية Twofish بواسطة شبكة فيستيل FN ، بينما تتم مُعالجة المُعطيات في الخوارزميتين RC6 و MARS بواسطة شبكة فيستيل المعدلة MFN [3,4].

4- تختلف خوارزميات التشفير المُتناظر من حيث عدد الدورات اللازمة لمُعالجة المُعطيات، حيث يُمكننا مُلاحظة أن الخوارزمية AES يُمكن أن تُطبّق 10 دورات أو 12 دورة أو 14 دورة ، للحصول على كُتلة مُعطيات النص المُشفّر. أما الخوارزمية RC6 فتُطبّق 20 دورة ، في حين تُطبّق الخوارزميتين MARS و Serpent 32 دورة للحصول على كتلة مُعطيات النص المُشفّر ، و 16 دورة بالنسبة للخوارزمية Towfish .

مما سبق يمكننا القول أن زيادة طول المفتاح المُستخدم تؤدي إلى رفع عامل مُستوى أمن خوارزمية التشفير المُتناظر، كما تؤدي زيادة عدد دوراتها إلى رفع هذا العامل أيضاً . ولكن بالمُقابل فإن زيادة عدد الدورات تؤدي إلى إنقاص سرعة الخوارزمية في مُعالجة المُعطيات (تشفير – فك تشفير) ، وبالتالي ستنخفض إنتاجيتها ، حيث تُبيّن المُنحنيات المُوضّحة في الشّكل(2)، وجود علاقة عكسية بين زيادة عدد دورات الخوارزمية من جهة وبين سرعتها وبالتالي إنتاجيتها من جهة أخرى.



الشكل (2) : العلاقة بين زيادة عدد دورات الخوارزمية وسرعتها وإنتاجيتها

يُمكننا تعريف تابع الأمن Security Function ، كما يلي [4] :

$$Security\ \ Function = \frac{(Number\ of\ Rounds)\times(Number\ Of\ Key\ Bytes)}{Encryption\ Throughput} \qquad (1)$$

يُعطي هذا التابع قيماً لها واحدة الزمن ، وبالتالي وباعتبار أن الإنتاجية تقدر بـ (Byte/sec) فإنه يُمكننا تعريف عامل الأمن Security Factor ، بالعلاقة التالية:

$$Security\ \ Factor = \frac{Security\ \ Function}{Encryption\ Time} \qquad (2)$$

في الواقع ليس هناك صيغة دقيقة لتحديد عدد الدورات المطلوبة لضمان تشفير آمن ، وهذا عائد إلى الطبيعة الداخلية لكل خوارزمية تشفير ، وعلى أية حال فقد اقترح Knudsen في العام 2000 [3,4] ، صيغة تجريبية لتحديد ذلك العدد ، حيث اقترح Knudsen صيغة رياضية لتحديد العدد الأصغري من الدورات اللازمة لتحقيق مُستوى الأمن المطلوب لخوارزمية التشفير، كما يلي:

$$r \geq d.n/w \qquad (3)$$

حيث أنَّ :

r : العدد الأصغري من الدورات اللازمة لتحقيق مُستوى الأمن المطلوب للخوارزمية.

w : الحجم الأصغري للكلمة المُستخدمة كدخل في أي مرحلة مزج ضمن بنية الخوارزمية .

d : العدد الأعظمي للمراحل التي تعالج كلمة واحدة بطول w bits .

n : حجم كتلة المُعطيات المُعالَجة .

وحسب Knudsen، تمكنّا من تحديد العدد الأصغري من الدورات اللازمة لتحقيق مُستوى الأمن المطلوب لكل خوارزمية من الخوارزميات المدروسة ، كما هو مُبيّن بالجدول (5) :

الجدول (5) : عدد الدورات الأصغري اللازم لتحقيق مُستوى الأمن المطلوب للخوارزمية

Algorithm	d	n	w	r
AES	4	128	8	16
RC6	4	128	32	16
MARS	4	128	32	16
Serpent	4	128	32	16
Towfish	2	128	64	4

نستنتج مما سبق أن الخوارزميات Serpent, AES, Twofish, MARS, RC6 توفر سويّات أمنية مُختلفة للمعلومات تبعاً لطول المفتاح المُستخدم فيها، كما أنها تُحقق العدد الأصغري من الدورات اللازمة لتحقيق مُستوى الأمن المطلوب.

