مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (٨) العدد (٥) ٢٠٢٤
Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (8) No. (5) 2024

حماية البيانات الضخمة باستخدام خوارزمية تشفير هجينة

د. راغب عثمان طعمه*

أ.د. صلاح الدين نور الدين**

م. علياء عيسى داود***

(تاريخ الإيداع ٢٠٢٤/٢/٢٦ . قُبِل للنشر في ٢٠٢٤/٥/٢٦)

□ ملخّص□

تعد البيانات الضخمة (Bigdata) من أهم مجالات التكنولوجيا ، إذ يُطلق على أي حجم معقد ومتنوع وكبير من البيانات التي لا تستطيع تطبيقات المعالجة التقليدية معالجتها اسم البيانات الضخمة، ومع تزايد حجمها أصبحت الأنظمة السحابية مهمة لتخزينها حيث فشلت الأدوات التقليدية في تخزين هذا النوع من البيانات، ومع ذلك تبقى محط أنظار المخترقين إذا ما تم تخزينها دون تشفير.

لذلك يعد التشفير أحد الطرق المعتمدة لحماية البيانات، تم في هذا البحث دمج (خوارزمية التشفير المتماثل AES، خوارزمية التشفير غير المتماثل RSA، وخوارزمية التحكم بالوصول CPABE) للحصول على خوارزمية هجينة، ثم تطبيق الخوارزمية الهجينة على مجموعة من البيانات الضخمة بهدف تقييم أداء الخوارزمية الهجينة ومدى فعاليتها في تشفير البيانات الضخمة. أظهرت النتائج العملية تأثير الخوارزمية الهجينة المقترحة على زمن التشفير وزمن فك التشفير والانتاجية، كما تم مقارنة النتائج مع خوارزمية BigCrypto الناتجة عن دمج خوارزميات (AES, RSA)، وأظهرت النتائج تحسين في زمن التشفير وزمن فك التشفير ومعدل الإنتاجية.

تم تطبيق الخوارزمية المقترحة على عدة ملفات بأحجام (1GB ,1.5 GB, 2GB,3GB)، فمن أجل ملف بحجم 2GB حصلنا على زمن تشفير و زمن فك التشفير على التوالي min(28.24, 25.34) ، أما عند تطبيق خوارزمية BigCrypto على نفس البيانات السابقة، كان زمن التشفير وزمن فك التشفير على التوالي min(40.73 ,35.69).

الكلمات المفتاحية: البيانات الضخمة، خوارزمية AES، خوارزمية RSA، خوارزمية التشفير المعتمد على السمات –CP ABE، التشفير، فك التشفير، انتاجية التشفير، انتاجية فك التشفير .

*مدرس في قسم هندسة تكنولوجيا المعلومات – كلية هندسة تكنولوجيا المعلومات والاتصالات – جامعة طرطوس – سوريا.

** أستاذ مدرس في قسم العلوم الأساسية – كلية العلوم الأساسية – جامعة طرطوس – سوريا.

*** طالبة دراسات عليا (ماجستير) في قسم هندسة تكنولوجيا المعلومات – كلية هندسة تكنولوجيا المعلومات والاتصالات – جامعة طرطوس .

مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (٨) العدد (٥) ٢٠٢٤
Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (8) No. (5) 2024

Protecting big data with a hybrid encryption algorithm

Dr. Ragheb Othman Toemeh*
Prof. Salhadin Nouradin**
Eng. Alia Essa Daood ***

(Received 26/2/2024 . Accepted 26/5/2024)

□ ABSTRACT □

Big data is one of the most important areas of technology. Any complex, diverse, and large volume of data that traditional processing applications cannot process is called big data. As its size increases, cloud systems have become important for storing it, as traditional tools have failed to store this type of data. However, it remains vulnerable to hackers if it is stored without encryption.

Therefore, encryption is one of the methods adopted to protect data. In this research, (the AES symmetric encryption algorithm, the RSA asymmetric encryption algorithm, and the CPABE access control algorithm) were combined to obtain a hybrid algorithm, then the hybrid algorithm was applied to a set of huge data with the aim of evaluating the performance of the hybrid algorithm. And its effectiveness in encrypting huge data. The practical results showed the effect of the proposed hybrid algorithm on encryption time, decryption time, and throughput. The results were also compared with the BigCrypto algorithm resulting from merging the (RSA and AES) algorithms. The results showed an improvement in encryption time, decryption time, and throughput rate.

The proposed algorithm was applied to several files of sizes (1GB, 1.5 GB, 2GB, 3GB). For a file of size 2GB, we obtained the encryption time and decryption time, respectively, (28.24, 25.34) min. However, when applying the BigCrypto algorithm to the same previous data The encryption time and decryption time were (40.73, 35.69) min respectively.

Keywords: Big Data, Advanced Encryption Standard algorithm, RSA algorithm, CP-ABE feature-based encryption algorithm, encryption , decryption , encryption throughput, decryption throughput

* Teacher, Information Technology Engineering Department, Information and Communication Technology Engineering, Tartous University, Syria.

** Professor, Basic Sciences Department, Basic Sciences , Basic Sciences , Tartous University , Syria.

** Student Master, Information Technology Engineering Department, Information and Communication Technology Engineering, Tartous University, Syria

١. مقدمة

أدّى التطور التكنولوجي السريع وظهور انترنت الاشياء ومواقع التواصل الاجتماعي إلى توليد كمية هائلة من البيانات الضخمة التي تُخزن في الأنظمة السحابية، فعند الحديث عن السحابة تظهر العديد من المشكلات الأمنية بسبب مشاركة الموارد مع أطراف أخرى [1]، وبما أن أساليب الاختراق تتطور يوم بعد يوم بوتيرة متسارعة، كان لابد من إيجاد تقنيات جديدة لحماية البيانات أو تطوير تقنيات سابقة، وهذا ما يسعى إليه الباحثون والمهندسون حيث تعتبر تقنية التشفير من أهم تقنيات حماية البيانات [2].

