مجلة جامعة دمشق للعلوم الهندسية
المجلد 40- العدد 1 (2024) 165-200

Damascus university Journal for engineering sciences
Vol 40 No.1 (2024): 165-200

خوارزميّة AES المُعدَّلة من أجل أجهزة أندرويد مُقيَّدة الموارد

جعفر سلطان[1]، د. محمد إياد الخياط[2]

[1]طالب دراسات عليا في قسم النُّظُم والشبكات الحاسوبيّة – كليّة الهندسة المعلوماتيّة – جامعة دمشق.

[2]مُدرّس في قسم النُّظُم والشبكات الحاسوبيّة – كليّة الهندسة المعلوماتيّة – جامعة دمشق.

المُلخَّص

تُوفِّر نظم تشفير التخزين في أندرويد آليّةً فعّالةً لحماية البيانات المُخزَّنة على الجهاز من الوصول غير المُصرَّح به؛ وهي تعتمد بشكلٍ أساسيٍّ على خوارزميّة التشفير AES. من أجل المنصّات التي لا تدعم معماريّات وحدات المُعالَجة المركزيّة فيها تسريع AES؛ يدعم أندرويد طريقة التشفير Adiantum التي تُوفِّر تشفيراً قويّاً مع عبءٍ مُضافٍ قليل. يُعتبَر أداء فكّ التشفير في نظم تشفير التخزين الأكثر أهميّةً نظراً لأنّ عمليّات القراءة تكون أكثر تكراراً من عمليّات الكتابة، وهي تُؤثِّر بشكلٍ عامٍّ على وقت الاستجابة الذي يتوقَّعه المُستخدِم. في الحقيقة؛ إنّ تصميم تحويل خلط الأعمدة/ خلط الأعمدة العكسيّ في الخوارزميّة AES يجعل زمن فكّ التشفير يزيد عن ضعف زمن التشفير؛ ممّا يُؤثِّر على أداء نظام التشفير. يعود السبب في ذلك إلى قِيَم مُعامِلات كثير الحدود المُستخدَمة في مصفوفة MDS، تُظهِر الدراسة المرجعيّة وجود بعض الأوراق البحثية حول تحسين أداء الخوارزميّة AES ولكنّ معظمها لا تُوفِّر مُقايَضةً جيدةً بين تحسُّن الأداء وأمان الخوارزميّة. قُمنا بتعديل تحويل خلط الأعمدة بحيث يستخدم مصفوفةً قريبةً من MDS دائريّةً ثنائيّةً مَعكوسةً ذاتيّاً، وتعويض خصائص النشر المُنخفِضة عن طريق زيادة عدد الجولات بمِقدار جولتين؛ بما يضمن هامش أمانٍ من أربع جولاتٍ مُناسِباً ضدّ الهجمات المعروفة، تُوفِّر الخوارزميّة المُعدَّلة BMC-AES أداءً مُحسَّناً وثابتاً في عمليّات التشفير وفكّ التشفير، وكلفةً أقلّ في التنجيزات البرمجيّة والعتاديّة، وهي تُحافِظ على الميزات المُفضَّلة للخوارزميّة AES. تُشير نتائج الاختبارات العمليّة إلى تحسُّن مردود التشفير بنسبة 22% وتحسُّن مردود فكّ التشفير

تاريخ الإيداع: 2022/6/29
تاريخ القبول: 2022/9/13

بنسبة 93% على منصّة أندرويد التي تدعم تسريع AES. وبالمُقابِل؛ يتحسَّن مَردود التشفير بنسبة 53% ويتحسَّن مَردود فكّ التشفير بنسبة 240%على منصّات أندرويد التي لا تدعم تسريع AES. لذلك تُعتبَر الخوارزميّة BMC-AES أكثر مُلاءمَةً للاستخدام في أجهزة أندرويد بشكلٍ مُستقِلٍّ عن تَوافُر دعم تعليمات AES.

الكلمات المفتاحيّة: خوارزميّة AES، خوارزميّة BMC-AES، تحويل MixColumns، InvMixColumns، BinMixColumns، رقم الفرع، طريقة التشفير Adiantum، أندرويد، نُظُم تشفير التخزين.

Modified AES Algorithm for Resource-constrained Android Devices

Jafar Sultan[1], Dr. Mhd. Iyad Alkhayat[2]

[1]Master student in the Department of Computer Systems and Networking - Faculty of Information Technology Engineering - Damascus University.

[2]Lecturer in the Department of Computer Systems and Networking - Faculty of Information Technology Engineering - Damascus University.

Received: 29/6/2022
Accepted: 13/9/2022

Abstract

Storage encryption systems in Android provide an effective mechanism to protect the data stored on the device from unauthorized access, and it is mainly based on the AES encryption algorithm. For platforms where CPU architectures do not support AES acceleration; Android supports Adiantum encryption method which provides strong encryption with little added overhead. For storage encryption systems, decryption performance matters most because reads are more frequent than writes, and they generally affect user-predicted latency. In fact, the MixColumns/ InvMixColumns transformation design in AES algorithm makes the decryption time more than twice the encryption time; which affects the performance of the encryption system. This is due to the coefficients of the polynomial used in the MDS matrix. The reference study shows that there are some research papers on improving performance of the AES algorithm but most of them do not provide a good trade-off between performance improvement and algorithm security. We modified the MixColumns transformation to use an involution binary circulant Almost MDS matrix, and compensate for low diffusion characteristics by increasing the number of rounds by two, which ensures a proper four-round margin of safety against known attacks. The modified algorithm BMC-AES provides improved and constant performance in encryption and decryption operations, lowers the cost in hardware and software implementations, and preserves the preferred features of the AES algorithm. The results of practical tests indicate that the encryption throughput is improved by 22% and the decryption throughput is improved by 93% on the Android platform that supports AES acceleration. On the other hand, the encryption throughput is improved by 53% and the decryption throughput is improved by 240% on Android platforms that do not support AES acceleration, so the BMC-AES algorithm is more suitable for use in Android devices independently of the availability of AES instruction support.

Keywords: AES algorithm, BMC-AES algorithm, MixColumns transformation, InvMixColumns, BinMixcolumns, branch number, Adiantum encryption method, Android, storage encryption systems.

1- المُقدِّمة:

أصبحت الهواتف المحمولة جزءاً مُتكامِلاً مع نمط حياتنا اليوميّ، ليس لكونها تُوفِّر الحركيّة العالية وسهولة الاتصال فحسب؛ بل لكونها تحوَّلت إلى مخزنٍ للمعلومات الشخصيّة. يُعدُّ نظام التشغيل أندرويد الأكثر انتشاراً في سوق الأجهزة المحمولة، فهو يُهيمِن على سوق أنظمة تشغيل الأجهزة المحمولة في جميع أنحاء العالم بحِصّةٍ تصل إلى 71.45% [12]، وهو ثاني نظام من حيث الانتشار في سوق أنظمة تشغيل الأجهزة اللوحيّة في جميع أنحاء العالم بحِصّةٍ تصل إلى 46.3% [13]. يعود السبب وراء ذلك إلى الدعم العالي وتطوير ميزاته بصورةٍ مُستمرّةٍ وتَوفُّر متجر التطبيقات الرسميّ الذي يحوي ملايين التطبيقات المُتاحة بشكلٍ مجّانيّ، ولذلك فهو يحوز على النصيب الأكبر من اهتمام المُخترِقين ومُطوِّري البرامج الخبيثة التي تهدف إلى سرقة البيانات الشخصيّة للمُستخدِمين.

تُوفِّر نُظُم تشفير التخزين في أندرويد آليّةً فعّالةً لحماية البيانات المُخزَّنة على الجهاز من الوصول غير المُصرَّح به حتّى في حال سرقة أو ضياع الجهاز المُشفَّر، وهي تعتمد على التشفير التناظريّ الذي يستخدم المفاتيح المُؤمَّنة باستخدام مُصادَقة المُستخدِم والمدعومة بالأجهزة، تقوم نُظُم تشفير التخزين بتشفير جميع البيانات التي يُنشئها المُستخدِم تلقائيّاً قبل كتابتها على القرص، وتقوم بفكّ تشفير البيانات تلقائيّاً عند قراءتها قبل إعادتها إلى العمليّة التي تطلبها [3].