3-1 دراسة خوارزميات التشفير المُتناظر من حيث مُتطلبات الذاكرة:

تُعتبر مُتطلبات الذاكرة مؤشراً هاماً يدل على ميزات الخوارزمية ، ويُقصد بها سعة الذاكرة اللازمة أثناء تشغيل الخوارزمية لاستيعاب الكود اللازم للتشغيل ، وحسب الجدول (6) ، يُمكننا مُلاحظة مُتطلبات الذاكرة التي تحتاجها كل خوارزمية من الخوارزميات المدروسة [6,7] :

الجدول (6) : يُبيّن مُتطلبات الذاكرة للخوارزميات المُتناظرة

Algorithm	Code Size (Bytes)	Heap Usage (Bytes)
RC6	7077	432
AES	12158	18360
Twofish	17189	7600
MARS	18110	4808
Serpent	39290	4680

تُبيّن هذه النتائج حجم الكود ومُتطلبات الذاكرة المُنخفضة جداً للخوارزمية RC6 مُقارنة مع باقي الخوارزميات حيث لا يتجاوز حجم الذاكرة الذي تحتاجه الخوارزمية RC6 نسبة 3% من حجم الذاكرة الذي تحتاجه الخوارزمية AES ، كما أنه لا يتجاوز نسبة 6% من حجم الذاكرة الذي تحتاجه الخوارزمية Twofish ونسبة 9% من حجم الذاكرة الذي تحتاجه كل من الخوارزميتين MARS و Serpent .

4-1 مُقارنة خصائص خوارزميات التشفير المُتناظر:

بعد مُراجعة الخصائص المُميزة لخوارزميات التشفير المُتناظر المدروسة ، تبيّن لنا ما يلي:

1. تتميز الخوارزمية RC6 ، بسرعة كبيرة وإنتاجية عالية في إنجاز عمليتي تشفير و فك تشفير البيانات مُقارنة مع باقي الخوارزميات .

2. تتمتع الخوارزمية RC6 بمُستوى أمن جيد ، من خلال استخدام مفتاح بطول 256 bits ، وعدد دورات يساوي 20 دورة ، لإعطاء كتلة مُعطيات مُشفرّة بطول 128 bits من النص الأصلي، وهي حسب Knudsen تُحقق العدد الأصغري من الدورات اللازمة لتحقيق مُستوى الأمن المطلوب.

3. تتمتع الخوارزمية RC6 بمُتطلبات ذاكرة مُنخفضة ، الأمر الذي يُمّيز هذه الخوارزمية بالبساطة في التصميم والمُرونة في التطبيق ، حيث يُمكن تطبيقها على مجال واسع من التطبيقات المختلفة .

إن هذه الميزات التي تتمتع بها الخوارزمية RC6 ، دعتنا إلى استخدامها في تصميم نظام أمن البريد الإلكتروني الهجين المُقترح ، كونها تُلبي تطلعاتنا في تحقيق الغاية المطلوبة من التصميم .

2- دراسة خوارزميات البعثرة Hash Algorithms :

تُعالج هذه الخوارزميات الرسائل ذات الطول العشوائي على شكل كُتل من المُعطيات لتُنتج بالنهاية رمز بعثرة H(M) ، ذو طول ثابت يختلف باختلاف الخوارزمية المُستخدمة . يحصل التصادم Collision [3] ، في هذا الصنف من الخوارزميات عندما تُنتج خوارزمية البعثرة نفس رمز البعثرة من أجل رسالتين مختلفتين ، أي :

$$H(M_1) = H(M_2) \qquad \Rightarrow \text{Collision} : M_1 \neq M_2$$

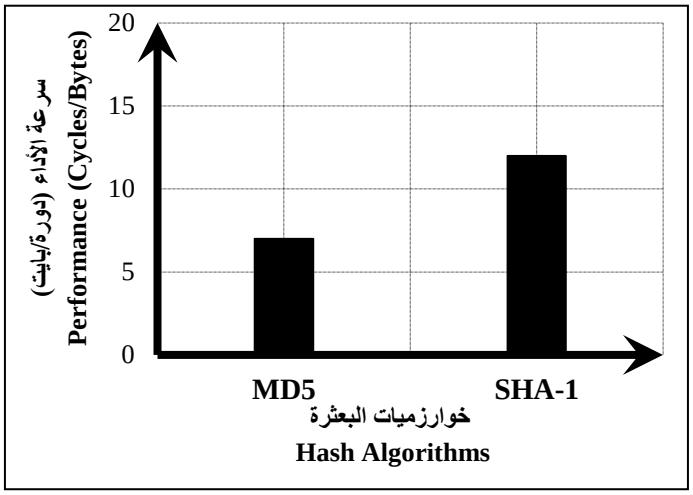
تُستخدم هذه الخوارزميات لتحقيق مُصادقة الرسائل المُتبادلة عبر البريد الإلكتروني من خلال إنجاز التوقيع الرقمي وأشهر خوارزميات هذا النمط : خوارزمية MD5 ، وخوارزمية SHA-1 .

