تطوير خوارزمية RSA لضمان التحقق وانسيابية البيانات

*رغد حيدري ** محمد ضاهر

(الإيداع: 25 شباط 2018، القبول: 4 نيسان 2018)

الملخص:

تعتمد خوارزمية RSA على توليد المفاتيح ثم إجراء عملية التشفير من خلال تجزئة الرسالة وتطبيق معادلة التشفير الخاصة بهذه الخوارزمية على كل جزء من أجزاء الرسالة، لكن هذا الأمر ينعكس سلباً على سرعة عملية التشفير وبالتالي انخفاض انسيابية مستوى البيانات المطلوب حمايتها وإرسالها عبر الشبكة، ويزداد مستوى الانسيابية انخفاضاً عند إجراء عملية التحقق من مرسل ومستقبل كتل هذه البيانات، لذلك يقدم هذا البحث آلية مقترحة لتطوير خوارزمية RSA بحيث نضمن التحقق من مرسل ومستقبل البيانات مع تحقيق مستوى انسيابية عالي لها.

أثبتت الآلية المطورة تفوقها على خوارزمية RSA الأصلية من أجل حماية البيانات بنسبة 0.56 ثانية من أجل عملية التشفير و 0.48 ثانية من أجل عملية فك التشفير، كما بلغت نسبة تفوقها من أجل عملية الحماية والتحقق 1.202 ثانية لزمن التشفير و0.936 ثانية لزمن فك التشفير، وذلك لخمس مستندات نصية مختلفة الحجم.

الكلمات المفتاحية: خوارزمية RSA، التشفير، فك التشفير، الانسيابية، التحقق.

* طالبة دراسات عليا (دكتوراه)، قسم الإحصاء ونظم المعلومات كلية الاقتصاد، جامعة حلب.

**أستاذ مساعد في قسم الإحصاء ونظم المعلومات كلية الاقتصاد، جامعة حلب.

Develop Algorithm RSA to Ensure Authentication and Smooth Flow Data

*Raghad HIDAREY **Mohammad, DAHER

(Received: 25 February 2018, Accepted:4 April 2018)

Abstract:

The RSA algorithm relies on key generation then process of encryption by fragmenting message and applying cryptographic equation on each part of the message.
but this operation negatively effects on speed of the encryption process, thus reducing the flow of the data level that require to be protected and transmit over the network. When conducting a verification of the sender and receiver of blocks of data, this research provides a proposed mechanism for the development of the RSA algorithm to ensure that the sender and the receiver of the data is verified with a high level of flow, the developed mechanism has proved superior to the RSA algorithm For protection of data by 0.56 seconds for the encryption process and 0.48 seconds for the decryption process, and for protection and verification, 1.202 seconds for encryption time and 0.936 seconds for decoding time for five different text documents In size

KEY WORDS: Algorithm RSA, Encryption, Decryption, Smoth Flow, Authentication.

*Postgraduate Student (PhD), Dept. of Statistics and Information Systems, Faculty of Economics, University of Aleppo.

**Dept. of Statistics and Information Systems, Faculty of Economics, University of Aleppo.

1-المقدمة:

تستخدم خوارزميات التشفير لحماية البيانات المنقولة عبر الشبكة الحاسوبية، فهي تقوم بعملية تحويل النص الأصلي Plain Text إلى نص مشفر Cipher Text لضمان حمايته بحيث لا يمكن الاستفادة من النص المشفر إلا عند القيام بعملية فك التشفير واستخلاص النص الأصلي منه[1]، وتعتبر خوارزمية RSA واحد من تقنيات التشفير غير المتناظر التي تقوم بتجزئة النص الأصلي إلى كتل جزئية وتطبيق معادلة التشفير باستخدام مفتاح التشفير Encryption Key في طرف المرسل، وتطبيق معادلة فك التشفير باستخدام مفتاح فك التشفير Decryption Key في طرف المستقبل، ثم تجميع البيانات الكتلية للحصول على كامل النص الأصلي[2].

2- مشكلة البحث:

انخفاض مستوى انسيابية (تدفق البيانات) البيانات المراد إرسالها عبر الشبكة عند تشفيرها وحمايتها باستخدام خوارزمية RSA نتيجة وجود عامل الرفع إلى قوة وباقي القسمة (العمليات الرياضية) في معادلتي التشفير وفك التشفير، حيث يزداد زمني التشفير وفك التشفير بشكل طردي مع ازدياد حجم البيانات المرسلة، ويزداد انخفاض مستوى الانسيابية عند تطبيق خوارزمية RSA من أجل عملية التحقق والحماية لإن ذلك يتطلب ضعف العمليات الرياضية المطلوبة على البيانات من أجل عمليتي التشفير وفك التشفير.

3-أهداف البحث:

يهدف البحث إلى تطوير آلية تشفير البيانات المنقولة عبر الشبكة الحاسوبية بحيث يمكن من خلال استخدامها تقليل الزمن اللازم لإجراء عمليتي التشفير وفك التشفير باستخدام خوارزمية التشفير RSA مما ينعكس ايجابياً على معدل انسيابية البيانات سواءاً في حالة تأمين الحماية للبيانات المنقولة أو في حال إجراء عملية التحقق من مرسلها ومستقبلها، أي زيادة معدل تدفق البيانات مع الحفاظ على سريتها وتأمين التحقق من مستلمها.

4-مواد وطرائق البحث:

تم إجراء البحث وفق ثلاث خطوات وهي:

الخطوة الأولى: دراسة وتحليل آلية عمل خوارزمية التشفير RSA الأصلية (الأساسية) وكيفية تطبيقها من أجل حماية البيانات والتحقق.

الخطوة الثانية: اقتراح آلية مطورة لإجراء خوارزمية التشفير RSA من أجل حماية البيانات والتحقق أيضاً.

الخطوة الثالثة: إجراء تطبيق عملي لكلتا الخوارزميتين (خوارزمية التشفير RSA الأصلية _ RSA المطورة) وتطبيقهما على مجموعة من المستندات النصية ومن ثم إجراء المقارنة بين الخوارزميتين وفق آليتين الأولى من أجل حماية البيانات والثانية من أجل الحماية والتحقق.