تعتمد نُظُم تشفير التخزين بشكلٍ أساسيٍّ على خوارزميّة التشفير AES، تدعم بعض معماريّات وحدات المُعالَجة المركزيّة في منصّات أندرويد تسريع AES في حين يفتقر بعضها الآخر إلى ذلك.

يدعم أندرويد 9 والإصدارات الأحدث طريقة التشفير Adiantum والتي تستخدم المُشفِّر AES بشكلٍ جزئيّ، تمّ تصميمها خصّيصاً للأجهزة التي تفتقر وحدات المُعالَجة المركزيّة الخاصّة بها إلى تعليمات AES لتُمكِّنها من استخدام تشفيرٍ قويٍّ مع عبءٍ مُضافٍ قليل (AOSP, 2021, 38)، يجب عدم استخدام الطريقة Adiantum في منصّات أندرويد المشحونة بوحدات مُعالَجةٍ مركزيّةٍ تدعم تسريع AES؛ حيث تكون الخوارزميّة AES أسرع على تلك المنصّات [2].

تمّ تصميم الخوارزميّة AES مع اعتبار أنّ أداء التشفير أكثر أهميّةً من أداء فكّ التشفير لسببين كانا وجيهَين آنذاك، السبب الأوّل هو أنّه في بعض أنماط عمل المُشفِّر الكتليّ تُستَخدَم وظيفة التشفير فقط في كلتا عمليّتَي تشفير وفكّ تشفير الكتلة، السبب الثاني هو أنّه يُمكِن استخدام الخوارزميّة AES في خدمة التحقُّق من الوثوقيّة والتي تستخدم وظيفة التشفير فقط (Stallings, 2017, 189).

في وقتنا الحالي؛ يُعتبَر أداء فكّ التشفير هو الأكثر أهميةً في نظم تشفير القرص وبشكلٍ خاصّ في الأجهزة المحمولة محدودة الموارد بما فيها أجهزة أندرويد؛ نظراً لأنّ عمليّات القراءة تكون أكثر تكراراً من عمليّات الكتابة، وهي تُؤثِّر بشكلٍ عامٍّ على وقت الاستجابة الذي يتوقّعه المُستخدِم، بينما يُمكِن لأنظمة التشغيل أن تُؤدّي عمليّات الكتابة بشكلٍ غير مُتزامِنٍ في الخلفيّة (Crowley *et al.*, 2018, 5).

يُؤثِّر بطء زمن فكّ التشفير مُقارَنةً بزمن التشفير في الخوارزميّة AES على أداء نظام التشفير؛ وبشكلٍ خاصٍّ في المنصّات التي لا تدعم وحدات المُعالَجة الخاصّة بها تسريع AES، يعود السبب في ذلك إلى أنّ قِيَم مُعامِلات كثير الحدود المُستخدَم في تحويل خلط الأعمدة في عمليّة

التشفير تكون أخفض وأسرع في التنجيز مُقارَنةً بقِيَم مُعامِلات كثير الحدود المُستخدَم في تحويل خلط الأعمدة العكسيّ في عمليّة فكّ التشفير.

تهدف الورقة البحثيّة إلى تعديل الخوارزميّة AES للحصول على خوارزميّة تُحافظ على الميزات المُفضَّلة وتُوفِّر أداءً أفضل ووزناً أخفّ ومستوى أمانٍ مناسباً ضدّ جميع الهجمات المعروفة، بحيث تكون مُلائمةً للاستخدامات العامّة والاستخدام الخاص في نُظُم تشفير التخزين في أجهزة أندرويد بشكلٍ مُستقلٍّ عن تَوافُر دعم تعليمات AES في وحدات المُعالَجة المركزيّة للأجهزة.

تمّ تنظيم باقي هذه الورقة على النحو التالي: في القسم الثاني سنقدِّم توصيفاً موجزاً للخوارزميّة AES والتحويلات التي تُجريها، في القسم الثالث سيتمّ استعراض أبرز الدراسات السابقة المُتعلِّقة بتحسين أداء الخوارزميّة AES، في القسم الرابع سنقدِّم توصيفاً للخوارزميّة AES المُعدَّلة المُقترَحة مع شرح التعديل المُقترَح على تحويل خلط الأعمدة، في القسم الخامس سنُناقش أمان الخوارزميّة المُعدَّلة بما يشمل مُقاوَمة الهجمات المعروفة والهجمات على التنجيز، في القسم السادس سنُجري مُقارَنةً لميزات الخوارزميّة المُعدَّلة وأدائها المُتوقَّع مع الخوارزميّة AES في التنجيزات المُخصَّصة، في القسم السابع سنستعرض نتائج التنجيز العمليّ للأداء وخصائص الخلط والنشر مع مُقارَنتها بنتائج الخوارزميّة AES على منصّة حاسوبٍ مكتبيٍّ ومنصّتَين من أجهزة أندرويد، وأخيراً سنقوم في القسم الثامن بمُناقَشة النتائج وتحليلها مع استعراض مجالات الاستخدام في النظام أندرويد وآفاق التطوير.

2- الخوارزميّة AES:

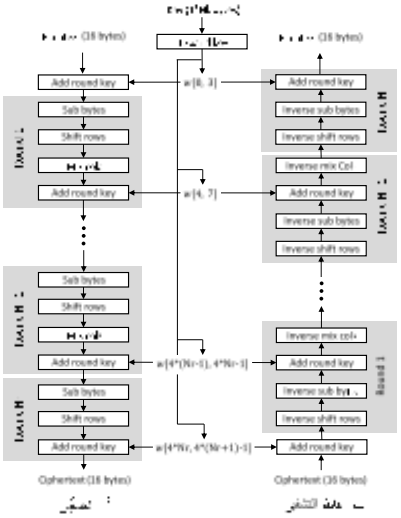
تستخدم الخوارزميّة AES في كلٍّ من المُشفِّر وفاكّ التشفير تحويل جولةٍ له هيكليّة شبكة الاستبدال والتقليب (Substitution-Permutation Network).

يتمّ تمرير كتلة النصّ الصريح المُكوَّنة من Nb = 4 كلمةً ذات 32 بت ومفتاح التشفير المُكوَّن من Nk كلمةً ذات 32 بت كدخلٍ للمُشفِّر. يُجري المُشفِّر سلسلةً من التحويلات على النصّ الصريح باستخدام مفتاح التشفير وينتج في خرجه كتلة النصّ المُشفَّر المُقابِلة، في بداية عمله؛ يقوم المُشفِّر بنسخ كتلة النصّ الصريح إلى مصفوفة الحالة (State)، كما يقوم بتوسيع مفتاح التشفير باستخدام روتين توسيع المفتاح للحصول على جدول مفاتيح الجولات، حيث أنّ الجدول يحتوي على (Nr+1) مفتاح جولة؛ كلٌّ منها بطولٍ يساوي حجم كتلة الحالة، ويقوم المُشفِّر بمُعالَجة مصفوفة الحالة كما يلي:

- إجراء إضافةٍ أوليّةٍ لمفتاح الجولة الأولى المُستخرَج من جدول المفاتيح إلى مصفوفة الحالة.
- تطبيق تحويل الجولة على مصفوفة الحالة لعدد جولاتٍ يساوي Nr، حيث أنّه في كلّ جولةٍ يُمرَّر مفتاح الجولة كمُعامِلٍ إلى تحويل الجولة.
- يُوجَد اختلافٌ بسيطٌ بين تحويل الجولة النهائيّة وتحويل الجولات التي تسبقها؛ حيث أنّ تحويل الجولة النهائيّة لا يحتوي على تحويل خلط الأعمدة بخلاف تحويل الجولات السابقة، والهدف من ذلك هو جعل المُشفِّر وفاكّ التشفير أكثر تشابهاً من حيث الهيكليّة، علماً أن ذلك لا يُحسِّن أو يقلِّل من أمان التشفير بأيّ شكلٍ من الأشكال.

يتمّ تمرير النصّ المُشفَّر ومفتاح التشفير كمُدخَلاتٍ لفاكّ التشفير ويُجري أربعة تحويلاتٍ متتاليةٍ لمُعالَجة الحالة ويُنتِج في خرجه كتلة النصّ الصريح المُقابِلة. يقوم فاكّ

التشفير بتوسيع مفتاح التشفير للحصول على جدول المفاتيح بنفس الطريقة المُعتمَدة في المُشفِّر، ويُجري سلسلةً من التحويلات المُعاكِسة لتلك المُنفَّذة في المُشفِّر؛ مع استخدام مفاتيح الجولات بترتيبٍ مُعاكسٍ لترتيب استخدامها في المُشفِّر؛ كما هو مُوضَّحٌ في الشكل 1.