تم دراسة خوارزميات البعثرة حيث يوضح الجدول (7) ، الإختلاف بين خوارزميتي البعثرة MD5 و SHA-1 من حيث حجم كتلة المُعطيات المُعالَجة ، وطول رمز البعثرة الناتج على الخرج ، وعدد دورات الخوارزمية وقيم الخوارزمية الإبتدائية ، ودرجة تعقيد التصادم [2,3] :

الجدول (7) : الإختلاف بين خوارزميتي البعثرة MD5 و SHA-1

الدّالة	Function	MD5	SHA-1
طول الكتلة	Block Length	512 bits	512 bits
طول الخرج	Output Length	128 bits	160 bits
عدد الدورات	Rotation Steps	64 Steps	80 Steps
القيم الإبتدائية	Initialization Variables	4	5
درجة تعقيد التصادم	Collision Complexity	2^{64}	2^{80}

كما يُبيّن الشّكل (3)، سرعة أداء الخوارزميتينMD5 و SHA-1، في مُعالجة المُعطيات مُقدّرة بواحدة (Cycles/Bytes) [14]:

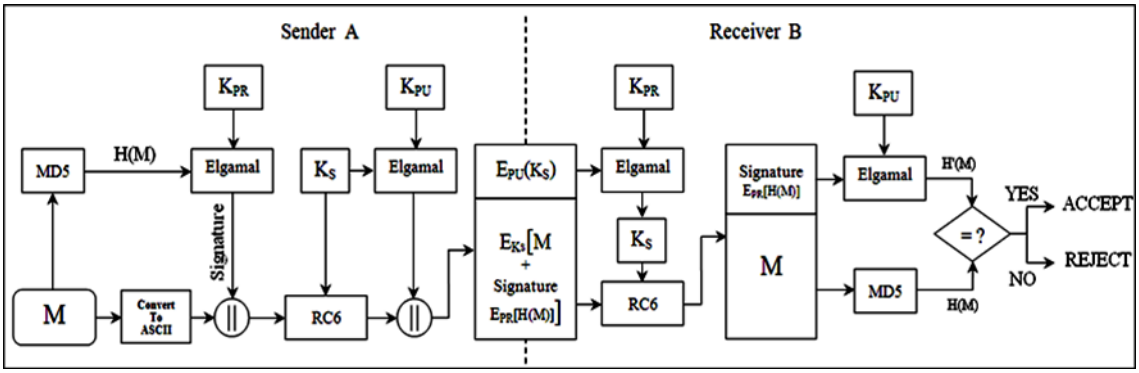


الشّكل (3) : مُقارنة سرعة أداء الخوارزميتين MD5 و SHA-1 في مُعالجة المُعطيات

نستنتج من الدراسة السابقة أن الخوارزمية MD5 تتمتع بأداء جيد بالمُقارنة مع الخوارزمية SHA-1 ، فهي تتميز بسرعة عالية في مُعالجة المُعطيات إضافةً إلى مُقاومتها العالية ضد التصادم ، الأمر الذي دعانا إلى استخدامها في تصميم نظامنا المُقترح.

3- تصميم نظام أمن البريد الإلكتروني الهجين المُقترح:

بنتيجة الدراسة المقارنة التحليلية السابقة التي قُمنا بها ، تمكنّا من وضع المُخطط الأمني لنظام تشفير هجين لأمن البريد الإلكتروني كما هو مُبيّن في الشّكل (4) ، حيث اعتمدنا في تصميم هذا المُخطط على استخدام الخوارزميات التالية : RC6, Elgamal, MD5 نظراً لأن خصائصها المُميزة تُلبي تطلعاتنا في تحقيق الأهداف المرجوة من هذا البحث.



الشّكل (4) : المُخطط الأمني لنظام أمن البريد الإلكتروني الهجين المُصمم

1-3 خوارزمية التشفير المُتناظر RC6 (Rivest Cipher, Version 6) [2,3] :

صُمّمت هذه الخوارزمية في نظامنا المُقترح بمفتاح سري K_S بطول 256 bits ، وعدد دورات r = 20 دورة وهي حسب Knudsen ، تحقق مُستوى الأمن المطلوب. تُستخدم هذه الخوارزمية لتشفير البيانات ، حيث تُبدي مُمانعة عالية ضد عمليات الإختراق المُحتملة عند هذه القيم للمفتاح وعدد الدورات ، الأمر الذي يضمن سريّة وخُصوصية الرسائل المُتبادلة عبر البريد الإلكتروني ، ويُلبّي تطلعاتنا في التصميم .

تتم عملية التشفير باستخدام الخوارزمية RC6-w/r/b ، وفق الآلية التالية:

الدخل : النص الأصلي المخزن في أربعة مسجلات دخل بطول w بت ، (A , B , C , D) .