4-1 – حماية البيانات باستخدام خوارزمية التشفير RSA الأصلية والتحقق منها:

اقترح العلماء Ron Rivest, Adi Shamir and Len Adlemen في عام 1977 خوارزمية التشفير RSA وسميت باسمهم، وهي خوارزمية المفتاح العام (غير المتناظرة)، وتعد من إحدى خوارزميات التشفير الأكثر استخداماً في مجال تشفير المعلومات المنقولة عبر الشبكات الحاسوبية [3,4,5]، ولا سيما في أنظمة الحكومات الإلكترونية التي دخلت حيز التطبيق في العديد من الدول المتقدمة[7] ، وتعتمد آلية عملها في حال كان لدينا مجموعة من البيانات تشكل رسالة M ومطلوب حمايتها باستخدام خوارزمية RSA على الخطوات التالية[6]:

A. نقوم بتجزئة الرسالة الأصلية M إلى كتل بيانات: كالاتي:

$$M=[m_1, m_2,\ldots\ldots, m_z] \qquad (1)$$

حيث: z عدد كتل البيانات.

و$m1, m2$: كتل البيانات (الكتلة الأولى، الكتلة الثانية، ..، الكتلة الأخيرة)

B. يتم تشفير كتل البيانات وفق علاقة التشفير الخاصة بخوارزمية RSA:

$$c_i = m_i^{er} \bmod n \qquad (2)$$

حيث C_i الكتلة المشفرة الموافقة لكتلة البيانات m_i

وتمثل $[e_r,n]$ المفتاح العام للمستلم والذي يتم الحصول من عملية توليد المفاتيح التي سيتم شرحها.

C. يتم تشكيل النص المشفر C من الكتل المشفرة c_i كالاتي:

$$C=[c_1, c_2,\ldots\ldots\ldots, c_z] \qquad (3)$$

في طرف المستقبل يتم فك تشفير الرسالة المشفرة C بالعلاقة التالية:

$$m_i = c_i^{dr} \bmod n \qquad (4)$$

وتمثل $[d_r, n]$ المفتاح الخاص للمستلم.

D. عملية توليد المفاتيح والتي تتألف مما يلي:

1. توليد عددين أوليين ونرمز لهما بـ p،.q

2. حساب قيمة المعامل n والذي يساوي جداء العددين الأولين p، q

$$n=p*q \qquad (5)$$

3. حساب قيمة $\phi(n)$ والتي تحسب من العلاقة

$$\phi(n)=(p-1)(q-1) \qquad (6)$$

4. تم اختيار عدد عشوائي eكمفتاح عام والذي يوجد بالعلاقة:

$$GCD(\phi(n), e) = 1 \qquad (7)$$

5. اختيار المفتاح الخاص d، والذي يتم حسابه بعكس قيمة eبالاعتماد على قيمة $\phi(n)$، وتكون قيمته وفق العلاقة:

$$d*e = 1 \bmod \phi(n) \qquad (8)$$

E. في حال أردنا إجراء عملية التحقق وحماية البيانات:

1. يتم تشفير البيانات الكتلية باستخدام المفتاح الخاص للمرسل $[d_s,n]$ ومن ثم المفتاح العام للمستلم $[e_r,n]$ كما يلي:

$$c_i = [m_i^{ds} \bmod n]^{er} \bmod n \qquad (9)$$

2. في طرف المستقبل يتم فك التشفير باستخدام المفتاح الخاص للمستلم[dr,n] ومن ثم المفتاح العام للمرسل$[e_s,n]$ وفق العلاقة:

$$m_i = [c_i^{dr} \bmod n]^{es} \bmod n \qquad (10)$$

هذه الآلية تضمن عدم الوصول للنص الأصلي للرسالة إلا من قبل المستلم حصراً لأنه الوحيد الذي يمتلك المفتاح الخاص الأمر الذي يجعل مرسل الرسالة يتحقق أن البيانات لا يمكن الاطلاع عليها إلا من قبل المستلم الذي يحدده، كما أن المستلم

يقوم بفك تشفير النص المشفر باستخدام المفتاح العام للمرسل وبالتالي يستطيع معرفة هوية الشخص مرسل هذه الرسالة، ومنه لا يمكن انكار المرسل من إرسال الرسالة أو المستلم من أنه الوحيد القادر على استلامها، وبالتالي هذه الآلية تسهم بالتحقق من مرسل ومستلم البيانات.

2-4- خوارزمية RSA المطورة:

تكمن عملية التأخير الحاصلة في طريقة التشفير وفك التشفير في خوارزمية RSA بإجراء عملية الرفع إلى قوة لأكثر من مرة، لذلك تم اقتراح تطوير RSA بحيث يتم إجراء التشفير في الطرف المرسل لهذه الخوارزمية المطورة على خطوتين من أجل الحصول على النص المشفر C :

الخطوة الأولى: يتم تقسيم الرسالة M إلى كتل بيانات ذات عدد خانات ثابت يتناسب مع طول المعامل n كالتالي:

$$M= [m_1, m_2, \ldots\ldots, m_N] \qquad (11)$$

حيث N عدد كتل البيانات المشكلة للرسالة M وتحسب وفق العلاقة:

$$N = \frac{NO.bits\ of\ M}{NO.bits\ of\ n} \qquad (12)$$

الخطوة الثانية: يتم حساب شيفرات الكتل الجزئية وفق العلاقة المقترحة التالية:

$$c_i= \begin{cases} Fc\ (m_1)\ ;\ i=1 \\ m_1\ xor\ m_i;\ i=2, 3,..,N \end{cases} \qquad (13)$$

حيث **fc** هو عبارة عن تابع تشفير الغاية منه حساب شيفرة كتلة البيانات الأولى c_1، ويعطى بالعلاقة:

$$c_1= fc(m_1) = \begin{cases} m_1^{er} mod\ n; & F=0 \\ (m_1\ xor\ n)^{er} mod\ n\ ; & F=1 \end{cases} \qquad (14)$$

حيث F متغير دلالي لمعرفة مقدار m_1 بالنسبة لـ n ويحسب بالعلاقة التالية:

$$F = \begin{cases} 0 & ;\ m_1 < n \\ 1 & ;\ m_1 > n \end{cases} \qquad (15)$$

وبعد ذلك يتم إرسال النص المشفر التالي:

$$C=[c_1, c_2, \ldots, c_N, F] \qquad (16)$$

ويتم إجراء فك التشفير في الطرف المستلم وفق العلاقة التالية:

حيث fc^{-1} التابع العكسي للتابع **fc** المعطى بالعلاقة (14) ويعطى هذا التابع بـ:

$$m_i= \begin{cases} fc^{-1}(c_1);\ i=1 \\ m_1\ xor\ c_i;\ i=2, 3,\ldots N \end{cases} \qquad (17)$$

$$m_1= fc^{-1}(c_1) = \begin{cases} c_1^{dr} mod\ n\ ; & F=0 \\ [c_1^{dr}\ mod\ n]\ xor\ n\ ; & F=1 \end{cases} \qquad (18)$$

تؤمن الخوارزمية المطورة السابقة حماية البيانات لإن الكتلة الأولى m_1 لا يمكن للمستلم الحصول عليها إلا بعد فك تشفير c_1 وهذا لا يمكن إلا في حال كان يمتلك المفتاح الخاص $[d_r,n]$، كما أن الخوارزمية المطورة تستغرق تكلفة حسابية أقل.