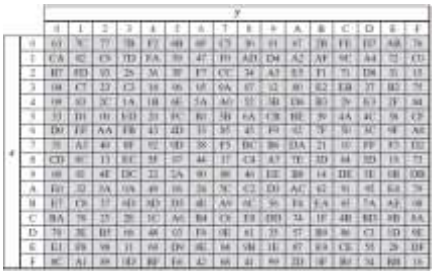
الشكل (1) هيكليّة المُشفِّر وفاكّ التشفير في الخوارزميّة AES

2-1- تحويل استبدال البايتات (SubBytes():

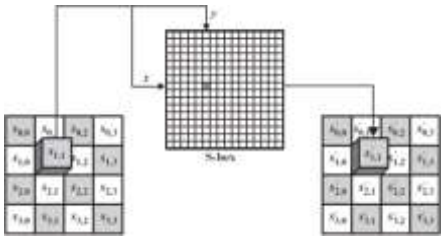
هو تحويلٌ لاخطيٌّ يقوم باستبدال كلّ بايتٍ في مصفوفة الحالة بشكلٍ مُستقلٍّ باستخدام جدول الاستبدال S-box. إنّ جدول الاستبدال S-box هو جدول بحثٍ (lookup table) مُمثَّلٌ بالصيغة الستّ عشريّة؛ كما هو مُبيَّنٌ في الشكل 2.

يتمّ استخدام جدول الاستبدال S-box لاستبدال البايت من مصفوفة الحالة 'xy' = $s_{r,c}$ بحيث يتمّ التوصُّل إلى نتيجة الاستبدال $s'_{r,c}$ على أنّها العنصر في جدول الاستبدال المُحدَّد بتقاطُع الصفّ ذي الدليل x مع العمود ذي الدليل y؛ كما هو مُوضَّحٌ في الشكل 3. ويُمكِن التعبير عن ذلك بالعلاقة التالية:

SubByte('xy') = S-box[x, y]



الشكل (2) جدول الاستبدال S-box للقيمة الستّ عشريّة 'xy'



الشكل (3) تطبيق صندوق الاستبدال على بايتات مصفوفة الحالة

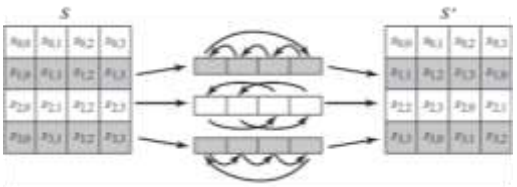
يستخدم التحويل العكسيّ لاستبدال البايت InvSubBytes في فاكّ التشفير جدول الاستبدال العكسيّ IS-box المُوضَّح في الشكل 4.

الشكل (4) جدول الاستبدال IS-box
للقيمة الستّ عشريّة 'xy'

2-2- تحويل إزاحة الصفوف ShiftRows:

هو تحويلٌ خطيٌّ يُجري إزاحةً تدويريّةً لبايتات صفوف مصفوفة الحالة باتجاه اليسار، بحيث تتمّ إزاحة بايتات الصفّ r_i نحو اليسار بمِقدار i بايت حيث أنّ $0 < i < 4$؛

وهذا يعني أنّ الصفّ الأول يبقى دون إزاحةٍ[1]؛ كما هو مُوضَّحٌ في الشكل 5.



الشكل (5) تحويل إزاحة الصفوف في الخوارزميّة AES

يُجـري التحويـل العكسـي لإزاحـة الصـفوف InvShiftRows في فاكّ التشفير الإزاحة التدويريّة باتجاه اليمين.

2-3- تحويل خلط الأعمدة MixColumns:

هو تحويلٌ خطـيٌّ يُعالِج أعمدة الحالة بشكلٍ مُستقلٍّ، حيث يُعامَل عمود الحالة (c) ككثير حدودٍ ذي أربعة حدودٍ (من الدرجة دون الرابعة) وبمُعامِلاتٍ في الحقل المُنتهي $GF(2^8)$، بحيث يُعرَّف كما يلي:

$$s_c(x) = s_{3,c}\,x^3 + s_{2,c}\,x^2 + s_{1,c}\,x + s_{0,c}$$

يُجري تحويل خلط الأعمدة عمليّة الجداء المعياريّ (باستخدام المعيار $M(x) = x^4 + 1$) بين كثير الحدود الذي يُعبِّر عن عمود الحالة $s_c(x)$ وكثير الحدود المُثبَّت $a(x)$ المعرَّف كما يلي:

$$a(x) = '03'\,x^3 + '01'\,x^2 + '01'\,x + '02'$$

يُشار للجداء المعياريّ بالرمز $\otimes$، وبالتالي يتم حساب عمود الحالة الناتج $s'_c(x)$ كما يلي:

$$\begin{aligned} s'_c(x) &= a(x) \otimes s_c(x) \\ &= [a(x) * s_c(x)] \bmod M(x) \\ &= s'_{3,c}\,x^3 + s'_{2,c}\,x^2 + s'_{1,c}\,x + s'_{0,c} \end{aligned}$$

يُمكِن صياغة الجداء المعياريّ السابق على أنّه عمليّة ضرب المصفوفة العموديّة التي تُعبِّر عن شعاع كثير الحدود $s_c(x)$ بالمصفوفة الدائريّة[2] التي يُعتبَر $a(x)$ كثير الحدود المُصاحِب لها، وبناءً على ذلك فإنّه يُمكِن حساب قيمة عمود الحالة الناتِج $s'_c(x)$ عن طريق ضرب المصفوفات كما يلي:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} '02' & '03' & '01' & '01' \\ '01' & '02' & '03' & '01' \\ '01' & '01' & '02' & '03' \\ '03' & '01' & '01' & '02' \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad ; 0 \le c < 4$$

إنّ عناصر المصفوفة هي عناصر في الحقل المُنتهي $GF(2^8)$ ويُمكِن تمثيلها كأعدادٍ صحيحةٍ تُعبِّر عن كثيرات حدودٍ مُعرَّفةٍ على الحقل المُنتهي Z_2 أي أنّ مُعامِلاتها تنتمي إلى المجموعة {0, 1}.

وبالتالي فإنّ عمليّة جمع عنصرين تُكافئ إجراء عمليّة XOR بين العنصرين على مستوى البت، وعمليّة ضرب عنصرين تُكافئ إجراء العمليّات الحسابيّة المعياريّة (موديولو 2) على مُعامِلات كثيرات الحدود، مع اختزال ناتج عمليّة الضرب بالنسبة لكثير الحدود الغير قابل للاختزال[3] من الدرجة الثامنة $m(x) = x^8 + x^4 + x^3 + x + 1$ الذي يُكافئ القيمة الستّ عشريّة '11B' وهو ما يُعرَف بالضرب المعياري ويُشار له بالرمز •.

كنتيجةٍ لذلك تُحسَب مُعامِلات كثير الحدود $s'_c(x)$ وفق العلاقات التالية؛ حيث أنّ العمليّات على الدليل i هي موديولو 4:

$$s'_{i,c} = ('02' \bullet s_{i,c}) \oplus ('03' \bullet s_{i+1,c}) \oplus s_{i+2,c} \oplus s_{i+3,c}$$

---

[1] يُجري تحويل إزاحة الصفوف في الخوارزميّة Rijndael إزاحةً تدويريّةً للبايتات الموجودة في الصفوف الثلاثة الأخيرة من مصفوفة الحالة بمَقاديرَ مختلفةٍ من الإزاحات (k1, k2, k3) تعتمد قِيَمها على حجم الكتلة؛ أي أنّها تتعلّق بقيمة المُعامِل Nb.

[2] المصفوفة الدائريّة (circulant matrix): هي مصفوفةٌ مُربَّعةٌ يتمّ توصيفها بشكلٍ كاملٍ باستخدام شعاعٍ واحدٍ يُمثِّل كثير الحدود المُصاحِب للمصفوفة (associated polynomial)، وفيها يتمّ تدوير كلّ شعاع صفٍّ عنصراً واحداً إلى اليمين بالنسبة إلى شعاع الصفّ السابق.