تُعتبرُ أهمية الحصول على خوارزمية تشفير قوية ومرنة عند تشفير أحجام كبيرة من البيانات الدّافع الأوّل وراء عملنا، حيث تم في هذه الدراسة إيجاد خوارزمية هجينة مناسبة لتشفير البيانات الضخمة تحقق سرعة وأمان وذلك من خلال دمج خوارزميات التشفير المتماثل وخوارزميات التشفير غير المتماثل للاستفادة من ميزات التشفير المتماثل لتحقيق سرعة في التشفير وفك التشفير كما تم التغلب على قيود التشفير المتماثل من خلال الاعتماد على التشفير غير المتماثل [3]، حيث تقوم الخوارزمية الهجينة المقترحة بتشفير البيانات وفق خوارزمية AES 128 ثم حماية مفتاح التشفير بمستويي أمان ، المستوى الأول وفق خوارزمية CPABE والمستوى الثاني باستخدام خوارزمية RSA ثم حساب زمن التشفير وزمن فك التشفير ومعدل الإنتاجية.

في [1] تم تشفير وفك تشفير البيانات باستخدام خوارزميات RSA & AES والتحكم في الوصول المستند الى الدور (Role Based Access Control) RBAC لتأمين البيانات التي يتم تخزينها في السحابة، يقوم النظام المقترح على أربع أدوار للمستخدمين (المالك، المسؤول، المدير، المستخدمون) كما تم استخدام Experience Based Trust وهو أحد أنظمة إدارة الثقة التي تمكن من اتخاذ قرارات الثقة بناء على السلوك التاريخي للكيان. كما قدمت [2] شرح تفصيلي لخطوات خوارزمية AES والعمليات الرياضية والمصفوفات التي تستخدمها الخوارزمية لتوليد المفتاح السري وتشفير وفك تشفير البيانات .

وفي [3] تم اقتراح نهج BigCrypto الذي يستخدم التشفير المتماثل والتشفير غير المتماثل لتشفير وفك تشفير الملفات الكبيرة باستخدام خوارزميات AES 128 ثم تشفير مفتاح التشفير بخوارزمية RSA، تم تقييم سلوك التشفير وفك التشفير باستخدام برنامج pretty good privacy حيث تم إظهار الأداء من خلال تشغيل نفس الخوارزمية على ثلاث منصات مختلفة .

أيضاً اقترحت [4] إضافة أمان من ثلاث مستويات لحماية البيانات الموجودة في السحابة العامة باستخدام خوارزمية RSA & CP–ABE حساسية الوقت لتحقيق مستوى أفضل من الأمان والخصوصية للبيانات في السحابية، حيث أن التشفير ب CP–ABE يحقق التحكم في الوصول إلى البيانات إلا أنه لا يُحقق سرية البيانات، لذلك تم استخدام خوارزمية RSA لتأمين سرية البيانات قبل نقلها إلى السحابة، واعتمد في سياسة الوصول على سمتين فقط (الموقع–الجنس)، بحيث لا يمكن فك التشفير إلا في حال تطابق سمات المستخدم مع السمات المحددة من قبل مالك البيانات،

كما استخدمت [5] بيانات صغيرة الحجم، حيث تم تقسيم كل ملف مدخل الى مجموعة من الكتل وتشفير الكتلة الأولى بخوارزمية AES وتشفير مفتاح التشفير بخوارزمية CP-ABE ثم بخوارزمية RSA أما باقي الكتل يتم تشفيرها بخوارزمية RSA ، تم تقييم أداء الخوارزمية المقترحة من خلال دراسة زمن التشفير وزمن فك التشفير ومقارنة النتائج مع نتائج تنفيذ كل خوارزمية على حدا. وفي [6] تم تأمين البيانات الضخمة التي يتم تخزينها في HDFS (Hadoop Distributed File System) باستخدام خوارزمية هجينة تعتمد على خوارزميات CP-ABE & AES تم تقييم أداء الخوارزمية المقترحة بالاعتماد على زمن التشفير وزمن فك التشفير ومقارنة النتائج مع خوارزميات 3DES, DES , Blowfish .

٢. أهمية البحث وأهدافه:

هدفَ البحث إلى ضرورة حماية البيانات من خلال تشفيرها قبل تخزينها سحابياً وذلك بسبب مشكلات الأمان التي يحتويها التخزين السحابي، تكمن أهمية البحث في انشاء خوارزمية تشفير هجينة تحقق الأمان والخصوصية للبيانات وتأخذ زمن تشفير وزمن فك تشفير أقل ما يمكن، وبالتالي الحصول على إنتاجية عالية للتشفير وفك التشفير .

٣. طرائق البحث ومواده

تم إجراء الدراسة العملية باستخدام لغة البرمجة python بالاعتماد على برنامج PyCharm والمكتبات pycryptodome و times و charm_crypto لإنشاء الخوارزمية المقترحة وتطبيقها على مجموعة من البيانات الضخمة، ودراسة تأثير الخوارزمية المقترحة على زمن التشفير وزمن فك التشفير والإنتاجية باستخدام برنامج matlab ، كما تم الاعتماد على دراسات ومراجع علمية حديثة تختص بمجال أمن وتشفير البيانات.

٣-١ التشفير

يعد التشفير من التقنيات المهمة في حماية البيانات، ويقصد بالتشفير ترميز البيانات وفق تنسيق غير قابل للقراءة بحيث لا يتمكن الاشخاص غير المصرح لهم من الحصول على البيانات، يستخدم التشفير في عدة مجالات من أجل حماية الخصوصية، تأمين الاتصالات، تأمين الأنظمة والحفاظ على سرية البيانات.

ومع تزايد حجم البيانات وتطور أساليب الاختراق يكون التشفير أداة أساسية للحفاظ على الخصوصية والأمان في عالم مليء بالتهديدات السيبرانية مما دفع الباحثين لتطوير خوارزميات التشفير أو اقتراح خوارزميات جديدة يصعب على المخترقين كسرها[3].

٣-٢- أنواع خوارزميات التشفير

يوجد نوعان من خوارزميات التشفير المستخدمة للتشفير وفك التشفير، مثل خوارزميات التشفير المتماثل وخوارزميات التشفير غير المتماثل [2].

٣-٢-١ خوارزميات التشفير المتماثل

تعد خوارزميات التشفير المتماثل من أبسط تقنيات التشفير حيث يتم تشفير وفك تشفير البيانات بمفتاح سري واحد، مما يجعلها خوارزميات مرنة وسريعة لامتلاكها زمن تشفير وزمن فك تشفير قليل مقارنة بخوارزميات التشفير غير المتماثل، ومن خوارزميات التشفير المتماثل [3] Blowfish , DES, AES , RC4 , Camellia , Serpent .