عدد الدورات : r دورة .

مفاتيح دورة بطول w bits : S [0 , 1 , 2 , 3..... , 2r + 3] .

الخرج : النص المُشفّر مخزن في المسجلات A , B , C , D .

الإجرائية :

Procedure : $B = B + S[0]$, $D = D + S[1]$

for i = 1 to r do **{**

$t = (B \times (2B + 1)) <<< 1g\ w$

$u = (D \times (2D + 1)) <<< 1g\ w$

$A = ((A \oplus t) <<< u) + S[2i]$

$C = ((C \oplus u) <<< t) + S[2i + 1]$

$(A, B, C, D) = (B, C, D, A)$ **}**

$A = A + S[2r + 2]$, $C = C + S[2r + 3]$

كما تتم عملية فك التشفير باستخدام الخوارزمية RC6-w/r/b ، وفق الآلية التالية:

الدخل : النص المُشفّر المخزن في أربعة مسجلات دخل بطول w بت وهي (A , B , C , D) .

عدد الدورات : r دورة .

مفاتيح دورة بطول w bits : S [0,1,2,3....., 2r + 3] .

الخرج : النص الأصلي مخزن في المسجلات A , B , C , D .

الإجرائية :

Procedure : $C = C - S[2r + 3]$, $A = A - S[2r + 2]$

for i = r down to 1 do **{**

$(A, B, C, D) = (D, A, B, C)$

$u = (D \times (2D + 1)) <<< 1g\ w$

$t = (B \times (2B + 1)) <<< 1g\ w$

$C = ((C - S[2i + 1]) >>> t) \oplus u$

$A = ((A - S[2i]) >>> u) \oplus t$ **}**

$D = D - S[1]$, $B = B - S[0]$

2-3 خوارزمية التشفير اللامُتناظر Elgamal [3]:

بالرغم من الإستخدام الواسع لخوارزمية التشفير اللامُتناظر RSA في إنجاز خدمات التوقيع الرقمي و تبادل المفتاح السّري بشكل آمن ، إلا أن الخوارزمية Elgamal تُعتبر أكثر أماناً منها، كما أنها أكثر سُرعة وفعالية في مُعالجة المُعطيات بالمُقارنة مع الخوارزمية RSA [12] ، الأمر الذي دعانا إلى اختيارها في تصميم نظامنا المُقترح .

يعتمد المُخطط الأمني لخوارزمية Elgamal على صُعوبة حساب اللوغاريتم المُتقطع في حقول غاليوس GF(p) [4]، حيث صُمّمت هذه الخوارزمية في نظامنا المُقترح بمفتاح طوله 1024 bits. تُستخدم هذه الخوارزمية من أجل إنجاز خدمة التوقيع الرقمي ، إضافة إلى تأمين عملية تبادل المفتاح السّري K_S للخوارزمية RC6 بشكل آمن بين المُرسل والمُستقبل (Key Management) .

3-3 خوارزمية البعثرة MD5 (Message Digest, Version 5) [13]:

تُستخدم هذه الخوارزمية في نظامنا المُقترح من أجل إنجاز خدمة التوقيع الرقمي وذلك بالمشاركة مع خوارزمية Elgamal ، حيث تُنتج هذه الخوارزمية مُلخّص ثابت للرسالة مهما كان حجمها ويُعبّر عنه برمز البعثرة H(M) =128 bits ، كما تُستخدم أيضاً من أجل التّحقق من مُصادقة الرسائل المُستقبلة.

4- اختبار نظام أمن البريد الإلكتروني الهجين المُصمم :

استخدمنا برنامج NetBeans IDE 6.9.1 في تصميم واجهتي الإرسال والإستقبال لنظام أمن البريد الإلكتروني الهجين المُقترح ، حيث اختبرنا إرسال وإستقبال العديد من الرسائل ذات الحُجوم المُختلفة وبناءً عليه تمكنّا من تحديد الزمن المُستغرق عند تنفيذ كل مرحلة من مراحل النظام وذلك عند إنجاز تشفير الرسائل في طرف الإرسال وفك تشفيرها في طرف الإستقبال ، كما تَمكنّا من تحديد إنتاجية النظام المُصمم .