يمكن تطبيق الآلية المقترحة لإجراء عملية التشفير وفك التشفير لضمان التحقق (التحقق من مرسل ومستلم الرسالة) بالإضافة لحماية البيانات وذلك باستبدال العلاقة (14) بالعلاقة التالية:

$$c_1= fc_r(fc_s) = \begin{cases} fc_s^{er} \bmod n ; & Fr=0 \\ (fc_s \text{ xor } n)^{er} \bmod n ; & Fr=1 \end{cases} \quad (19)$$

حيث fc_s هو عبارة عن تابع تشفير يعطى بالعلاقة:

$$fc_s(m_1) = \begin{cases} m_1^{ds} \bmod n ; & Fs=0 \\ (m_1 \text{ xor } n)^{ds} \bmod n ; & Fs=1 \end{cases} \quad (20)$$

حيث $F=Fs$ ، أما Fr فيعطى بالعلاقة:

$$Fr = \begin{cases} 0 & ; fc_s < n \\ 1 & ; fc_s > n \end{cases} \quad (21)$$

يتم حساب شيفرات الكتل الجزئية وفق العلاقة المقترحة التالية

$$C_i= \begin{cases} fc_r(fc_s) ; & i=1 \\ m_1 \text{ xor } m_i ; & i=2, 3,...N \end{cases} \quad (22)$$

ويكون النص المشفر معطى بالعلاقة:

$$C=[c_1, c_2,....,c_N, Fs,Fr] \quad (23)$$

أما عملية فك التشفير لضمان التحقق وحماية البيانات فتتم باستخدام الخوارزمية المطورة باستبدال العلاقة 18 بالعلاقة التالية:

$$m_1= fc_r^{-1}(fc_s)^{-1} = \begin{cases} [fcs^{-1}]^{dr} \bmod n ; & Fr=0 \\ [([fcs^{-1}]^{dr} \text{ xor } n) \bmod n] ; & Fr=1 \end{cases} \quad (24)$$

حيث تابع فك التشفير fcs^{-1} معطى بالعلاقة التالية:

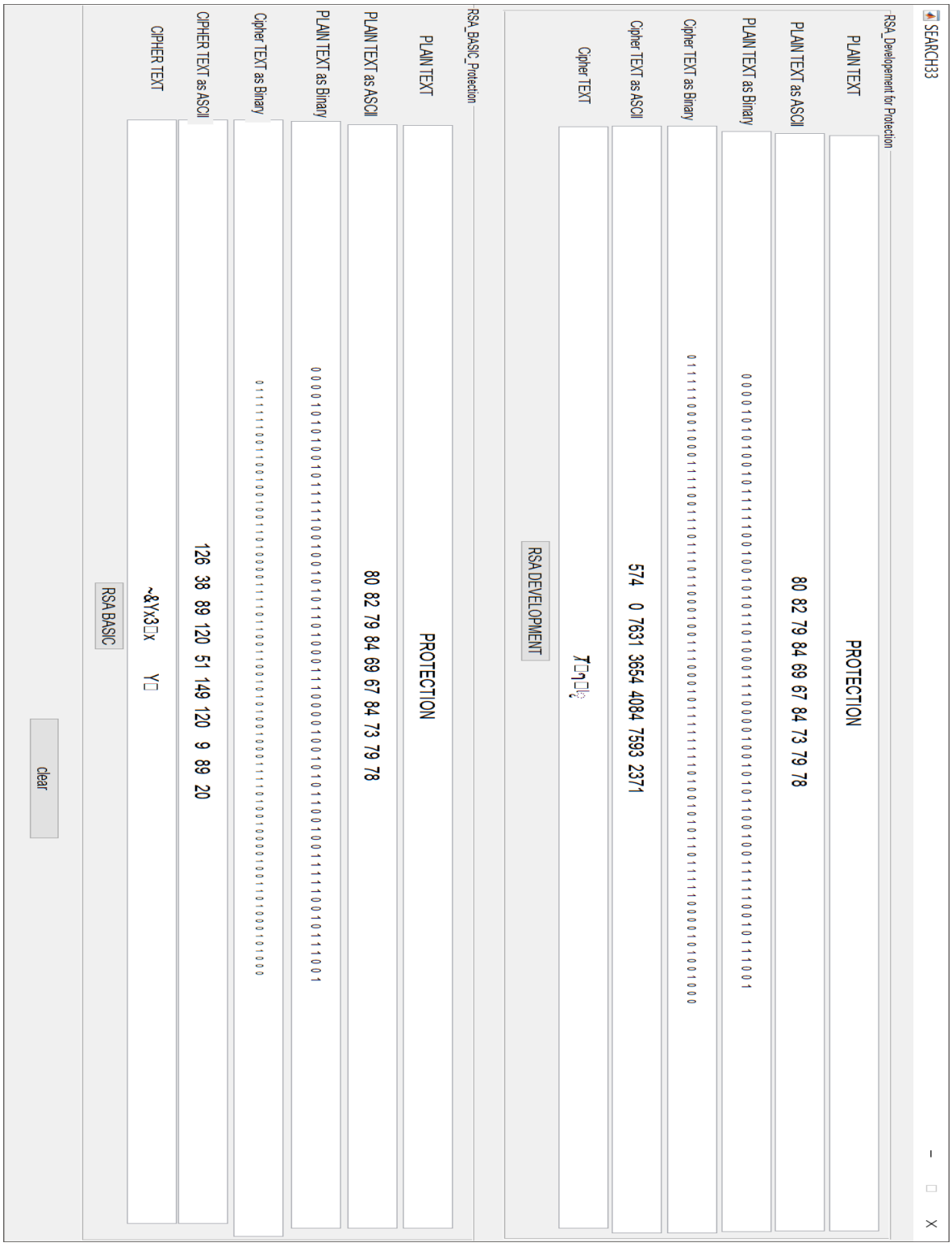
$$fc_s^{-1} = \begin{cases} [c_1]^{es} \bmod n ; & Fs=0 \\ [([c_1 \text{ xor } n)^{es} \bmod n] ; & Fs=1 \end{cases} \quad (25)$$

5_ النتائج والمناقشة:

تم تطبيق خوارزمية RSA الأصلية والمطورة باستخدام برنامج MATLAB كما هو مبين بالشكلين التاليين (1,2)، تم تصميم الشكل الأول من أجل حماية البيانات والشكل الثاني من أجل عملية الحماية وتوثيق البيانات، حيث قسمت كل واجهة إلى قسمين القسم الأول يعرض خوارزمية RSA المطورة وخوارزمية RSA الأصلية:

1–**الحالة الأولى:** حماية البيانات(Protection):

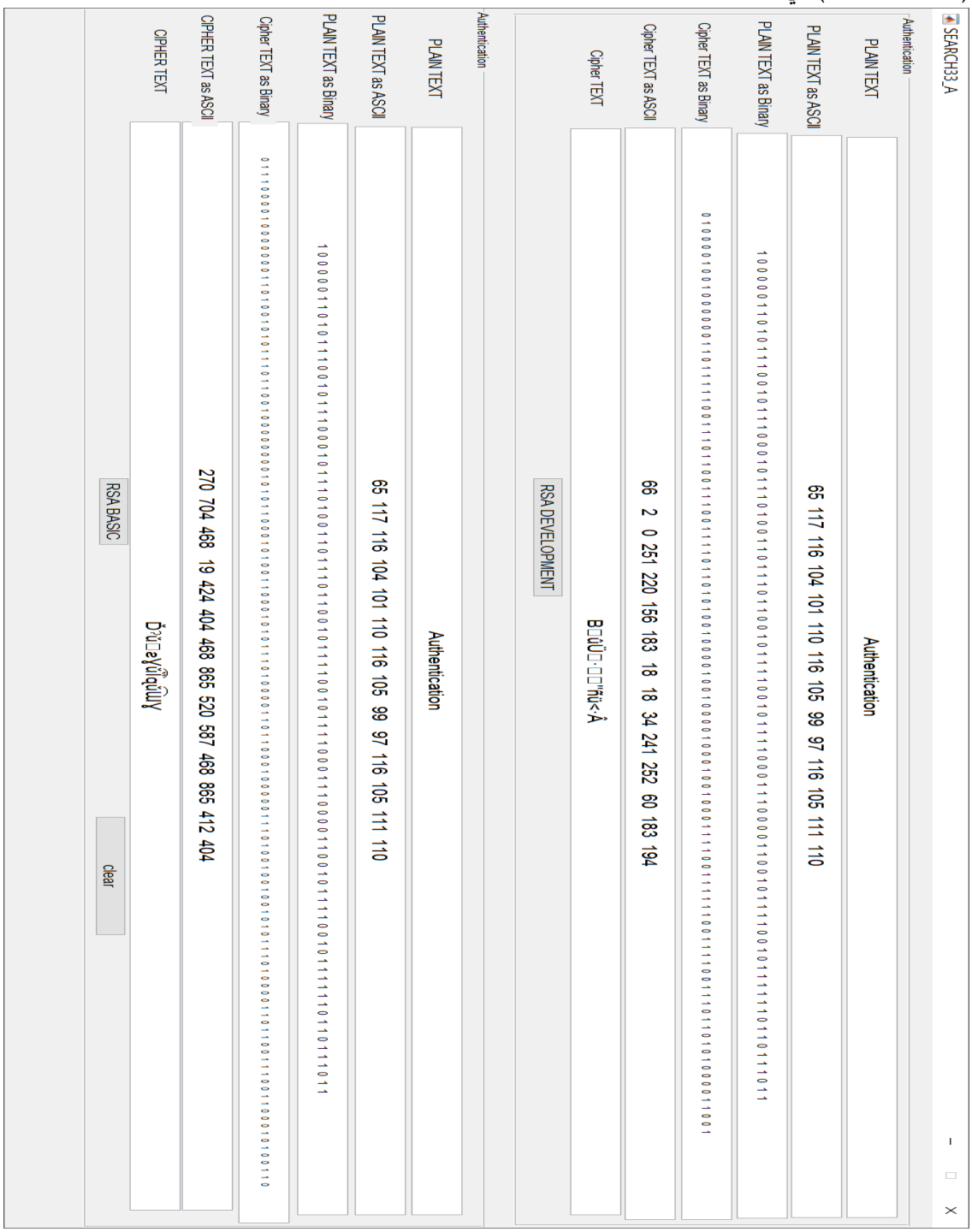
يتم إدخال النص الأصلي plain text وبالضغط على الزر RSA_Basic يتم أخذ قيم ASCII الموافقة لمحارف كل النص وتحويلها لقيم ثنائية ثم تطبيق معادلة التشفير المعطى بالعلاقة 2، أما عند النقر على زر RSA_Development يتم تطبيق خوارزمية RSA المطورة والمستخدمة من أجل حماية البيانات والمعطاة بالعلاقات من 11 حتى 15.



الشكل رقم (1): الواجهة البرمجية لإجراء التشفير باستخدام خوارزمية RSA الأصلية والمطورة من أجل عملية الحماية

2-الحالة الثانية: توثيق وحماية البيانات (Authentication & Protection):

يتم في هذا التطبيق إجراء عملية التوثيق والتحقق عن طريق الزر RSA_Basic حيث يتم تطبيق العلاقة 9 على كتل بيانات الرسالة المدخلة في Plain Text، أما خوارزمية RSA المطورة من أجل توثيق وحماية البيانات فقد تم برمجة العلاقات (19,20,21,22) في الزر RSA_Development.



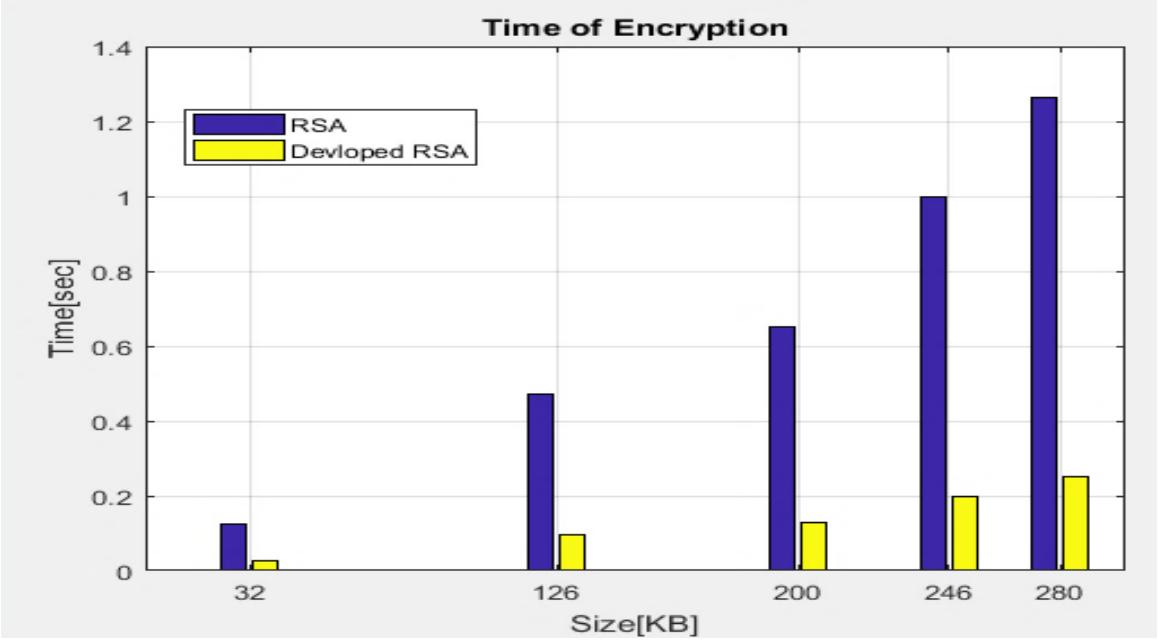
الشكل رقم (2): الواجهة البرمجية لإجراء التشفير باستخدام RSA الأصلية والمطورة من أجل عملية الحماية وتوثيق البيانات