٣-٢-٢ خوارزميات التشفير غير المتماثل

تعتبر خوارزميات التشفير غير المتماثل من أقوى خوارزميات التشفير حيث يتم تشفير البيانات بمفتاح تشفير عام من قبل المرسل، أما فك التشفير يتم باستخدام مفتاح فك تشفير خاص يمتلكه مستلم البيانات فقط، مما يجعلها خوارزميات قوية وصعبة الاختراق لكنها تستغرق زمن تشفير وزمن فك تشفير كبيرين بسبب مراحلها المعقدة في التشفير وتوليد المفاتيح السرية لذلك تكون خوارزميات غير مرنة عند استخدامها لتشفير البيانات الضخمة [3] .

٣-٢ خوارزمية معيار التشفير المتقدم AES (Advance Encryption Standard)

من أهم خوارزميات التشفير المتماثل التي نشرها المعهد الوطني للمعايير والتكنولوجيا NIST عام ٢٠٠٠ ، حيث تم استبدال خوارزمية DES ب خوارزمية AES بعد ظهور بعض نقاط الضعف في خوارزمية DES مثل (الأداء – الأمن – التكلفة)، و تتعامل AES مع ثلاث أحجام للمفاتيح ويختلف عدد الجولات باختلاف حجم المفتاح [2] .

- ١٠ جولات عندما يكون المفتاح ١٢٨ بت.
- ١٢ جولة عندما يكون حجم المفتاح ١٩٢ بت.
- ١٤ جولة عندما يكون حجم المفتاح ٢٥٦ بت.

كل جولة تتكون من ٤ خطوات باستثناء الجولة الأخيرة تتكون من ٣ خطوات وفي كل خطوة يتم تطبيق مجموعة من العمليات الرياضية على عدة مصفوفات واستخدام مفتاح موسع، هذه الخطوات هي:

(1-Sub Byte, 2-Shift row, 3-Mix Columns, 4-Add Round Key)

يتم تكرار الخطوات السابقة في جميع جولات الخوارزمية باستثناء الجولة الأخيرة يتم حذف الخطوة الثالثة من مراحل الخوارزمية.

- **مراحل عمل خوارزمية AES**

١. تحديد المفتاح: يتم اختيار مفتاح سري عشوائي باستخدام التابع العشوائي randomوبحجم يحدده عدد جولات الخوارزمية، في حال كان هناك ١٠ جولات للخوارزمية فإن حجم المفتاح يكون ١٢٨ بت.

٢. تقسيم البيانات: يتم تقسيم البيانات التي تحتاج للتشفير إلى كتل (بلوكات) معينة، حيث يجب أن تكون كل الكتل بحجم ثابت ١٢٨ بت.

٣. التشفير وفك التشفير: يتم تطبيق تحويلات رياضية على كل كتلة من البيانات باستخدام المفتاح السري، وهذه التحويلات تتضمن استخدام مصفوفات ثنائية محددة وعمليات مثل الاستبدالات والتحويلات الخطية والإضافات ثنائية الأعداد.

٤. التكرار: يتم تكرار خطوة التشفير/فك التشفير بتطبيق سلسلة من تحويلات المفتاح والبيانات.

٥. الإخراج: بعد الانتهاء من التشفير/فك التشفير لجميع الكتل، يتم الحصول على النص المشفر/الأصلي.

٣-٣ خوارزمية RSA (Rivest Shamir Adlemen)

تعد خوارزمية RSA من أهم خوارزميات التشفير الغير متماثل المستخدمة في التوقيع الرقمي digital signatureو التشفير encryption، تمت تسمية هذه الخوارزمية نسبةً إلى مخترعيها (Ron Rivest – Adi Shamir – Leonard Adleman)، تستخدم هذه الخوارزمية مفتاح عام لتشفير البيانات ومفتاح خاص لفك التشفير مما يجعلها قوية في تأمين البيانات ومستخدمة على نطاق واسع في توقيع الرسائل الرقمية وتأمين الاتصالات الالكترونية [3] [8].

- **مراحل عمل خوارزمية RSA [9]**

١. اختيار عددين أوليين كبيرين ومختلفين بشكل عشوائي، يُعرفان بـ "العاملين الأوليين" p و q.

٢. حساب حاصل ضرب العاملين الأوليين: N = p * q. هذه القيمة تستخدم لحساب المفاتيح العامة والخاصة.

٣. حساب دالة أويلر لـ N، تُستخدم هذه الدالة في توليد المفتاح العام والخاص وتعطى دالة اويلر بالعلاقة التالية:

$$\varphi(n) = (p - 1) * (q - 1) \qquad (١)$$

٤. اختيار عدد صحيح عشوائي e، حيث يجب أن يكون e أقل من دالة أويلر لـ N ، حيث e و φ(n) أوليان فيما بينهما . وقيمة e الأكثر شيوعاً هي (2^16 + 1) = 65537.

٥. حساب المفتاح العام: هو المفتاح الذي يتكون من الزوج (e, N)، حيث يُستخدم لتشفير الرسائل أو البيانات.

٦. حساب المفتاح الخاص: هو المفتاح الذي يتكون من الزوج (d, N)، حيث يتم حساب قيمة d باستخدام خوارزمية (Extended Euclidean Algorithm) وفق العلاقة $d = e^{-1} \pmod{\emptyset(n)}$.

٧. التشفير وفك التشفير: يتم استخدام المفتاح العام لتشفير البيانات أو التوقيع الرقمي، ويتم استخدام المفتاح الخاص لفك التشفير أو التحقق من التوقيع الرقمي.