1-4 اختبار النظام المُصمم في إنجاز تشفير الرسائل :

استخدمنا البرنامج المُصمم في إنجاز تشفير رسائل ذات حُجوم مختلفة ، وحصلنا على النتائج المُبينة في الجدول (8)، حيث يُبيّن هذا الجدول الزمن المُستغرق (بالميللي ثانية) في كل مرحلة من مراحل النظام المُصمم عند تشفير رسائل ذات حُجوم مُختلفة . حيث يُعبّر الحقل الأول عن حجم الرسالة المراد إرسالها مُقدّرة بالكيلو بايت ، في حين يُعبّر الحقل الثاني عن الزمن اللازم لتحويل الرسالة إلى ترميز الآسكي المُوافق ، والحقل الثالث يُعبّر عن الزمن اللازم لإنجاز التوقيع الرقمي باستخدام خوارزميتي Elgamal-MD5 ، أما الحقل الرابع فيُعبّر عن الزمن اللازم لإنجاز تشفير جُملة الرسالة المُذيلة بالتوقيع الرقمي باستخدام الخوارزمية RC6 ، ويُعبّر الحقل الخامس عن الزمن اللازم لتشفير المفتاح السّري للخوارزمية RC6 ، أما الحقل السادس فيعبّر عن الزمن الكلي لإنتاج كُتلة المُعطيات المُشفّرة، ويُعبّر الحقل السابع عن إنتاجية تشفير النظام المُصمم مُقدّرة بواحدة (Byte/sec) .

تُبين النتائج الواردة في الجدول (8) ما يلي :

1. نلاحظ تزايد زمن تحويل الرسالة إلى ترميز الآسكي المُوافق لها بشكل كبير مع تجاوز حجم الرسائل القيمة 40 KB، حيث يزداد هذا الزمن بنسبة تتراوح بين %(47-21) ، مع زيادة حجم الرسائل من (40→80 KB) .

الجدول (8) : الزمن المُستغرق في كل مرحلة من مراحل النظام المُصمم عند إنجاز تشفير رسائل ذات حُجوم مُختلفة

Message Size (KB)	Convert to ASCII (msec)	Signature Time (msec)	(Message and Signature) Encryption Time (msec)	Secret Key Encryption Time (msec)	Total Time (msec)	Encryption Throughput (Byte/Sec)
0.5	16	16	562	31	625	800
1	16	31	797	31	875	1142.86
1.5	16	15	1016	31	1078	1391.47
2	16	16	1250	31	1313	1523.23
2.5	16	16	1484	31	1547	1616.1
3	16	16	1718	47	1797	1669.45
3.5	32	46	1922	32	2032	1722.44
4	31	15	2156	32	2234	1790.51

4.5	32	32	2406	47	2517	1787.84
5	31	32	2610	47	2720	1838.24
6	47	16	3078	47	3188	1882.1
7	62	15	3531	47	3655	1915.18
8	78	16	3984	62	4140	1932.37
9	94	30	4469	63	4656	1932.99
10	109	32	4922	63	5126	1950.84
12	172	16	5844	62	6094	1969.15
14	218	31	6766	94	7109	1969.33
16	281	15	7735	94	8125	1969.23
19	437	16	9219	125	9797	1939.37
22	625	32	10656	157	11470	1918.1
26	953	31	12625	187	13796	1884.6
30	1375	31	14610	234	16250	1846.15
34	2032	31	16625	297	18985	1790.89
38	2828	47	18687	344	21906	1734.68
44	3531	16	21484	438	25469	1727.6
50	5109	32	24547	547	30235	1653.71
56	6187	32	27407	671	34297	1632.8
63	8750	32	30812	828	40422	1558.58
70	12219	30	34531	1015	47795	1464.59
80	18000	32	39782	1281	59095	1353.75

2. عملياً ووفقاً للقيم التجريبية، نجد أن زمن إضافة التوقيع الرقمي للرسالة النّصية مهما اختلف حجمها هو زمن مُنخفض لا تتجاوز قيمته نسبة 4% من الزمن الكلّي اللازم لتوليد كتلة المُعطيات المُشفرة الكُليّة ، وهذه في الحقيقة نتيجة منطقية نظراً لأن التوقيع الرقمي يُنجز باستخدام رمز البعثرة للرسالة ذو الطول الثابت 128 bits.

كما يُمكننا مُلاحظة أنه من أجل الرسائل ذات الحُجوم 3.5 Kbyte و 38 Kbyte يكون الزمن اللازم لإضافة التوقيع الرقمي لهذه الرسائل هو 46 msec و 47 msec على التوالي ، وهو في الحقيقة زمن مُرتفع بالمُقارنة مع زمن إضافة التوقيع الرقمي لحُجوم الرسائل الأخرى ، ويعود السبب في ذلك إلى اختلاف الزمن اللازم لتوليد زوج (المفتاح العام – المفتاح الخاص) (Key generation) للخوارزمية Elgamal من أجل كل رسالة يتم إرسالها، حيث تتراوح قيمة هذا الزمن بين (34 msec -5) ، ويُضاف إلى الزمن اللازم لإنجاز التوقيع الرقمي.