تم إجراء عملية المقارنة بين خوارزمية RSA الأصلية والمطورة من أجل زمني التشفير وفك التشفير وذلك بالنسبة لخمسة بيانات نصية ذات أحجام kb [32,126,200,246,280] على الترتيب وتمت المقارنة للحالات التالية:

الحالة الأولى: عملية حماية البيانات فقط:

1. **من أجل عملية التشفير:**

يبين الشكل (3) زمن التشفير لكل من خوارزمية RSA الأصلية والمطورة، حيث نلاحظ انخفاض زمن تشفير الخوارزمية المطورة ومنه وجود نسبة تحسين في أداء الخوارزمية هذا التحسين يؤدي إلى زيادة انسيابية البيانات عبر الشبكة وذلك لنقصان الزمن اللازم لتشفيرها.

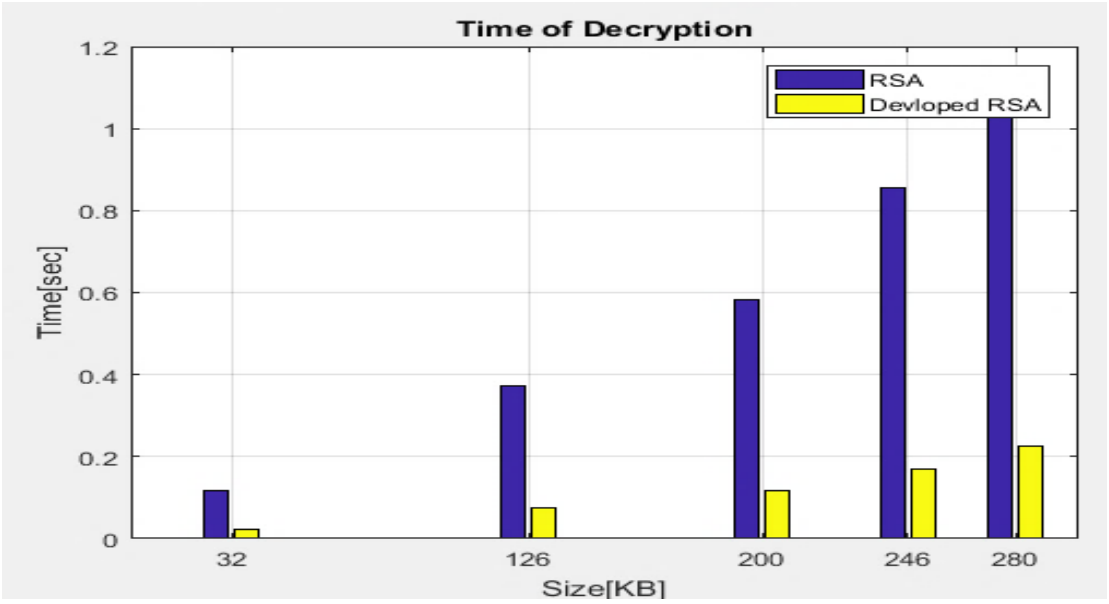


الشكل رقم(3):المقارنة في زمن التشفير بين خوارزمية RSA الأصلية والمطورة من أجل عملية الحماية

2. **من أجل عملية فك التشفير:**

يبين الشكل (4) زمن فك التشفير لكل من خوارزمية RSA الأصلية والمطورة، حيث نلاحظ انخفاض زمن فك تشفير الخوارزمية المطورة عن زمن فك تشفير الخوارزمية الأصلية.

الشكل رقم (4): المقارنة في زمن فك التشفير بين خوارزمية RSA الأصلية والمطورة من أجل عملية الحماية



يرجع سبب انخفاض زمن التشفير وفك التشفير في خوارزمية RSA المطورة إلى قلة العمليات الرياضية المستخدمة فيها مقارنة مع خوارزمية RSA الأصلية، حيث تقوم خوارزمية RSA الأصلية بإجراء عمليات الرفع إلى قوة وباقي القسمة لكل

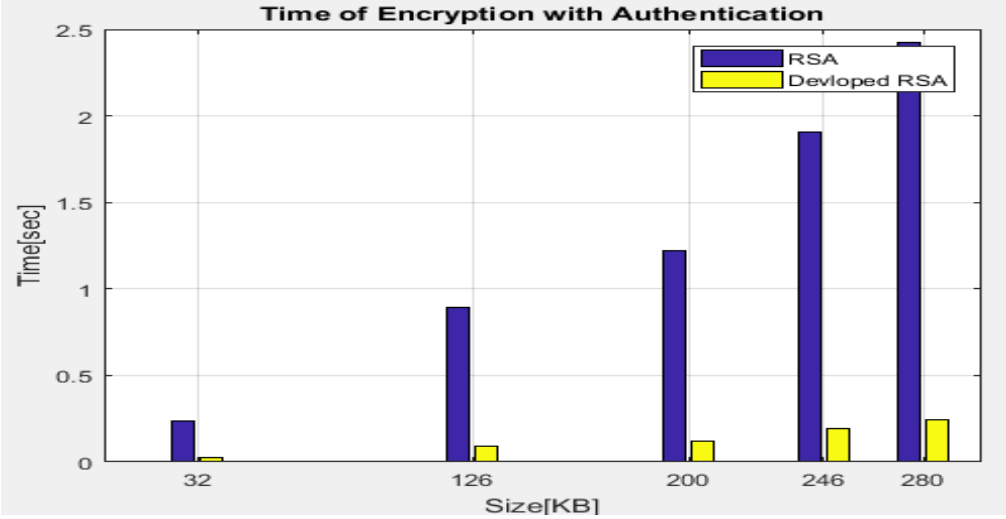
كتل بيانات الرسالة، في حين أن خوارزمية RSA المطورة تقوم بهذه العمليات لأول كتلة من بيانات الرسالة فقط، في حين تقتصر على العملية XOR من أجل تشفير أو فك تشفير باقي الكتل النصية للرسالة، لذلك نلاحظ وجود زيادة طفيفة في زمن التشفير وفك التشفير من أجل خوارزمية RSA المطورة مع زيادة حجم النص كما هو مبين بالشكلين(3,4) ومنه زيادة نسبة التحسين في كل من زمن التشفير وفك التشفير مع ازدياد حجم البيانات.