٣-٤ خوارزمية التشفير المستندة الى السمات CP-ABE (Ciphertext-Policy Attribute_Based_Encryption)

تعد هذه الخوارزمية نوع من تقنيات التشفير السماتية (Attribute–Based Encryption)، التي تتيح للمستخدمين تشفير وفك تشفير البيانات باستخدام مجموعة من السمات (attributes) بدلاً من استخدام المفاتيح التقليدية ، حيث يتم تعريف سمات تصف المستخدمين ، و تحويل البيانات إلى شكل مشفر، و تحديد سياسة الوصول (access policy) باستخدام هذه السمات، ثم تحويل المفتاح السري إلى مفتاح عام ، مما يمكن الأطراف التي تحمل السمات المناسبة من استخدام هذا المفتاح العام لفك تشفير البيانات المشفرة و الوصول إليها ، تسمح خوارزمية CP–ABE بتنفيذ

قواعد الولوج المعقدة بناءً على سمات المستخدمين ، وتوفر مرونة وسهولة في تطبيق نماذج التحكم في الوصول الى البيانات[6] [7].

مراحل عمل خوارزمية CP-ABE

١. تحديد السمات والمفاتيح: تحديد السمات المتاحة وتعيين المفاتيح المناسبة للأطراف المختلفة.

٢. توليد المفتاح الرئيسي: توليد المفتاح الرئيسي الذي يستخدم لإنشاء المفاتيح العامة والخاصة للمستخدمين.

٣. توليد المفاتيح العامة والخاصة: يتم توليد المفاتيح العامة للمستندات والمفاتيح الخاصة للمستخدمين.

٤. تشفير وفك تشفير البيانات: تشفير البيانات باستخدام المفاتيح العامة وفك تشفيرها باستخدام المفاتيح الخاصة للمستخدمين المصرح لهم.

٤. الحصول على البيانات الضخمة

تم الحصول على البيانات بأحجام مختلفة 1GB , 1.5GB , 2GB , 3GB , 4GB ,5GB من عدة مواقع وذلك لصعوبة وجود البيانات الضخمة في موقع ويب واحد.

٥. الدراسة العملية ومناقشة النتائج

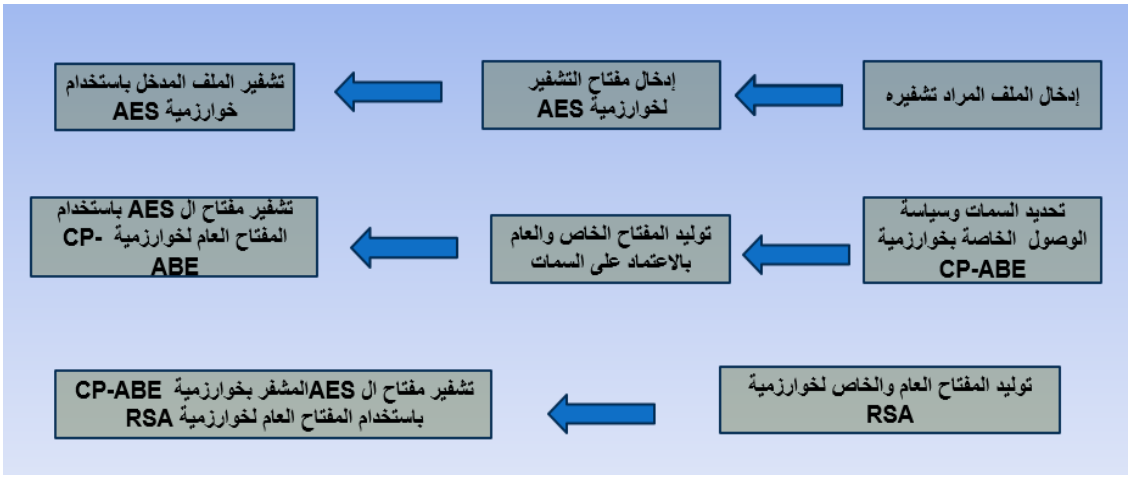
تم دمج الخوارزميات (AES ,RSA ,CP-ABE) للحصول على خوارزمية تشفير هجينة ، بهدف الاستفادة من ميزات كلاً منها، حيث :

- تم استخدام خوارزمية AES لتشفير البيانات الضخمة بسبب سرعتها في تشفير وفك تشفير البيانات، بالرغم من قوة خوارزمية RSA إلا أنه لا يُمكن استخدامها لتشفير البيانات الضخمة لذلك تم استخدمها لتشفير المفتاح السري فقط.
- تم تشفير مفتاح ال AES باستخدام خوارزمية CP-ABE، من أجل التحكم بالوصول.
- تم تشفير مفتاح AES المشفر بخوارزمية CP-ABE باستخدام خوارزمية RSA، من أجل تحقيق الموثوقية وإدارة المفاتيح.

٥-١ ألية عمل الخوارزمية المقترحة

تهدف الخوارزمية الهجينة المقترحة إلى تحسين زمن التشفير وزمن فك التشفير دون التأثير على قوة التشفير، لذلك تم دمج الخوارزميات الثلاثة في خوارزمية هجينة واحدة لتحقيق المرونة والسرعة التي توفرها خوارزمية AESو القوة التي تعطيها خوارزميةRSA والتحكم بالوصول الذي تؤمنه خوارزمية CPABE، وفق الخطوات الموضحة في مبدأ التشفير ومبدأ فك التشفير.

٥-١-١ مبدأ التشفير



الشكل (١) – المخطط الصندوقي لمبدأ التشفير بالخوارزمية المقترحة

تمت برمجة الخوارزمية من خلال دمج الخوارزميات الثلاث (AES , CPABE , RSA)، لتعمل وفق الخطوات التالية :

الخطوة ١: إدخال الملف المراد تشفيره، في هذا البحث تم إدخال ملفات بأحجام مختلفة:

.1GB, 1.5GB, 2GB, 3GB, 4GB ,5GB

الخطوة ٢: إدخال مفتاح التشفير لخوارزمية AES، بطول ١٦ بايت، لأنه تم اختيار خوارزمية AES 128 كخوارزمية أساسية لتشفير البيانات، من اجل تحقيق سرعة للتشفير وفك التشفير.

الخطوة ٣: تشفير الملف الذي تم إدخاله باستخدام خوارزمية AES وفق الخطوات المذكورة سابقاً في مراحل عمل خوارزمية AES.