3. يزداد الزمن اللازم لإنجاز تشفير جُملة الرسالة المُذيلة بالتوقيع الرقمي باستخدام الخوارزمية RC6 بنسبة تتراوح بين %(12-27) مع تزايد حجم الرسالة ، حيث يُشكّل هذا الزمن نسبة تُقدّر بـ 68% من الزمن الكلّي اللازم لتوليد كُتلة المُعطيات المُشفرة الكُليّة من أجل الرسائل ذات الحجم 80 KB.

4. لا تتجاوز قيمة الزمن اللازم لتشفير المفتاح السّري وتجهيز كُتلة المُعطيات المُشفّرة الكُليّة نسبة 5% من الزمن الكلّي اللازم لتوليد كُتلة المُعطيات المُشفرة الكُليّة .

5. يزداد الزمن الكُلّي اللازم لتوليد كُتلة المُعطيات المُشفّرة (التي تضم جُملة الرسالة المُذيلة بالتوقيع الرقمي المُشفّرة باستخدام المفتاح السّري K_S ، بالإضافة إلى المفتاح السّري المُشفّر باستخدام المفتاح العام لخوارزمية Elgamal) ، بنسبة تتراوح بين %(8-21) مع زيادة حجم الرسائل حتى 40 KB . بعد هذه القيمة يزداد هذا الزمن بشكل ملحوظ بنسبة تقدر بحوالي %(24-30) حتى الوصول إلى رسائل ذات حجم يُساوي 80 KB.

6. تُعرّف إنتاجية التشفير للنظام بأنها كمية المعلومات التي يُمكن تشفيرها في وحدة الزمن حيث يُبين الجدول (8)، أن إنتاجية التشفير للنظام المُصمم تزداد بشكل ملحوظ بنسبة تتراوح بين %(1-43) مع زيادة حجم الرسائل، وتبلغ قيمتها

العظمى (1969.33 Byte/sec) من أجل رسائل ذات حجم يُساوي 14 KB، بعد هذه القيمة تبدأ إنتاجية النظام بالإنخفاض تدريجياً بنسبة %(3-2)، حتى الوصول إلى حجم رسائل يُساوي 30 KB ، بعدها تنخفض إنتاجية النظام بشكل كبير بنسبة تتراوح بين %(7-3) مع زيادة حجم الرسائل حتى 80 KB . ويعود السبب في ذلك إلى ازدياد زمن التشفير للخوارزمية RC6 ، حيث يزداد هذا الزمن بشكل كبير بنسبة تتراوح بين %(18-12) مع زيادة حجم الرسائل من (80 KB←30) ،الأمر الذي يُؤدّي إلى إنخفاض إنتاجية التشفير للنظام بشكل كبير.

2-4 إختبار النظام المُصمم في إنجاز فك تشفير الرسائل:

استخدمنا البرنامج المُصمم في إنجاز فك تشفير رسائل ذات حُجوم مُختلفة ، وحصلنا على النتائج المُبينة في الجدول (9) ، حيث يُبيّن هذا الجدول الزمن المُستغرق (بالميللي ثانية) في كل مرحلة من مراحل النظام المُصمم عند فك تشفير رسائل ذات حُجوم مُختلفة. حيث يُعبّر الحقل الأول عن حجم الرسالة المُستقبلة مُقدّرة بالكيلو بايت ، ويُعبّر الحقل الثاني عن الزمن اللازم لفك تشفير واستخلاص المفتاح السّري للخوارزمية RC6 ، في حين يُعبّر الحقل الثالث عن الزمن اللازم لإنجاز فك تشفير جُملة الرسالة المُذيلة بالتوقيع الرقمي ، أما الحقل الرابع فيُعبّر عن الزمن اللازم للتحقق من تكاملية و مُصادقة الرسالة، ويُعبّر الحقل الخامس عن الزمن الكُلّي اللازم لاستخلاص الرسالة ومُصادقتها ، في حين يُعبّر الحقل السادس عن إنتاجية فك التشفير للنظام المُصمم مُقدّرة بواحدة (Byte/sec).

تُبين النتائج الواردة في الجدول (9) ما يلي :

1. إن الزمن اللازم لفك تشفير المفتاح السّري باستخدام المفتاح الخاص لخوارزمية Elgamal ، هو زمن ثابت تقريباً ولا يتعلق بحجم الرسالة، حيث لا تتجاوز قيمته نسبة 2% من الزمن الكُلّي اللازم لاستخلاص الرسالة ومُصادقتها.
2. إن الزمن اللازم للتحقق من تكاملية الرسالة و مُصادقتها هو زمن مُنخفض لا تتجاوز قيمته نسبة 5% من الزمن الكلي اللازم لاستخلاص الرسالة.
3. يزداد الزمن اللازم لفك تشفير جُملة الرسالة المُذيّلة بالتوقيع الرقمي بنسبة تتراوح بين %(20-10) مع زيادة حجم الرسائل حتى 40 KB ، بعد هذه القيمة يزداد هذا الزمن بشكل كبير بنسبة تتراوح بين %(27-23) ، حتى الوصول إلى رسائل ذات حجم يُساوي 80 KB.