[3] يُعَدّ كثير الحدود m(x) غير قابلٍ للاختزال إذا كان لا يُمكِن التعبير عنه كجداء كثيرَي حدودٍ من درجةٍ أقلّ.

بنـاءً علـى وصـف الضـرب المعيـاريّ؛ فـإنّ ضـرب العنصـر '02' بالعنصـر b تُكـافئ عمليّـة ضـرب كثيـرات الحدود التالية: $x \bullet b(x) = [x * b(x)] \bmod m(x)$

والتي تُترجَم على مستوى البايت بإجراء إزاحةٍ لليسار مَتبوعةً بعمليّة XOR شرطيّةٍ مع البايت ذي القيمة '1B'؛ وفق الآتي:

$$x \bullet b(x) = \begin{cases} \{b_6b_5b_4b_3b_2b_1b_00\} & ; b_7=0 \\ \{b_6b_5b_4b_3b_2b_1b_00\} \oplus \text{'1B'} & ; b_7=1 \end{cases}$$

يُشـار إلـى العمليّـة $x \bullet b(x)$ بـِ xtime(b)، يُمكِـن إجراء الضرب بقوىً أعلى لِـ x عن طريق التطبيق المُتكرِّر لِـ xtime.

فـي فـاكّ التشـفير؛ يُجـري تحويـل خلـط الأعمـدة العكسـيّ InvMixColumns() نفـس العمليـات الحسـابيّة ولكـن باسـتخدام كثيـر الحـدود المُثبَّـت b(x) والـذي يمثـل المعكـوس الضـربيّ فـي الحقـل المُنتهـي $GF(2^8)$ لكثيـر الحدود المُثبَّت a(x)، يُعرَّف كثير الحدود المُثبَّت b(x) كما يلي:

$$b(x) = \text{'0B'}\,x^3 + \text{'0D'}\,x^2 + \text{'09'}\,x + \text{'0E'}$$

وبنـاءً علـى ذلك فإنّـه يُمكِن حساب قيمـة عمود الحالـة الناتِج $s'_c(x)$ عن طريق ضرب المصفوفات كما يلي:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} \text{'0E'} & \text{'0B'} & \text{'0D'} & \text{'09'} \\ \text{'09'} & \text{'0E'} & \text{'0B'} & \text{'0D'} \\ \text{'0D'} & \text{'09'} & \text{'0E'} & \text{'0B'} \\ \text{'0B'} & \text{'0D'} & \text{'09'} & \text{'0E'} \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad ; 0 \le c < 4$$

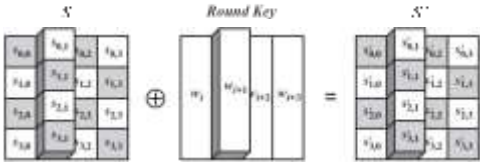
كنتيجـةٍ لـذلك تُحسَـب مُعـامِلات كثيـر الحـدود $s'_c(x)$ وفـق العلاقـات التاليـة؛ حيـث أنّ العمليّـات علـى الـدليل i هـي موديولو 4:

$$s'_{i,c} = (\text{'0E'} \bullet s_{i,c}) \oplus (\text{'0B'} \bullet s_{i+1,c}) \oplus (\text{'0D'} \bullet s_{i+2,c}) \oplus (\text{'09'} \bullet s_{i+3,c})$$

يُشـار إلـى المصـفوفة المسـتخدمة فـي تحويـل خلـط الأعمـدة بأنّهـا مصـفوفة MDS[4]، وبنـاءً علـى وصـف الضـرب المعيـاريّ وقِيَم المعـاملات المُستخدَمة في كثيري الحـدود المُثبَّتـين a(x) و b(x)؛ يكـون أداء تحويـل خلـط الأعمدة أفضل من أداء تحويل خلط الأعمدة العكسيّ.

2-3- تحويـــل إضـــافة مفتـــاح الجولـــة AddRoundKey():

هو تحويلٌ خطيٌ يقوم بإضافة مفتاح الجولة إلى الحالة عن طريق إجراء عمليّة XOR بسيطةٍ على مستوى البت، حيث يكون طول مفتاح الجولة مساوياً لحجم كتلة الحالة، ويتمّ استخراج مفتاح الجولة من جدول المفاتيح الناتج عن روتين توسيع المفتاح وإضافة كلّ كلمةٍ من كلماته إلى العمود المُقابِل لها من أعمدة مصفوفة الحالة، كما هو مُوضَّحٌ في الشكل 6. يُعتبَر تحويل إضافة مفتاح الجولة معكوساً ذاتيّاً كونه يستخدم عمليّة XOR البسيطة، ولذلك فهو يستخدم في كل من المشفر وفاكّ التشفير.



الشكل (6) تحويل إضافة المفتاح في الخوارزميّة AES

3- الدراسات السابقة:

اقترح Wadi *et al.* (2014) [14] نسخةً مُعدَّلةً عن الخوارزميّة AES بهدف تحسين الأداء في تشفير الصور عاليـة الدّقـة، يشـمل التعـديل تخفيض عـدد مرّات تطبيـق تحويل خلط الأعمدة (يتمّ تطبيقه في الجولات الفرديّة فقط) واسـتخدام صـندوق اسـتبدالٍ معكـوسٍ ذاتيّـاً لتخفيض كلفـة التنجيز العتاديّ، وتطبيق تحويل خلط الأعمدة على كلمات

[4] المصفوفة ذات أقصى مسافةٍ قابلةٍ للفصل (MDS matrix) هي مصفوفةٌ تُمثِّل وظيفةً خطيّةً ذات خصائص نشرٍ مُعيَّنةٍ مفيدةٍ تُوفِّر رقم فرعٍ أعظميّاً.

مفتاح الجولة قبل استخدامها في تحويل إضافة المفتاح، تُشير النتائج إلى اختصار زمن التشفير بنسبة 30%، ولكنّ صندوق الاستبدال المُقترَح يحمل خصائص خطيّةً واضحةً $S[xy] = \bar{x}\bar{y}$ وبالتالي فهو غير مقبولٍ للاستخدام، كما أنّ اختصار عدد مرّات تطبيق تحويل خلط الأعمدة يُهدِّد أمان الخوارزميّة ضدّ هجمات تحليل الشيفرة، وإنّ تطبيق تحويل خلط الأعمدة على كلمات مفاتيح الجولة قبل استخدامها في تحويل إضافة المفتاح لا يزيد من تعقيد العمليّات اللازمة لاسترجاع مفاتيح جولاتٍ أخرى كونه تحويلٌ خطيٌّ قابلٌ للعكس.

اقترح Abdulgader *et al.* (2015) [1] تحسين الخوارزميّة AES من أجل تشفير الصور عن طريق استخدام تحويل تبديل مَواقع البايتات بالاعتماد على الخرائط الفوضويّة (Arnold Cat Map) بدلاً من تحويل خلط الأعمدة وإجراء إزاحة جدول الاستبدال بشكلٍ ديناميكيٍّ بالاعتماد على مفتاح التشفير، تُظهِر النتائج أنّ أداء الخوارزميّة المُحسَّنة أسرع بحوالي ثلاثة أضعاف من أداء الخوارزميّة AES، لا تُحسِّن إزاحة جدول الاستبدال الديناميكيّة من مُقاوَمة هجمات تحليل الشيفرة بل توفِّر خلطاً إضافيّاً بما يخصّ المفتاح، كما أنّ الاستغناء عن تحويل خلط الأعمدة يؤدّي إلى نتائج سيّئةٍ من حيث أثر الانهيار وضعفٍ ضدّ تحليل الشيفرة الفرقيّ والخطيّ.

اقترح Wenceslao (2018) [15] نسخةً مُعدَّلةً عن الخوارزميّة AES بهدف تحسين الأداء؛ تستخدم صندوقَي استبدالٍ ويُشار لها AES-2SBOX، حيث يتمّ تطبيق طبقتَي استبدال بايت في الخوارزميّة المُعدَّلة، الطبقة الأولى هي تحويل استبدال البايت الذي يستخدم صندوق استبدال الخوارزميّة AES اللّاخطيّ، والطبقة الثانية هي تحويل استبدال بايتٍ جديدٌ بديلٌ عن تحويل خلط الأعمدة يستخدم صندوق استبدالٍ خطيّاً، تُظهِر النتائج تسريعاً جيّداً في أداء عمليّتَي التشفير وفكّ التشفير ولكنّه مُقترِنٌ بنتائج سيئةٍ للغاية في أثر الانهيار، يُعدُّ ذلك نتيجةً طبيعيةً لفقدان خصائص النشر التي يُحقِّقها تحويل خلط الأعمدة، يُضاف إلى ذلك أنّ الخوارزميّة المُعدَّلة غير آمنةٍ ضدّ تحليل الشيفرة الفرقي والخطي، وبالمقابل فإنّ زمن تنفيذ تحويل استبدال البايت ثابتٌ؛ ممّا يُحسِّن من مُقاوَمة الخوارزميّة لهجمات التوقيت.