الخطوة ٤: تحديد السمات للأشخاص الذين يمكنهم الوصول الى مفتاح وتحديد سياسة الوصول بالاعتماد على السمات التي تم تحديدها وهنا تم اختيار سمتين (الدرجة العلمية للمستخدم – القسم الذي ينتمي اليه المستخدم) ، وذلك وفقاً لنوع البيانات التي حصلنا عليها وهي بيانات علمية خاصة بهندسة تكنولوجيا المعلومات، تم وضع سياسة الوصول بناءً على السمات حيث تم تحديد الدرجة العلمية للمستخدم (ماجستير أو دكتوراه) و القسم (تكنولوجيا المعلومات) .

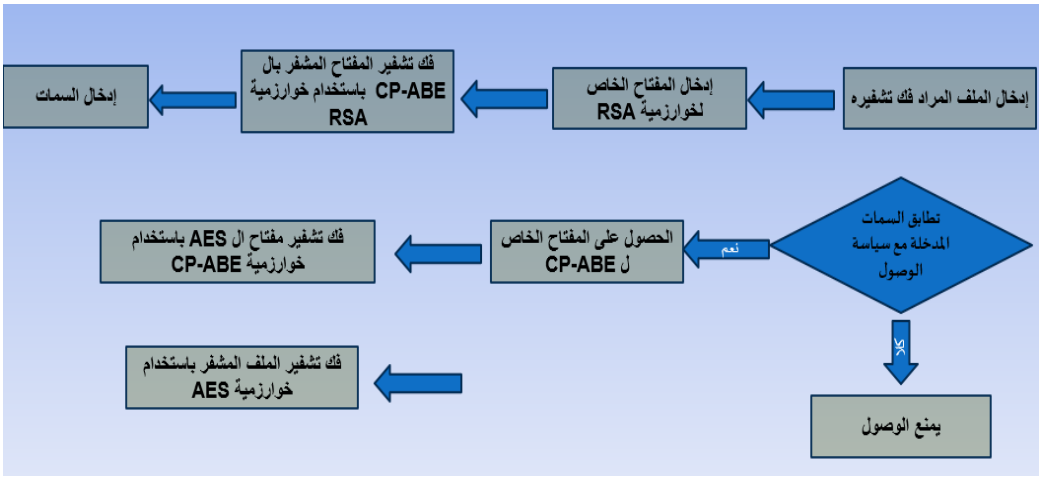
الخطوة ٥: توليد المفتاح العام والخاص اعتماداً على السمات التي تم تحديدها وفق خطوات خوارزمية CPABE الموضحة في الفقرات السابقة .

الخطوة ٦: تشفير مفتاح AES باستخدام المفتاح العام لخوارزمية CP–ABE.

الخطوة ٧: توليد المفتاح العام والخاص لخوارزمية RSA.

الخطوة ٨: تشفير مفتاح AES المشفر بخوارزمية CP–ABE باستخدام خوارزمية RSA.

٥-١-٢ مبدأ فك التشفير



الشكل (٢) –المخطط الصندوقي لمبدأ فك التشفير بالخوارزمية المقترحة

تقوم الخوارزمية المقترحة بفك تشفير البيانات التي تم تشفيرها سابقاً وفق الخطوات التالية:

الخطوة ١: إدخال الملف المراد فك تشفيره، تم إدخال ملفات مشفرة بالخوارزمية المقترحة بأحجام مختلفة 1GB ,5GB, 4GB , 3GB , 2GB 1.5GB .,

الخطوة ٢: إدخال (مفتاح فك التشفير) المفتاح الخاص لخوارزمية RSA.

الخطوة ٣: فك تشفير المفتاح المشفر بخوارزمية CPABEباستخدام المفتاح الخاص لخوارزمية RSA.

الخطوة ٤: يقوم المستخدم بإدخال السمات المطلوبة.

الخطوة ٥: في حال لم تكن السمات المدخلة متطابقة مع السمات التي تم استخدامها وفق سياسة الوصول التي تم تحديدها في مرحلة التشفير يمنع المستخدم من الوصل الى البيانات وإلا يحصل المستخدم على (مفتاح فك التشفير) المفتاح الخاص لخوارزمية CPABE.

الخطوة ٦: بعد الحصول على المفتاح الخاص لخوارزمية CP–ABE يتم فك تشفير مفتاح AES المشفر بخوارزمية CP–ABE .

الخطوة ٧: بعد الحصول على مفتاح فك التشفير يتم فك تشفير الملف والحصول على البيانات الأصلية.

٥-٢ مقاييس الأداء:

تم الاعتماد على زمن التشفير وزمن فك التشفير ومعدل الإنتاجية كمقاييس لتقييم أداء الخوارزمية، كما تمت مقارنة قيم هذه البارمترات الناتجة عن تشفير البيانات باستخدام الخوارزمية المقترحة مع قيم البارمترات الناتجة عن تشفير نفس البيانات باستخدام الخوارزمية الهجينة (BigCrypto) المقترحة في الدراسة الثالثة [٣]، والتي تقوم بتشفير البيانات بخوارزمية AES 256 ثم تقوم بتشفير مفتاح التشفير بخوارزمية RSA، يوضح الجدول (١) مقارنة بين الخوارزمية المقترحة وخوارزمية BigCrypto.

الجدول(١): الخوارزمية المقترحة مقارنةً بخوارزمية BigCrypto

	الخوارزمية المقترحة	BigCrypto
تشفير البيانات	AES	AES
مفتاح AES	128 bit	256 bit
جولات AES	١٠ جولات	١٤ جولة
مستوى الأمان الأول لمفتاح التشفير	CPABE	RSA
مستوى الأمان الثاني لمفتاح التشفير	RSA	لا يوجد
طول مفتاح RSA	2048 bit	2048 bit
السمات	الدرجة العلمية – التخصص	لا يوجد

زمن التشفير

يعد زمن التشفير أحد أهم البارمترات المتعلقة بأداء خوارزمية التشفير فكلما قل زمن التشفير كانت الخوارزمية مرنة وسريعة وهذا ما تتطلبه البيانات الضخمة.