الجدول (9) : الزمن المُستغرق في كل مرحلة من مراحل النظام المُصمم عند إنجاز فك تشفير رسائل ذات حُجوم مُختلفة

Message Size (KB)	Secret Key Decryption Time (msec)	(Message and Signature) Decryption Time (msec)	Verifying Time (msec)	Total Time (msec)	Decryption Throughput (Byte/Sec)
0.5	15	609	63	687	727.81
1	16	875	47	938	1066.1
1.5	15	1109	47	1171	1280.96
2	16	1344	47	1407	1421.47
2.5	16	1610	47	1673	1494.327
3	15	1829	47	1891	1586.47
3.5	16	2094	47	2157	1622.65
4	15	2328	47	2390	1673.64
4.5	16	2578	47	2641	1703.9
5	15	2829	47	2891	1729.5
6	15	3328	62	3405	1762.15
7	15	3798	63	3876	1805.98
8	16	4344	63	4423	1808.73
9	16	4873	63	4952	1817.44
10	15	5423	47	5485	1823.15

12	15	6502	46	6563	1828.43
14	16	7576	63	7655	1828.87
16	15	8718	47	8780	1822.32
19	15	10485	63	10563	1798.73
22	16	12407	47	12470	1764.23
26	16	14967	46	15029	1729.99
30	15	17687	62	17764	1688.81
34	15	20812	63	20890	1627.57
38	16	23954	47	24017	1582.2
44	15	30016	62	30093	1462.13
50	16	36155	47	36218	1380.53
56	16	42594	47	42657	1312.8
63	16	52313	62	52391	1202.5
70	15	65266	63	65344	1071.25
80	16	83016	62	83094	962.76

4. تُعرّف إنتاجية فك التشفير للنظام بأنها كمية المعلومات التي يُمكن فك تشفيرها في وحدة الزمن ، حيث يُبيّن الجدول (9) ، أن إنتاجية فك التشفير للنظام المُصمّم تزداد بشكل ملحوظ بنسبة تتراوح بين %(46-1) مع زيادة حجم الرسائل، وتبلغ قيمتها العُظمى (1828.87 Byte/sec) من أجل رسائل ذات حجم يُساوي 14 KB ، بعد هذه القيمة تبدأ إنتاجية النظام بالإنخفاض تدريجياً بنسبة %(2-1) حتى الوصول إلى حجم رسائل يُساوي 30 KB. بعدها تنخفض إنتاجية النظام بشكل كبير بنسبة تتراوح بين %(10-4) مع زيادة حجم الرسائل حتى 80 KB . ويعود السبب في ذلك إلى ازدياد زمن فك التشفير للخوارزمية RC6 ، حيث يزداد هذا الزمن بشكل كبير بنسبة تتراوح بين %(27-18) مع زيادة حجم الرسائل من (30→80 KB) ، الأمر الذي يُؤدّي إلى إنخفاض إنتاجية فك التشفير للنظام بشكل كبير.

الاستنتاجات والتَوصيات:

من خلال الدراسة المقارنة التحليلية التي أجريناها لتقييم أداء وفعالية خوارزميات التشفير المُستخدمة في بناء المُخطط الأمني العام لنُظم أمن البريد الإلكتروني الهجينة، ومن خلال دراسة واختبار النتائج التي حصلنا عليها من النظام المُصمم ، يُمكننا تقديم التّوصيات والمُقترحات التالية :

1- تُحقق الخوارزميات RC6 – Elgamal - MD5 ، مُستوى أمنٍ عالٍ وكفاءة جيدة وبساطة في التصميم وبشكل عام يمكن الإعتماد عليها في تصميم النظم الهجينة التي تساعد في ضمان أمن رسائل البريد الإلكتروني.

2- من خلال دراستنا للخوارزمية RC6 ، لاحظنا وجود علاقة عكسية بين زيادة عدد الدورات و سرعة الأداء لذا لا يُحبّذ تصميم هذه الخوارزمية بعدد دورات كبير، لأن ذلك سيؤثر سلباً على سرعة و إنتاجية النظام ، حيث وجدنا حسب Knudsen أن العدد الأصغري من الدورات اللازمة لتحقيق مُستوى الأمن المطلوب في هذه الخوارزمية هو 16 دورة ، لذا قُمنا بتصميمها بعدد دورات يساوي 20 دورة، وذلك لإعطاء هامش أمن مقبول لهذه الخوارزمية يُحقق تطلعاتنا في التصميم دون أن يُؤثر ذلك على أداء النظام.