اقترح Gamido *et al.* (2018) [9] نسخةً مُعدَّلةً عن الخوارزميّة AES بهدف تحسين الأداء في تشفير الصور والملفات النصيّة، قام الباحثون بالاستغناء عن تحويل خلط الأعمدة والاستعاضة عنه بتحويل تقليب البتّات ضمن العمود، تُظهِر النتائج تحسُّناً بسيطاً في الأداء من حيث زمن التنفيذ واستخدام المُعالِج، تُحسِّن عمليّة تقليب البتات بشكلٍ ثابتٍ من مُقاوَمة الخوارزميّة لهجمات التوقيت، ولكنّ مُخطَّط التقليب المُقترَح يستهلك حجم ذاكرةٍ إضافيّاً وهو غير مناسبٍ للتنجيزات البرمجية على مُعالِجات 8 بت كونه يُعالِج كتل بتّاتٍ جزئيّةً من أعمدة الحالة؛ الأمر الذي يُخفِّض الأداء.

بناءً على ما سبق؛ نستنتج أن المَنهجيّة الأساسيّة في تحسين أداء الخوارزميّة AES تعتمد على تبسيط تحويل النشر الخطيّ، ويشمل ذلك استبدال تحويل خلط الأعمدة بتحويلٍ أبسط [1] [9] [15] أو اختصار تكرار تنفيذه [14].

يُحسِّن التعديل [9] من أداء المُشفِّر بشكلٍ بسيطٍ مع تحسين أمان الخوارزميّة ضدّ الهجمات على التنجيز، في حين تُحسِّن التعديلات [1] [14] [15] من أداء المُشفِّر بشكلٍ كبيرٍ ولكنّها تُضعِف أمان الخوارزميّة ضد الهجمات المعروفة حسب ما تمّ شرحه في مُناقَشة كلٍّ من التعديلات السابقة.

4- الخوارزميّة المُقترَحة:

تُعتبَر الخوارزميّة المُقترَحة نسخةً مُعدَّلةً من الخوارزميّة AES تُوفِّر أداءً مُحسَّناً وكلفةً أقلّ، تستخدم الخوارزميّة المُقترَحة نفس الهيكليّة العامّة للخوارزميّة AES ونفس روتين توسيع المفاتيح، وتتقيَّد باستراتيجيّة تصميم المسار العريض (Wide Trail Strategy) ذاتها ممّا يجعل مُقارَنتها بالخوارزميّة AES أمراً سهلاً.

يتركَّز التعديل في الخوارزميّة المُقترَحة على تبسيط تحويل النشر في طبقة التقليب الخطيّة، حيث أنّه تمّ تبسيط العمليّات المُنفَّذة في تحويل خلط الأعمدة عن طريق استخدام مصفوفةٍ قريبةٍ من MDS دائريّةٍ ثنائيّةٍ معكوسةٍ ذاتيّاً تُوفِّر خصائصاً أخفّ وزناً مع زيادة عدد الجولات التي يتمّ تنفيذها (بمِقدار جولتين) بهدف ضمان هامش أمانٍ ضدّ الهجمات المعروفة، يُشار إلى تحويل خلط الأعمدة المُعدَّل بتحويل خلط الأعمدة الثنائيّ BinMixColumns وتُسمّى الخوارزميّة المُعدَّلة بـِ "خوارزميّة AES التي تستخدم تحويل خلط الأعمدة الثنائيّ" ويُطلَق عليها الاسم BinaryMixColumns-AES (BMC-AES).

تعتمد الخوارزميّة BMC-AES نفس المُعامِلات وبُنى المعطيات المُستخدَمة في الخوارزميّة AES، حيث يُحدَّد حجم كتلة الدخل وكتلة الخرج وكتلة الحالة بـِ 128 بت (باعتماد المُعامِل Nb = 4)، وتدعم الخوارزميّة ثلاثة أطوالٍ لمفتاح التشفير 128 أو 192 أو 256 بت (باعتماد المُعامِل Nk = 4, 6, 8).

يُشار إلى الخوارزميّة BMC-AES بإحدى التسميات "BMC-AES-128" أو "BMC-AES-192" أو "BMC-AES-256" بناءً على طول المفتاح المُستخدَم، كما تدعم الخوارزميّة قابليّة استخدام أحجام كتلةٍ إضافيّةٍ هي 192، 256 بت (باعتماد المُعامِل Nb = 4, 6, 8) ويعتمد عدد الجولات (Nr) التي يتمّ إجراؤها خلال تنفيذ الخوارزميّة على طول المفتاح وحجم الكتلة كما هو مُوضَّحٌ في الجدول 1.

الجدول (1) عدد الجولات في الخوارزميّة BMC-AES

Nr	Nb = 4	Nb = 6	Nb = 8
Nk = 4	12	14	16
Nk = 6	14	14	16
Nk = 8	16	16	16

4-1- تحويل خلط الأعمدة الثنائيّ:

هو تحويلٌ خطيٌّ مشابهٌ لتحويل خلط الأعمدة في المُشفِّر AES، فهو يُعالِج أعمدة مصفوفة الحالة بشكلٍ مُستقلّ، حيث يُعامَل عمود الحالة رقم (c) ككثير حدودٍ ذي أربعة حدودٍ (من الدرجة دون الرابعة) وبمُعامِلاتٍ في الحقل المُنتهي $GF(2^8)$، بحيث يُعرَّف كما يلي:

$$s_c(x) = s_{3,c}\,x^3 + s_{2,c}\,x^2 + s_{1,c}\,x + s_{0,c}$$

يستخدم التحويل BinMixColumns() كثير الحدود الثنائيّ a(x) الذي يكون معكوساً ذاتيّاً ولذلك فهو يُستخدَم في عمليّة التشفير وفكّ التشفير، ويُعرَّف كما يلي:

$$a(x) = x^3 + x^2 + x.$$

يقوم التحويل BinMixColumns() بحساب قيمة عمود الحالة الناتج $s'_c(x)$ بإجراء الجداء المعياريّ $s'_c(x) = a(x) \otimes s_c(x)$ بالنسبة لكثير حدودٍ أوليٍّ نسبياً من الدرجة الرابعة $M(x) = x^4 + 1$، يُمكِن صياغة الجداء المعياريّ على أنّه عمليّة ضرب المصفوفة العموديّة التي تُعبِّر عن شعاع كثير الحدود $s_c(x)$ بالمصفوفة الدائريّة التي يُعتبَر a(x) كثير الحدود المُصاحِب لها.

وبناءً على ذلك فإنّه يمكن حساب قيمة عمود الحالة الناتج $s'_c(x)$ عن طريق ضرب المصفوفات كما يلي:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad ; 0 \le c < 4$$

كنتيجةٍ لذلك تُحسَب مُعامِلات كثير الحدود $s'_c(x)$ وفق العلاقات التالية؛ حيث أنّ العمليّات على الدليل i هي موديولو 4:

$$s'_{i,c} = s_{i+1,c} \oplus s_{i+2,c} \oplus s_{i+3,c}$$

يُشار إلى مصفوفة النشر المُستخدَمة في تحويل خلط الأعمدة الثنائيّ بأنّها مصفوفةٌ قريبةٌ من MDS[5] دائريّةٌ ثنائيّةٌ[6] معكوسةٌ ذاتيّاً.

4-2- المنطق وخيارات التصميم:

يُجري تحويل خلط الأعمدة الثنائيّ خلطاً جيّداً بين بايتات كلّ عمودٍ في الحالة بشكلٍ مُستقلّ، وهو يضمن أن تعتمد جميع بايتات الخرج على جميع بايتات الدخل بعد مرور أربع جولات، تمّ اختيار تحويل خلط الأعمدة الثنائيّ من فضاء التحويلات الخطيّة التي تُجري عمليّاتها على شعاعٍ مُكوَّنٍ من 4 بايت بما يُحقِّق المعايير التالية:

1- أن يكون معكوساً ذاتيّاً: بما يُحقِّق التناظر ودرجة الكفاءة نفسها في عمليّات التشفير وفكّ التشفير.