تمت برمجة الخوارزمية المقترحة باستخدام برنامج PyCharm ثم حساب زمن التشفير باستخدام مكتبة time التي تحوي الدالة time () يتم استدعاؤها في بداية تنفيذ الخوارزمية المقترحة لحساب زمن بداية التشفير ويتم استدعاؤها في نهاية تنفيذ الخوارزمية لحساب الزمن الذي ينتهي فيه تنفيذ الخوارزمية

Encryption time = end time for encryption − start time for encryption

على سبيل المثال: إذا تم تشغيل الخوارزمية في الساعة ١:٠٠:٠٠ وانتهى تنفيذ الخوارزمية في الساعة ١:٣٠:٣٠ يكون زمن التشفير:

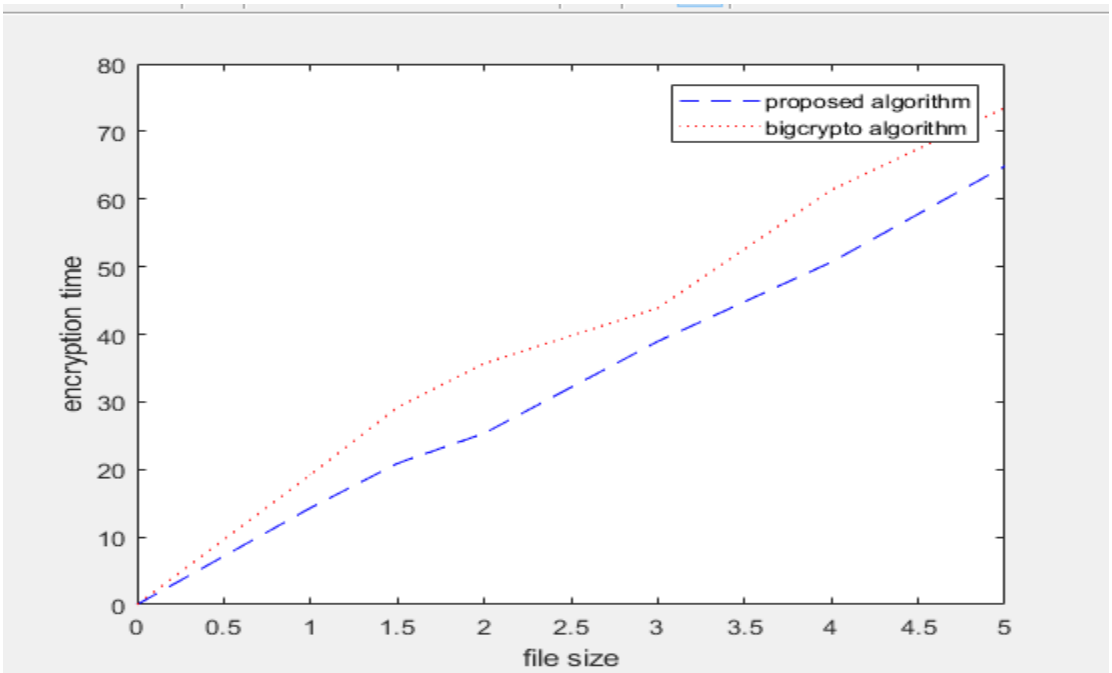
Encryption time = ١:٠٠:٠٠ − ١:٣٠:٣٠ = ٠٠:٣٠:٣٠

ثم يتم تحويل الزمن الناتج إلى وحدة الثانية فيكون زمن التشفير 1860 second

هذا الزمن يتضمن زمن توليد المفاتيح بالإضافة إلى زمن تشفير البيانات، يظهر الجدول (2) أزمنة التشفير لأحجام مختلفة من البيانات بعد تطبيق الخوارزمية المقترحة، مقارنةً بأزمنة تشفير ذات البيانات بخوارزمية bigcrypto.

جدول (2) – زمن التشفير بعد تنفيذ الخوارزمية المقترحة مقارنةً بزمن التشفير لخوارزمية BigCrypto

حجم الملف GB	1	1.5	2	3	4	5
زمن التشفير (proposed) Min	14.32	20.90	25.34	38.95	50.68	64.89
زمن التشفير (bigcrypto) Min	19.23	٢٩,١٩	35.69	43.86	61.35	73.51



الشكل (٣) – زمن التشفير بعد تنفيذ الخوارزمية المقترحة مقارنةً بزمن التشفير لخوارزمية BigCrypto

يوضح الشكل (٣) أن الخوارزمية المقترحة تفوقت على خوارزمية BigCrypto ، وبمقارنة الخطين البيانيين لزمن التشفير بالخوارزمية المقترحة و زمن التشفير بخوارزمية BigCrypto نجد أن الخوارزمية المقترحة أسرع وهذا ما يجعلها مناسبة لتشفير البيانات الضخمة مع ملاحظة أن زمن التشفير يزداد بزيادة حجم البيانات المراد تشفيرها .

إنتاجية التشفير

يتم حساب إنتاجية التشفير وفق القانون:

$$[5]\ \text{Encryption Throughput} = \frac{\sum file\ size}{\sum encrytion\ times} \qquad (2)$$

$\sum file\ size$:مجموع أحجام الملفات المراد تشفيرها بواحدة MB.

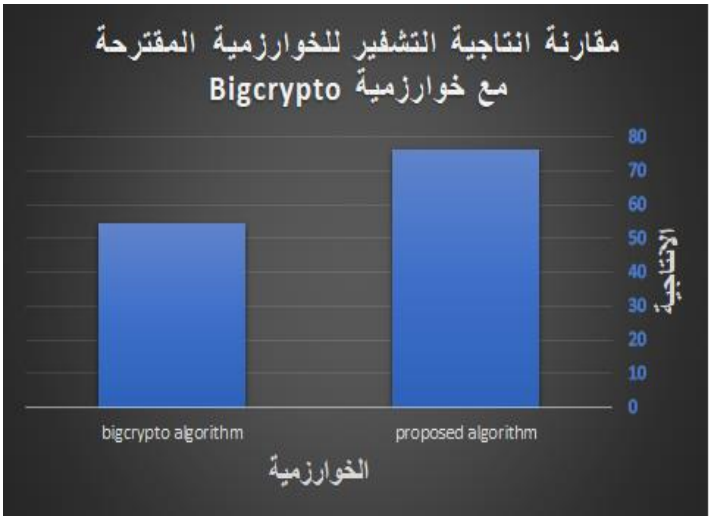
$\sum encrytion\ times$: مجموع أزمنة التشفير للملفات التي تم تشفيرها بالخوارزمية المقترحة بواحدة *min*.