3- تُحقق خوارزمية Elgamal أداء جيد في النظام المُصمم من خلال استخدامها لإنجاز التوقيع الرقمي وأيضاً لتشفير مُعطيات ذات حجم ثابت دائماً مهما كان حجم الرسالة، حيث تُستخدم لإنجاز التوقيع الرقمي بواسطة رمز البعثرة للرسالة ذو الطول 128 bits ، كما تُستخدم لتشفير المفتاح السّري للخوارزمية RC6 ذي الطول 256 bits .

4- تُبدي الخوارزمية MD5 كفاءة عالية وسرعة جيدة في إيجاد مُلخّص الرسالة (رمز البعثرة) ، حيث يُمكن إهمال الزمن اللازم لإيجاد مُلخّص الرسالة، من أجل الرسائل التي يتراوح حجمها بين (60 KB → 1) وذلك لصغره الشديد.

5- من نتائج اختبارات نظام أمن البريد الإلكتروني الهجين المُصمم عند إنجاز التشفير وفك التشفير تبيّن لنا أن هذا النظام يبدي أداء جيد من حيث السرعة والإنتاجية من أجل رسائل ذات حجم يصل حتى 14 KB، ولكن يمكن استخدامه لإرسال رسائل يصل حجمها حتى 30 KB دون أن يؤثر ذلك على أداء النظام ، وفي الحقيقة إن الحجم 30 KB هو حجم جيد للرسائل النصية المُرسلة والتي قد تتضمن عقود أو فواتير ..... الخ، والتي يكون حجمها ضمن مجال (30 KB → 14) .

6- نلاحظ أن زيادة حجم الرسالة المُرسلة أو المُستقبلة أكثر من 40 KB تقريباً، تؤدي إلى انخفاض تدريجي في سرعة وإنتاجية النظام المُصمم، لذا نُوصي بتجزئة الرسائل ذات الحُجوم الكبيرة و إرسالها على دفعات للحصول على أداء أفضل لهذا النظام.

References:

1.Murphy, S. *Cryptography: A Very Short Introduction*, Oxford University Press, 2012.

2. PFLEEGER,C. *Security in Computing*. 4th. ed. Prentice Hall, New York, October 2006, 880.

3. RHEE,M.Y. *Cryptographic principles, Algorithms, and protocols*. 2nd. ed. John Willey & Sons, Seoul National University, 2003, 391.

4. STALLINGS,W. *Cryptography and Network Security Principles and Practices*. 4th. ed. , Prentice Hall, New York, November 2005, 592.

5. SINGH,S.;SUDESH,K.;MAAKAR,K. *Enhancing the Security of DES Algorithm Using Transposition Cryptography Techniques*, International Journal of Advanced Research in Computer Science and Software Engineering, Vol.3,Issue 6, June 2013.

6. MONIKA,A.;PRADEEP,M. *A Comparative Survey on Symmetric Key Encryption Techniques*, International Journal on computer science and Engineering (IJCSE), Vol.4,No.5 May 2014, pp. 877-882.

7. ELMINAAM,D.;KADER,H.;HADHOUD,M. *Evaluating The Performance of Symmetric Encryption Algorithms* , International Journal of computer science and information security, Vol.10, No.3,May 2015, pp. 213-219.

8. MOHAN,H.;RAJI,R. *Performance Analysis of AES and MARS Encryption Algorithms*, International Journal of computer science and information security, Issues (IJCSI),Vol.8,2016.

9.LUNG SU,S.;CHYAU WUU,L.;WEI JHANG,J. *A New 256-bits Block Cipher-Twofish256*, 2016.

10. PRASAD,B. *A Performance Study on AES Algorithm*, International Journal of computer science and information security, Vol.8, No.6, September 2016, pp. 128-132.

11. SARODE,P.;GUPTA,P. *A Comparative Analysis of RSA and MD5 Algorithm*, Journal of computer science and Applications, pp.25-33,2014.

12. OKEYINKA,A. *Computational Speeds Analysis of RSA and ELGAMAL Algorithms on Text Data*, Proceedings of the World Congress on Engineering and Computer Science 2015,Vol.1.October 21-23 2015,San Francisco, USA.

13. Wikipedia, MD5, Available at : https//en.wikipedia.org/wiki/MD5.(2015).

14. KASGAR, A.K. ;*New Modified 256- bit MD5 Algorithm with SHA Compression Function*, International Journal of computer Applications, 2014.

15. DAEMEN,J.;RIJMEN,V. *The Design of Rijndael: AES-The Advanced Encryption Standard*, Springer 2012.