2- أن يكون خطيّاً في الحقل المُنتهي GF(2).

3- أن يتمتَّع بخصائص نشرٍ مناسبة.

4- أن يكون الأخفّ وزناً على مُعالِجات 8 بت.

5- بساطة الوصف.

إنّ المعيارين الثاني والخامس هما اللّذان قادا إلى خيار الجداء المعياريّ لكثيرات الحدود بالنسبة لكثير الحدود المعياريّ $1 + x^4$؛ والذي جرى اختياره كأبسط معيارٍ مُتاح، في حين أنّ المعيار الأول هو الذي قاد إلى اختيار كثير الحدود المُثبَّت a(x) المعكوس ذاتيّاً من مجموعة كثيرات الحدود الأوليّة بالنسبة للمعيار، كما أنّ المعيار الرابع يفرض أن يكون كثير الحدود a(x) ثنائيّاً.

يفرض المعياران الثالث والرابع أن تضمن قِيَم المُعامِلات المُستخدَمة في كثير الحدود المُثبَّت الوصول إلى رقم فرعٍ $B(\theta) = 4$ وهو ليس أعظميّاً (بقيمة 5)، وهذا يعني أنّ الفرق في بايتٍ واحدٍ من الدخل سينتشر إلى 3 بايتٍ من الخرج، والفرق في 2 بايتٍ من الدخل سينتشر إلى 2 بايتٍ من الخرج على الأقلّ، أي أنّ العلاقة الخطيّة بين بتّات الدخل والخرج تتضمَّن البتّات من 4 بايتٍ مختلفٍ من الدخل والخرج على الأقلّ.

تمّت زيادة عدد الجولات في الخوارزميّة بمِقدار جولتين بالنسبة لجميع أطوال المفاتيح المدعومة كون رقم الفرع لتحويل خلط الأعمدة الثنائيّ ليس أعظميّاً، بما يُعوِّض عن سرعة النشر البطيئة في طبقة التقليب الخطيّة ويزيد من هامش الأمان ضدّ هجمات تحليل الشيفرة الخطيّ والفرقيّ.

5- أمان الخوارزميّة BMC-AES:

5-1- مُقاوَمة تحليل الشيفرة الفرقيّ والخطيّ:

في مُشفِّرٍ كتليٍّ يُعالِج كتلة بياناتٍ بطول n_b بت: تتطلَّب مُقاوَمة تحليل الشيفرة الفرقيّ اختيار عدد الجولات في المُشفِّر بحيث لا تُوجَد مساراتٌ فرقيّةٌ ذات احتماليّةٍ أعلى من 2^{1-n_b}، كما تتطلَّب مُقاوَمة تحليل الشيفرة الخطيّ اختيار عدد الجولات في المُشفِّر بحيث لا تُوجَد مساراتٌ خطيّةٌ ذات مُساهَمة ارتباطٍ أعلى من $2^{-n_b/2}$ Daemen *et al.* (2002).

تستخدم الخوارزميّة BMC-AES استراتيجية تصميم المسار العريض (WTS)، بحيث لا تُوجَد مساراتٌ فرقيّةٌ

[5] المصفوفة القريبة من MDS (almost MDS matrix) هي مصفوفةٌ تُمثِّل وظيفةً خطيّةً ذات خصائص نشرٍ جيّدةٍ ولكنّها لا تُوفِّر رقم فرعٍ أعظميّاً.

[6] المصفوفة الثنائيّة (binary matrix) هي مصفوفةٌ تنتمي عناصرها إلى المجموعة {0، 1}.

عبر (4 ,8 ,12) جولةً ذات احتماليّةٍ أعلى من (2^{-96}, 2^{-192}, 2^{-288}) على التتالي، ولا تُوجَد مساراتٌ خطيّةٌ عبر (4 ,8 ,12) جولةً ذات مُساهَمة ارتباطٍ أعلى من (2^{-48}, 2^{-96}, 2^{-144}) على التتالي، يُمكِن إثبات ذلك بناءً على النقاط التالية:

- سعة الارتباط الخطيّ لمسارٍ خطيٍّ هي جداء الارتباطات الخطيّة لصناديق الاستبدال الفعّالة، واحتماليّة المسار الفرقيّ هي جداء احتمالات انتشار الفرق لصناديق الاستبدال الفعّالة.
- وزن الارتباط الخطيّ لصندوق الاستبدال هو اللوغاريتم السالب لسعة الارتباط الخطيّ، ووزن انتشار الفرق لصندوق الاستبدال هو اللوغاريتم السالب لاحتمال انتشار الفرق.
- تمّ استخدام صندوق استبدال الخوارزميّة AES والذي يضمن احتمال انتشار فرقٍ أعظميّاً للصندوق يُساوي (2^{-6})؛ وهذا يُقابِل وزن انتشار فرقٍ أدنى يُساوي $p = 6$ ، ويضمن سعة ارتباطٍ خطيٍّ أعظميّاً للصندوق تُساوي (2^{-3})؛ وهذا يُقابِل وزن ارتباطٍ خطيٍّ أدنى يُساوي $c = 3$ (Daemen *et al.*, 2002, 143).
- تمّ تصميم تحويل خلط الأعمدة الثنائيّ بحيث يُحقِّق رقم فرعٍ $B(\theta) = 4$ وبالتالي فإنّ الحدّ الأدنى لوزن الحزمة للمسارات عبر جولتين يُساوي $B(\theta) = 4$، وإنّ تحويل إزاحة الصفوف يُحقِّق النشر المثاليّ بين الأعمدة والوصول إلى حدٍّ أدنى لوزن الحزمة للمسارات عبر أربع جولاتٍ يُساوي $(\theta)^2 = 16$.
- القيمة الدنيا لوزن المسار الفرقيّ تُساوي وزن الحزمة مضروباً بوزن انتشار الفرق الأدنى لصندوق الاستبدال، والقيمة الدنيا لوزن المسار الخطيّ تُساوي وزن الحزمة مضروباً بوزن الارتباط الخطيّ الأدنى لصندوق الاستبدال.
- وزن الحزمة لمسارٍ عبر $4k$ جولةً يُساوي وزن الحزمة لذلك المسار عبر 4 جولاتٍ مضروباً بالعدد k.

بناءً على ما سبق؛ فإنّ القيمة الدنيا لوزن المسار الفرقيّ عبر 4 جولاتٍ تُساوي $(96 = 6 * 16)$ وهي عبر 8 جولاتٍ تُساوي $(192 = 96 * 2)$ وهي عبر 12 جولةً تُساوي $(288 = 96 * 3)$، وبالتالي فإنّ أقصى احتماليّةٍ للمسار الفرقيّ عبر (4 ,8 ,12) جولةً تُساوي (2^{-288}, 2^{-192}, 2^{-96}) على التتالي.

إضافةً إلى ذلك؛ فإنّ القيمة الدنيا لوزن المسار الخطيّ عبر 4 جولاتٍ تُساوي $(48 = 3 * 16)$ وهي عبر 8 جولاتٍ تُساوي $(96 = 48 * 2)$ وهي عبر 12 جولةً تُساوي $(144 = 48 * 3)$، وبالتالي فإنّ أقصى سعة ارتباط للمسار الخطيّ عبر (4 ,8 ,12) جولةً تُساوي (2^{-48}, 2^{-96}, 2^{-144}) على التتالي.

نتيجةً لما سبق؛ تُبدي الخوارزميّة BMC-AES مُقاومةً لتحليل الشيفرة الخطيّ والفرقيّ وبهامش أمانٍ لا يقلّ عن أربع جولاتٍ حسب حجم الكتلة المُستخدَم، حيث أنّ 8 جولاتٍ تُعتبَر كافيةً من أجل حجم كتلةٍ يُساوي 128 بت أو 192 بت، كما أنّ 12 جولة تُعتبَر كافيةً من أجل حجم كتلةٍ يُساوي 256 بت.