بتعويض القيم (من الجدول 2) التي حصلنا عليها بعد تنفيذ الخوارزمية المقترحة في المعادلة (2) نحصل على معدل إنتاجية 77.17 MB | min

$$\text{Encryption Throughput} = \frac{\sum file\ size}{\sum encrytion\ times} = \frac{1024*(1+1.5+2+3)}{14.32+20.90+25.34+38.95} = 77.17 \text{ MB I min}$$

بتعويض القيم (من الجدول 2) التي حصلنا عليها بعد تنفيذ خوارزمية BigCrypto في المعادلة (2) نحصل على معدل إنتاجية 54.78 MB | min

$$\text{Encryption Throughput (Bigcrypto algorithm)} = \frac{1024*(1+1.5+2)}{19.23+29.19+35.69} = 54.78 \text{ MB I min}$$



الشكل (4)– مقارنة إنتاجية التشفير للخوارزمية المقترحة مع انتاجية التشفير لخوارزمية BigCrypto

يوضح الشكل (٤) أن معدل إنتاجية التشفير بالخوارزمية المقترحة أعلى من معدل الإنتاجية بخوارزمية BigCrypto، حيث يعد معدل الإنتاجية مقياس لكفاءة خوارزمية التشفير و قدرتها على تشفير البيانات بسرعة عالية، مما يعزز كفاءة عمل النظام ويحمي البيانات بشكل أفضل، وبالتالي كلما كان معدل الإنتاجية أقل كانت الخوارزمية تستهلك وقتاً أطول وتتطلب موارد أكبر مما يؤثر سلباً على الأداء بشكل عام، والعكس صحيح .

زمن فك التشفير

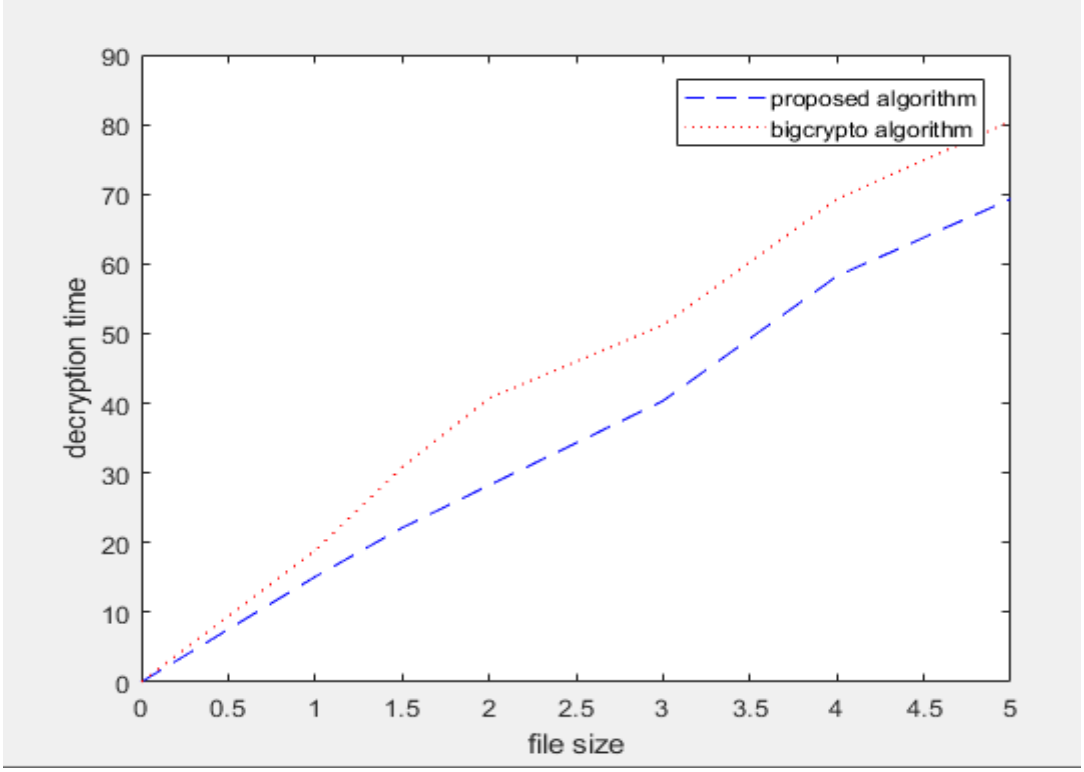
كما في حساب زمن التشفير تم حساب زمن فك التشفير باستخدام مكتبة time التي تحوي الدالة () time يتم استدعاؤها في بداية تنفيذ الخوارزمية المقترحة لحساب زمن بداية فك التشفير ويتم استدعاؤها في نهاية تنفيذ الخوارزمية لحساب الزمن الذي ينتهي فيه تنفيذ الخوارزمية.

Decryption time = end time for decryption − start time for decryption

يوضح الجدول (٣) زمن فك تشفير البيانات باستخدام الخوارزمية المقترحة و زمن فك تشفير البيانات باستخدام خوارزمية BigCrypto ، فعند فك تشفير ملف بحجم 1GB بالخوارزمية المقترحة يكون زمن فك التشفير ١٥,١١ min، أما عند فك تشفير هذا الملف بخوارزمية BigCrypto يكون زمن فك التشفير مساوي ١٨,٨٨ min ، وبالتالي تكون الخوارزمية المقترحة مناسبة لفك تشفير البيانات الضخمة حيث أن زمن فك تشفيرها أقل من زمن فك التشفير بخوارزمية BigCrypto .

جدول (٣) – زمن فك التشفير بعد تنفيذ الخوارزمية المقترحة زمن التشفير مقارنةً بزمن فك التشفير لخوارزمية BigCrypto

حجم الملف المشفر GB	1	1.5	2	3	4	5
زمن فك التشفير (proposed) min	15.11	22.13	28.24	40.35	58.24	69.34
زمن فك التشفير (bigcrypto) min	18.88	30.80	40.73	51.22	69.30	80.47



الشكل (5) – زمن فك التشفير بعد تنفيذ الخوارزمية المقترحة زمن التشفير مقارنةً بزمن فك التشفير لخوارزمية BigCrypto

الشكل (5) بمقارنة الخطين البيانيين لزمن فك التشفير بالخوارزمية المقترحة وزمن فك التشفير بخوارزمية BigCrypto نجد أن الخوارزمية المقترحة أسرع وهذا ما يجعلها مناسبة لفك تشفير البيانات الضخمة .