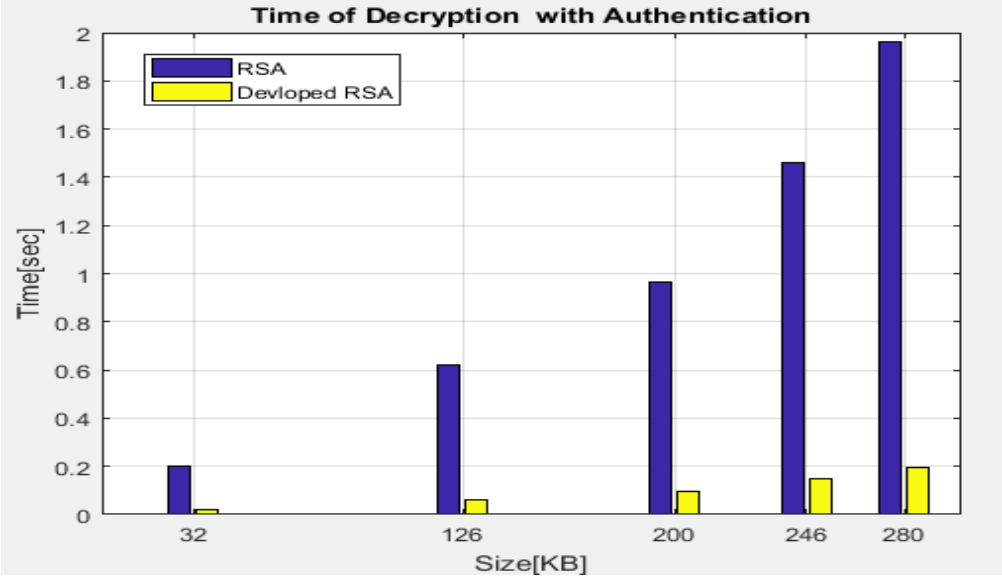
الحالة الثانية: عملية توثيق البيانات وحمايتها:

من أجل عملية التشفير وفك التشفير:

نلاحظ من الشكلين (6,5) زيادة انخفاض زمني التشفير وفك التشفير لخوارزمية RSAالمطورة عن الأصلية، وبالتالي زيادة نسبة التحسين في الخوارزمية المطورة لكل من زمني التشفير وفك التشفير.

الشكل رقم (5): المقارنة في زمن التشفير بين خوارزمية RSA الأصلية والمطورة من أجل عملية التوثيق وحماية البيانات





الشكل رقم (6): المقارنة في زمن فك التشفير بين خوارزمية RSA الأصلية والمطورة من أجل عملية التوثيق وحماية البيانات

ويرجع سبب زيادة انخفاض زمن التشفير وفك التشفير في الخوارزمية المطورة إلى أن الخوارزمية الأصلية من أجل التوثيق وحماية البيانات تقوم باستخدام ضعف العمليات الرياضية التي تحتاجها لإجراء عملية حماية البيانات فقط، وذلك نتيجة لتشفير البيانات عند إرسالها مرتين على التوالي مرة في المفتاح الخاص للمرسل ثم في المفتاح العام للمستلم كما تبينه العلاقة (9)، أما الخوارزمية المطورة فهي تستخدم نفس عدد العمليات المنطقية (XOR) من أجل حماية البيانات فقط أو توثيق وحماية البيانات معاً وتختلف فقط في عدد العمليات الرياضية (الرفع إلى قوة وباقي القسمة) المطبقة على أول كتلة من كتل البيانات، حيث تكون من أجل توثيق وحماية البيانات ضعف العدد الذي تحتاجه لإجراء عملية حماية البيانات فقط.

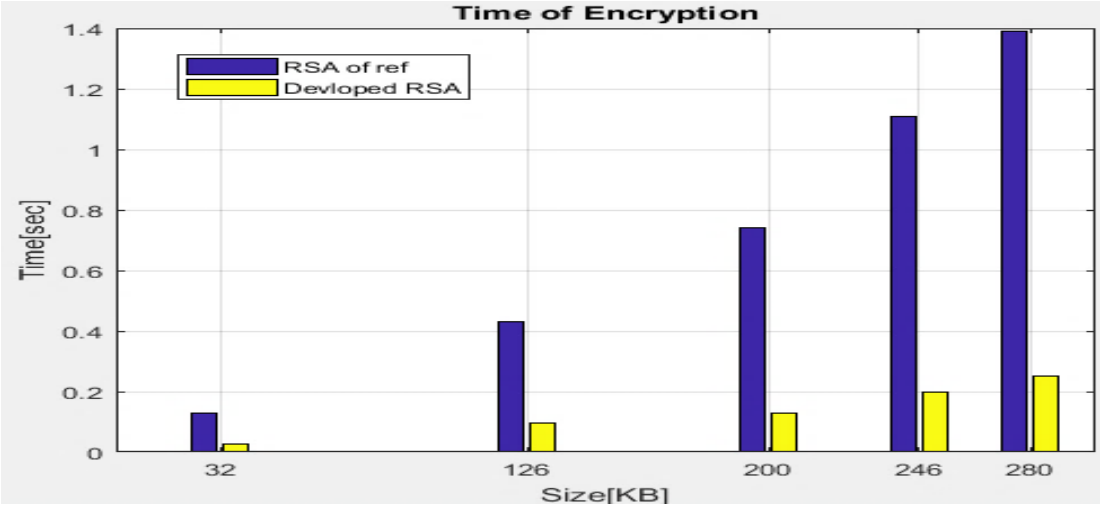
6_المقارنة:

قام الباحثان(Faiqa Maqsood & others)[2] بإجراء مقارنة بين أكثر من خوارزمية تشفير، إحدى الخوارزميات التي قاموا بدراستها وتطبيقها هي خوارزمية RSA ويبين الجدول(1) زمن التشفير وفكه لخوارزمية RSA:

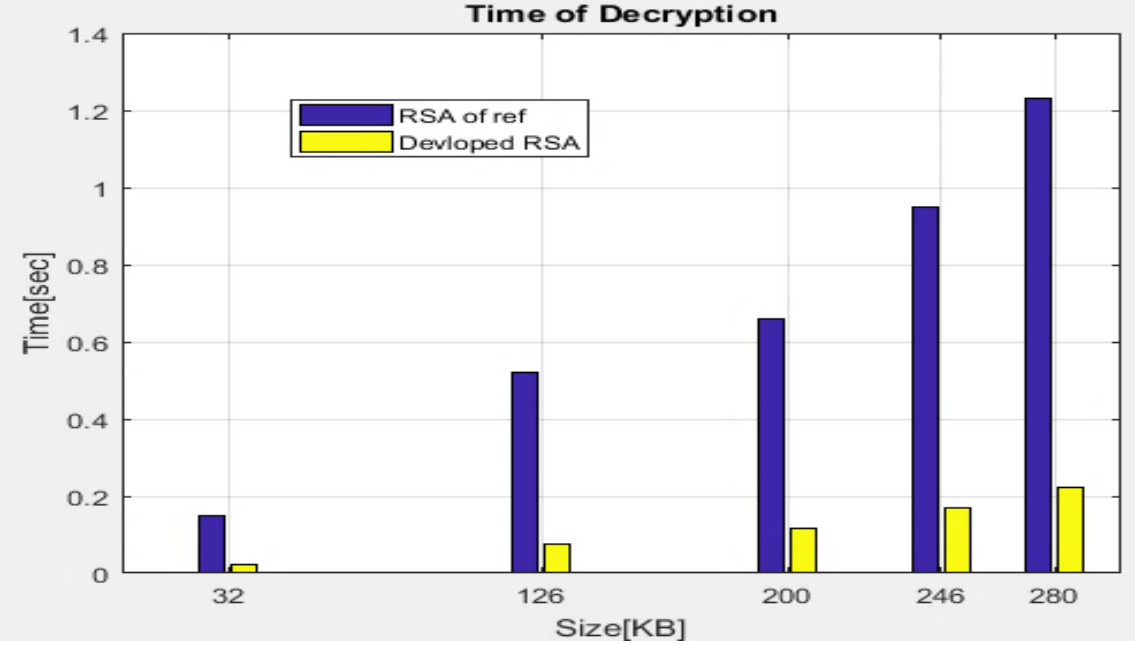
الجدول رقم (1): المقارنة بين خوارزميات التشفير المتناظرة وغير المتناظرة

Cryptography Algorithms	*File size (kilo bytes)*	*Encryption Time (in Seconds)*	*Decryption Time (in Seconds)*
DES	32	0.27	0.44
	126	0.83	0.65
	200	1.19	0.85
	246	1.44	1.23
	280	1.67	1.45
AES	32	0.15	0.15
	126	0.46	0.44
	200	0.72	0.63
	246	0.95	0.83
	280	1.12	1.10
RSA	32	0.13	0.15
	126	0.52	0.43
	200	0.74	0.66
	246	1.11	0.93
	280	1.39	1.23

وقد قمنا في هذا البحث بتحصيل النتائج لخمسة نصوص بأحجام متوافقة مع أحجام النصوص التي استخدمها الباحثان لنتمكن من إجراء المقارنة بين خوارزمية RSA المطورة في هذا البحث مع خوارزمية RSA التي قام الباحثان بتطبيقها[2]، ويبين الشكلان (7,8) هذه المقارنة من أجل عمليتي التشفير وفك التشفير.

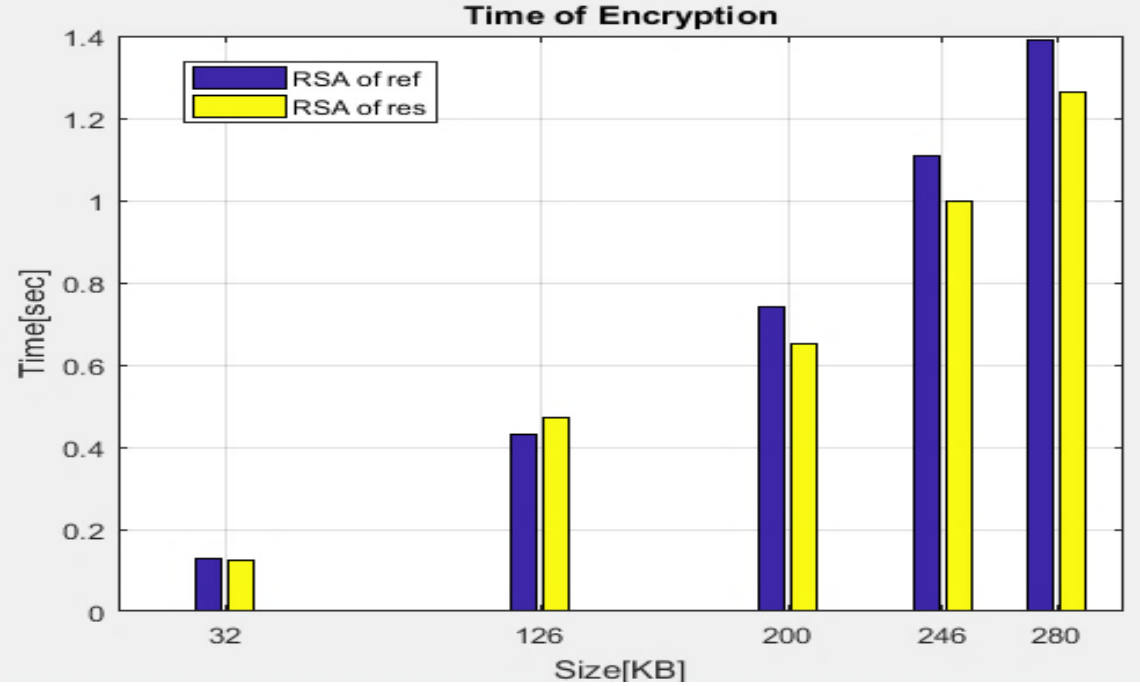


الشكل رقم (7): المقارنة في زمن التشفير لخوارزمية RSA المطورة عن RSA في المرجع [2] من أجل حماية البيانات