5-2- مُقاوَمة هجمات الإشباع:

يعود أصل هذا النوع من الهجمات إلى هجومٍ مُخصَّصٍ على المُشفِّر SQUARE، وهو يستغلّ البنية المُوجَّهة للبايت في المُشفِّر؛ لذلك يُسمّى هذا النوع من الهجمات أيضاً بالهجمات البنيويّة ويُصنَّف هذا النوع من الهجوم ضمن الهجمات باستخدام نصٍّ صريحٍ مُختار (Chosen plaintext attack).

بشكلٍ مُشابِهٍ للخوارزميّة AES؛ يُعتبَر تطبيق هجوم الإشباع على الخوارزميّة BMC-AES أسرع من البحث الشامل عن المفاتيح لإصداراتٍ ذات عدد جولاتٍ مُخفَّضةٍ تصل إلى ستّ جولات.

لشرح الهجوم المُمكِن على الخوارزميّة BMC-AES سنُعرِّف المجموعة S المُكوَّنة من 256 حالةً مُختلِفةً في بعض البايتات (فعّالة) حيث أنّ قِيَم البايت الفعّال تتراوح على كامل مجال القِيَم المُمكِنة، وباقي البايتات تكون متساويةً بين الحالات (غير فعّالة)، يُمكِن التعبير عن المجموعة S كما يلي:

$$\forall\, x,y \in S = \begin{cases} x_{i,j} \neq y_{i,j} & if\ \ (i,j)\ is\ active \\ x_{i,j} = y_{i,j} & otherwise \end{cases}$$

بما أنّ بايتات حالات المجموعة S هي إمّا ثابتةٌ أو تتراوح على كامل مجال القِيَم المُمكِنة؛ فإنّ ناتج عمليّة XOR لجميع حالات المجموعة S يساوي الصفر؛ وبالتالي فإنّ العلاقة التالية تكون مُحقَّقة:

$$\bigoplus_{x \in S} x_{i,j} = 0\,, \quad \forall\, i,j$$

إنّ تطبيق تحويل استبدال البايتات وإضافة المفتاح على حالات المجموعة S ينتج عنه مجموعة S جديدةٌ دون تغيير مكان أو عدد البايتات الفعّالة، كما أنّ تطبيق تحويل إزاحة الصفوف ينتج عنه مجموعة S جديدةٌ لها نفس عدد البايتات الفعّالة ولكن في أماكن مختلفة، في حين أنّ تطبيق تحويل خلط الأعمدة الخطيّ لا ينتج عنه بالضرورة مجموعة S مالم يكن دخل تحويل خلط الأعمدة يحتوي على بايتٍ واحدٍ فعّالٍ على الأكثر؛ وبالتالي يكون على خرجه 4 بايت فعّال (Daemen *et al.*, 2002, 150).

عند تتبُّع البايتات الفعّالة لمجموعة S ذات بايتٍ واحدٍ فعّالٍ عبر ثلاث جولاتٍ نجد أنّه في الجولة الأولى يقوم تحويل خلط الأعمدة الثنائيّ بتحويل البايت الفعّال الوحيد إلى ثلاثة بايتاتٍ فعّالةٍ ضمن العمود؛ وبالتالي تنتج عنه مجموعة S جديدةٌ تحتوي ثلاثة بايتاتٍ فعّالة.

في الجولة الثانية يقوم تحويل إزاحة الصفوف بنشر البايتات الثلاث الفعّالة الناتجة على ثلاثة أعمدةٍ مختلفة، ويُؤدّي تطبيق تحويل خلط الأعمدة الثنائيّ إلى الحصول على ثلاثة أعمدةٍ يحتوي كلٌّ منها على ثلاث بايتاتٍ فعّالة؛ وبالتالي تنتج عنه مجموعة S جديدةٌ تحتوي تسعة بايتاتٍ فعّالةٍ.

وصولاً إلى تحويل خلط الأعمدة الثنائيّ في الجولة الثالثة والذي لن ينتج عنه مجموعة S جديدةٌ لأنّ دخله يحتوي على أكثر من بايتٍ واحدٍ فعّال.

نُشير إلى مُدخَلات تحويل خلط الأعمدة الثنائيّ في الجولة الثالثة بالرمز a وإلى مُخرَجاته بالرمز b، وبالتالي من أجل جميع قِيَم i,j نحصل على ما يلي؛ حيث أنّ العمليّات على الدليل هي موديولو 4:

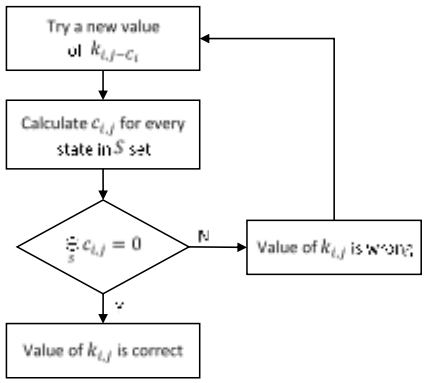
$$\begin{aligned} \bigoplus_{s} b_{i,j} &= \bigoplus_{s} BMC\left(a_{i,j}, a_{i+1,j}, a_{i+2,j}, a_{i+3,j}\right) \\ &= \bigoplus_{s} \left(a_{i+1,j} \oplus\ a_{i+2,j} \oplus\ a_{i+3,j}\right) \\ &= \bigoplus_{s} a_{i+1,j} \oplus \bigoplus_{s} a_{i+2,j} \oplus \bigoplus_{s} a_{i+3,j} \\ &= 0 \oplus 0 \oplus 0 = 0 \end{aligned}$$

أي أنّ ناتج عمليّة XOR لجميع حالات المجموعة S في خرج الجولة الثالثة يُساوي الصفر وتكون البايتات مُتوازِنة (balanced)، وعلى اعتبار أنّ الجولة الرابعة هي آخر جولةٍ في المُشفِّر –وبالتالي فهي لا تحتوي على تحويل خلط الأعمدة الثنائيّ– يكون مجموع كلّ بايتٍ في خرج الجولة الرابعة يعتمد على بايتٍ واحدٍ فقط من دخل الجولة الرابعة.

نُشير إلى مُدخَلات الجولة الرابعة بالرمز c وإلى مُخرَجاتها بالرمز d، وتكون العلاقة بين بايتات الدخل والخرج كما يلي؛ حيث أنّ العمليّات على الدليل هي موديولو 4:

$$d_{i,j} = S\text{-}box\left(c_{i,j+c_i}\right) \oplus k_{i,j}\,, \qquad \forall\, i,j$$
$$\Rightarrow c_{i,j} = S\text{-}box^{-1}\left(d_{i,j-c_i} \oplus k_{i,j-c_i}\right),\ \forall\, i,j \quad (*)$$

يقتضي الهجوم على إصدارٍ مُخفَّضٍ من الخوارزميّة BMC-AES ذي 4 جولاتٍ تجريب قيمة $k_{i,j-c_i}$ وحساب قِيَم $c_{i,j}$ المُقابِلة للنصوص المُشفَّرة الناتجة عن حالات المجموعة S وفق العلاقة (*)، بحال كانت قيمة $k_{i,j-c_i}$ المُجرَّبة تُوافِق قيمة بايت مفتاح الجولة الصحيحة يكون ناتج عمليّة XOR لِقِيَم $c_{i,j}$ المُقابِلة لحالات المجموعة S مساوياً الصفر (يتحقَّق شرط البايتات المُتوازِنة)؛ وإلّا فإنّ القيمة المُجرَّبة تكون خاطئةً وبالتالي يجب تجربة قيمة $k_{i,j-c_i}$ جديدة. يُوضِّح الشكل 7 إجرائيّة استرجاع بايتٍ من مفتاح الجولة باستخدام هجمات الإشباع:



الشكل (7) استرجاع بايتٍ من مفتاح الجولة باستخدام هجمات الإشباع

يتحقَّق شرط البايتات المُتوازِنة على الأغلب من أجل قيمةٍ واحدةٍ لبايت مفتاح الجولة من أصل 256 قيمةً تجريبيّةً مُمكِنة، بحال تحقُّق الشرط من أجل عدّة قِيَمٍ لبايت مفتاح الجولة؛ يُمكِن الاستعانة بمجموعة S ثانيةٍ وإزالة الالتباس، يُمكِن تكرار هذه الإجرائيّة من أجل باقي بايتات مفتاح الجولة وبالتالي استرجاع مفتاح الجولة كاملاً.