إنتاجية فك التشفير

يتم حساب إنتاجية فك التشفير وفق القانون :

$$\text{Decryption Throughput} = \frac{\sum file\ size}{\sum decrytion\ times} \quad (3)$$ [5]

$\sum file\ size$:مجموع أحجام الملفات المراد فك تشفيرها بواحدة MB .

$\sum decrytion\ times$: مجموع أزمنة فك التشفير للملفات التي تم فك تشفيرها بالخوارزمية المقترحة بواحدة min.

بتعويض القيم (من الجدول 3) الناتجة عن تنفيذ الخوارزمية المقترحة على مجموعة البيانات في المعادلة (٣) نحصل على معدل إنتاجية لفك التشفير 72.56 MB I min

$$\text{Decryption Throughput} = \frac{\sum file\ size}{\sum decrytion\ times} = \frac{1024*(1+1.5+2+3)}{15.11+22.13+28.24+40.35} = 72.56 \text{ MB I min}$$

بتعويض القيم (من الجدول ٣) الناتجة عن تنفيذ الخوارزمية BigCrypto على مجموعة البيانات في المعادلة (٣) نحصل على معدل إنتاجية لفك التشفير 50.96MB I min

$$\text{Decryption Throughput (Bigcrypto algorithm)} = \frac{1024*(1+1.5+2)}{18.88+30.80+40.73} = 50.96 \text{ MB I min}$$

الشكل (٦)– مقارنة إنتاجية فك التشفير للخوارزمية المقترحة مع إنتاجية فك التشفير لخوارزمية BigCrypto

يمثل الشكل (٦) إنتاجية فك التشفير عند تطبيق الخوارزمية المقترحة مقارنةً خوارزمية BigCrypto حيث أن إنتاجية فك التشفير عند تطبيق الخوارزمية المقترحة أكبر مقارنةً بخوارزمية bigcrypto وذلك لأن خوارزمية BigCrypto تستخدم مفتاح تشفير بطول ٢٥٦ بت بينما الخوارزمية المقترحة تستخدم مفتاح بطول ١٢٨ بت، بالإضافة أننا حققنا أمان أكبر لأنه تم تأمين مفتاح التشفير بمستويي أمان ، بينما في خوارزمية BigCrypto تم تأمين مفتاح التشفير بمستوى أمان واحد.

٦. الاستنتاجات

الشيء المميز في هذه الخوارزمية المقترحة بأنها مناسبة لتشفير البيانات الضخمة، حيث تم الاستفادة من ميزات خوارزمية التشفير المتماثل AES و التشفير غير المتماثل RSA والتحكم بالوصول CPABE من خلال دمجها في خوارزمية هجينة واحدة، تم الحصول على أزمنة تشفير و أزمنة فك تشفير مناسبة عند تشفير هذا النوع من البيانات وبالتالي الحصول على إنتاجية عالية بسبب تشفير البيانات بخوارزمية AES و حماية مفتاح التشفير بمستويين من الأمان من خلال تشفير المفتاح بخوارزمية CPABE ثم تشفيره بخوارزمية RSA .

٧. العمل المستقبلي

العمل على دمج خوارزميات تشفير أخرى مع الخوارزمية المقترحة أو إضافة تقنيات أمان أخرى للخوارزمية المقترحة بهدف زيادة الإنتاجية مع الحفاظ على قوة الخوارزمية ومرونتها عند تشفير البيانات الضخمة.

٨. المراجع

[1] Bokefode Jayant, D., Ubale Swapnaja, A., Pingale Subhash, V., Karande Kailash, J., & Apate Sulabha, S. (2015). Developing secure cloud storage system by applying AES and RSA cryptography algorithms with role based access control model. International Journal of Computer Applications, 975, 8887.

[2] Abdullah, A. M. (2017). Advanced encryption standard (AES) algorithm to encrypt and decrypt data. Cryptography and Network Security, 16(1), 11.

[3] Al Mamun, A., Salah, K., Al-Maadeed, S., & Sheltami, T. R. (2017, May). BigCrypt for big data encryption. In 2017 Fourth International Conference on Software Defined Systems (SDS) (pp. 93-99). IEEE.

[4] Radhakrishnan, P., Panicker, N. N., John, S. S., & SB, N. P. V. D. (2019). Attribute and Time Factors Combined CP-ABE and RSA based Access Control Scheme for Public Cloud. International Journal of Information, 8(2).

[5] Mahrousa, Z., Rahhal, M., & Alzin, N. (2021). Hybrid Secure Model for Securing Data in The Cloud Computing System by Combining RSA, AES and CP-ABE: نموذج هجين آمن لحماية البيانات في الحوسبة السحابية بدمج CP-ABE و AES و RSA. مجلة العلوم الهندسية و تكنولوجيا المعلومات، ٥(٤)، ٨٠-٥٩.

[6] Mohanraj, T., & Santhosh, R. (2022). Hybrid encryption algorithm for big data security in the Hadoop distributed file system. Computer Assisted Methods in Engineering and Science, 29(1–2), 33-48.

[7] Kumar, P. P., Kumar, P. S., & Alphonse, P. J. A. (2017, December). An efficient ciphertext policy-attribute based encryption for big data access control in cloud computing. In 2017 Ninth International Conference on Advanced Computing (ICoAC) (pp. 114-120). IEEE.

[8] Bhargavi, I., Veeraiah, D., & Maruthi Padmaja, T. (2017). Securing BIG data: a comparative study across RSA, AES, DES, EC and ECDH. In Computer Communication, Networking and Internet Security: Proceedings of IC3T 2016 (pp. 355-362). Springer Singapore.

[9] Satapathy, S. C., Bhateja, V., Raju, K. S., & Janakiramaiah, B. (2016). Computer Communication, Networking and Internet Security Proceedings of IC3T 2016. Proceedings of IC3T, 1.

[10] Li, S., & Gao, J. (2016). Security and privacy for big data. Big Data Concepts, Theories, and Applications, 281-313.