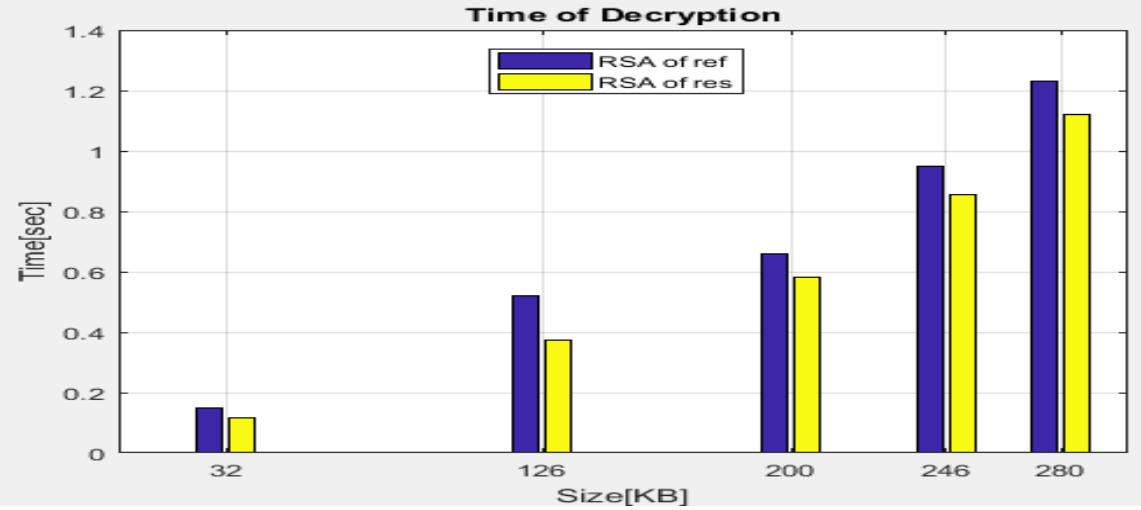


الشكل رقم (8) :المقارنة في زمن فك التشفير لخوارزمية RSA المطورة عن RSA في المرجع [2] من أجل حماية البيانات

إلا أنه نلاحظ انخفاض زمنهما لخوارزمية RSA الأصلية في هذا البحث عن زمنهما في المرجع كما يبينه الشكلان (9,10) ويرجع هذا الاختلاف بسبب اختلاف مواصفات الحاسوب المستخدم حيث استخدم الباحثان في المرجع [2] حاسوب بسرعة 2.34GHz وذاكرة 1GB، أما الحاسوب المستخدم في هذا البحث فهو عبارة عن حاسوب بسرعة 3.16 GHz وذاكرة 2GB، وتجدر الإشارة إلا أن خوارزمية RSA الأصلية المطبقة في الشكلان هي من أجل حماية البيانات فقط.



الشكل رقم (9): المقارنة في زمن التشفير لخوارزمية RSA الأصلية في بحثنا عن RSA في المرجع [2] من أجل حماية البيانات



الشكل رقم (10): المقارنة في زمن التشفير لخوارزمية RSA الأصلية في بحثنا عن RSA في المرجع [2] من أجل حماية البيانات

7_الاستنتاجات والتوصيات:

1) تؤمن خوارزمية RSA المطورة عملية حماية البيانات بزمني تشفير وفك تشفير أقل من الزمن الذي تحتاجه خوارزمية RSA الأصلية كما هو مبين في الشكلين (3,5)، وذلك لعدم إمكانية استرجاع البيانات المشفرة إلا بعد فك تشفير الكتلة الأولى للرسالة، وهذا الأمر يستوجب وجود المفتاح الخاص حصراً لدى مستلم الرسالة.
2) تزداد نسبة التحسين في زمني التشفير وفك التشفير في خوارزمية RSA المطورة عن خوارزمية RSA الأصلية مع ازدياد حجم البيانات، وذلك عند استخدامها من أجل حماية البيانات، مما يؤمن زيادة عدد البيانات المرسلة عبر الشبكة وبالتالي زيادة نسبة انسيابيتها.

3) تؤمن خوارزمية RSA المطورة عملية توثيق وحماية البيانات بزمني تشفير وفك تشفير أقل من الزمن الذي تتطلبه خوارزمية RSA الأصلية كما هو مبين في الشكلين (5,6)، وذلك لعدم إمكانية استرجاع البيانات المشفرة إلا بعد فك تشفير الكتلة الأولى للرسالة، وهذا الأمر يستوجب وجود المفتاح الخاص حصراً لدى مستلم الرسالة كما هو مبين بالعلاقتين (24,25)
4) تزداد انسيابية البيانات الموثقة والمحمية المرسلة عبر الشبكة عند استخدام خوارزمية RSA المطورة وذلك لزيادة نسبة التحسين في زمني التشفير وفك التشفير مع ازدياد حجم البيانات.
5) لا تتعلق نسبة التحسين في خوارزمية RSA المطورة بمواصفات الحاسوب المستخدم وذلك لتفوقها على خوارزمية RSA الأصلية عند تطبيقهما على نفس الحاسوب سواءاً لحماية البيانات فقط أو لتوثيق وحماية البيانات.
6) تزداد نسبة التحسين في خوارزمية RSA المطورة مع تحسين مواصفات الحاسوب المستخدم لتطبيقها لإن العمليات المنطقية (XOR) التي تحتاجها هذه الخوارزمية يتناقص زمن تنفيذها مع زيادة مواصفات الحاسوب.
7) تتميز خوارزمية RSA المطورة بزيادة نسبة التحسين الحاصلة في زمني التشفير وفك التشفير عند إجراء عملية توثيق البيانات وحمايتها، من أجل زمن التشفير وزمن فك التشفير، ومنه ازدياد نسبة التحسين بوجود التوثيق بمعدل الضعف تقريباً مما ينعكس ايجابياً على انسيابية البيانات الموثقة عبر الشبكة.
8) لا تختلف خوارزمية RSA المطورة عن خوارزمية RSA الأصلية في الآلية المستخدمة لتوليد المفاتيح، حيث تستخدمان نفس الآلية لذلك لم تتم مقارنة أزمنة توليد المفاتيح لكل منهما.

8- المراجع:

1. Ankita Verma, Paramita Guha, Sunita Mishra, 2016-**Comparative Study of Different Cryptographic Algorithms**, IJETTCS, Vol: 5, Issue 2.
2. Faiqa Maqsood, et al, 2017_" Cryptography: A Comparative Analysis for Modern Techniques", IJACSA, VOL.8, NO.6, PP445
3. Gary Kessler, 2017_"**An Overview of Cryptography**", Boca Raton: Auerbach Publications, pp_7.
4. Priya N. and Kannan M., 2017_ "**Comparative Study of RSA and Probabilistic Encryption**", IJECS, vol. 6, no. 1, pp. 19867 - 19871, January.
5. Pethe H. B. and Pande S. R., 2017_ "**Comparative Study and Analysis of Cryptographic Algorithms**", IJARCSMS, vol. 5, no. 1, pp. 48-56, 1 January.
6. Shivani Sharma, Yash Gupta, 2017- "**Study on Cryptography and Techniques**", IJSRCSEIT, Vol 2, Issue 1, ISSN: 2456-3307.
7. Shyam Kumar, 2015_"**Review on Network Security and Cryptography**", ITECES, Vol. 3, No. 1, pp_1-11.