لاسترجاع بايتٍ واحدٍ من مفتاح الجولة؛ يتطلَّب الهجوم مُعالَجة جميع حالات المجموعة S والبالغ عددها 2^8 حالةً وتجربة 2^8 قيمةً مُمكِنةً لبايت المفتاح على الأكثر، ممّا يعني الحاجة إلى تنفيذ 2^{16} من عمليّات XOR والبحث في الجدول (LUT) والتي تُكافئ تعقيداً يُقارِب 2^{10} تنفيذاً لمُشفِّر BMC-AES ذي 4 جولات.

يجب التحقُّق من عددٍ ضئيلٍ من قِيَم بايت مفتاح الجولة باستخدام المجموعة S الثانية، ويجب تكرار الهجوم 16 مرةً لاستعادة مفتاح الجولة كاملاً، ينتج عن ذلك تعقيدٌ إجماليٌّ يُكافِئ 2^{14} تنفيذاً لمُشفِّر BMC-AES ذي 4 جولات.

يُمكِن توسيع الهجوم كما في الخوارزميّة AES ليشمل إصداراً ذا ستّ جولاتٍ ولكن ينتج عنه تعقيدٌ إجماليٌّ يُكافئ 2^{70} تنفيذاً لمُشفِّر BMC-AES ذي 6 جولات (Daemen *et al.*, 2002, 153)، ومن أجل إصداراتٍ ذات عدد جولاتٍ أكبر من ستة؛ يُعدُّ التعقيد الإجمالي الناتج أكبر من البحث الشامل عن المفاتيح.

5-3- مُقاوَمة هجمات التوقيت:

يُمكِن شنّ هجوم التوقيت إذا كان وقت تنفيذ خوارزميّة التشفير يعتمد على قيمة المفتاح، يكون ذلك مُمكِناً في المُشفِّر الذي يُنفِّذ تعليماتٍ شرطيّةً بناءً على قيمةٍ مُحدَّدةٍ لنتيجةٍ وسيطةٍ b مُتعلِّقةٍ بالمفتاح.

في الخوارزميّة AES؛ يتمثَّل الضعف الوحيد المُحتمَل فيما يتعلَّق بهجمات التوقيت في تنفيذ عمليّات ضرب المُعامِلات التي تنتمي إلى الحقل المُنتهي $GF(2^8)$ والمُستخدَمة في تحويل خلط الأعمدة وبالتحديد في الروتين الفرعي xtime الذي يُجري عمليّات الضرب بِـ '02' ويُنفَّذ باستخدام عمليّة إزاحةٍ وعمليّة XOR شرطيّةٍ بناءً على قيمة المُعامِل.

يُمكِن منع هجمات التوقيت عن طريق تنجيز الروتين xtime بطريقةٍ تستغرق عدداً ثابتاً من دورات المُعالَجة بغضّ النظر عن قيمة المُعامِل، يُمكِن تحقيق ذلك عن

طريق غرس تعليماتٍ وهميّةٍ في المسارات الشرطيّة الأقصر ولكنّ هذا النهج يُؤدّي إلى نقاط ضعفٍ ضدّ هجمات تحليل القدرة، كما يُمكِن تحقيقه عن طريق تنجيز الروتين xtime كعمليّة بحثٍ في جدولٍ مُكوَّنٍ من 256 بايتٍ يحوي ناتج عمليّة الضرب بـ '02' (Daemen *et al.*, 2002, 158) ولكنّ هذا النهج يستهلك حجماً ذاكريّاً إضافيّاً وزمناً أطول كون الوصول إلى الذاكرة أبطأ من تنفيذ عمليّة الإزاحة وعمليّة XOR شرطيّة التي تُعتبَر بسيطةً نسبيّاً.

بخلاف الخوارزميّة AES؛ يستخدم تحويل خلط الأعمدة الثنائيّ في الخوارزميّة BMC-AES عدداً ثابتاً من عمليّات XOR؛ ممّا يمنح الخوارزميّةBMC-AES ميزة أمان التنجيز ضدّ هجمات التوقيت من خلال ضمان أن تستغرق جميع التحويلات المُستخدَمة في الخوارزميّة وقتاً ثابتاً مُستقلاً عن قيمة المفتاح أو قيمة المُعامِلات.

5-4- مُقاوَمة هجمات تحليل القدرة:

يُوجَد نوعان من هجمات تحليل القدرة هما: تحليل القدرة البسيط (Simple Power Analysis) وتحليل القدرة الفرقيّ (Differential Power Analysis).

في هجوم تحليل القدرة البسيط (SPA)؛ يحصل المُهاجِم على قياسات استهلاك القدرة للجهاز أثناء تنفيذ عمليّة تشفيرٍ واحدةٍ ويستغلّ اعتماد نمط استهلاك القدرة للجهاز على التعليمات التي يتمّ تنفيذها، فإذا كان نمط استهلاك القدرة للجهاز يعتمد على التعليمات التي يتمّ تنفيذها؛ يُمكِن للمُهاجِم استنتاج تسلسل التعليمات؛ وإذا كان تسلسل التعليمات أو نوعها يعتمد على قيمة المفتاح فإنّ نمط استهلاك القدرة يؤدّي إلى تسريب معلوماتٍ حول المفتاح.

يُمكِن تنجيز الخوارزميّة AES بسهولةٍ من خلال تسلسلٍ ثابتٍ من التعليمات؛ ممّا يمنع هذا النوع من الهجوم (Daemen *et al.*, 2002, 158)، وبما أن الخوارزميّة BMC-AES تستخدم تسلسلاً ثابتاً من التعليمات؛ فإنّ هذه الميزة تمنح الخوارزميّة BMC-AES ميزة أمان التنجيز ضدّ هجوم تحليل القدرة البسيط (SPA).

في هجوم تحليل القدرة الفرقيّ (DPA)؛ يجمع المُهاجِم بين قياسات استهلاك القدرة للجهاز من أجل العديد من عمليّات التشفير ويستغلّ اعتماد نمط استهلاك القدرة في التعليمات على قيمة المُعامِلات. بشكلٍ مُماثلٍ للخوارزميّة AES؛ تُعتبَر الخوارزميّة BMC-AES عرضةً لهذا النوع من الهجمات ويُمكِن حماية التنجيز عن طريق إجراءات حماية التعليمات الفرديّة وإجراءات إلغاء التزامن.

تُركِّز إجراءات حماية التعليمات الفرديّة على حماية التعليمات الفرديّة بشكلٍ مُنفصِل، وتتمّ عن طريق التقليل أو التخلُّص تماماً من اعتماد استهلاك القدرة على قيمة المُعامِلات، كما في الخوارزميّة AES؛ يُمكِن تطبيق إجراءات حماية التعليمات الفرديّة عند تنجيز الخوارزميّة BMC-AES باستخدام تقنيّة مُوازَنة الحمل (load balancing) أو تقنيّة تقنيع المُعامِلات (operands masking). تُوجَد سلبيّةٌ لتطبيق إجراءات حماية التعليمات الفرديّة وهي ضرورة تكرارها من أجل كلّ تعليمةٍ تستخدمها الخوارزميّة (Daemen *et al.*, 2002, 159)، تتمتَّع الخوارزميّة BMC-AES بإمكانيّة تنجيزها بشكلٍ كاملٍ باستخدام تعليمات XOR وعمليّات البحث في الجداول (LUT) فقط ممّا يُسهِّل تطبيق إجراءات حماية التعليمات الفرديّة.

تسعى إجراءات إلغاء التزامن إلى الحدّ من تأثير هجوم تحليل القدرة الفرقيّ على مجموعة التعليمات عن طريق تغيير تسلسل تنفيذها لكلّ تشفيرٍ أو جزءٍ من التشفير، ممّا

يُصعِّب على المُهاجِم الحصول على إحصائيّاتٍ مفيدة (Daemen *et al.*, 2002, 159). يُمكِن تطبيق إجراءات إلغاء التزامن عند تنجيز الخوارزميّة BMC-AES كون تحويل الجولة يتميّز بخصائص تسمح بالتنفيذ المتوازي والذي يسمح ببعض التباين في تسلسل التعليمات.

6- مُقارَنة ميزات الخوارزميّة BMC-AES مع الخوارزميّة AES:

ستتمّ مُقارَنة الخوارزميّة BMC-AES مع الخوارزميّة AES من حيث خفّة وزن تحويل خلط الأعمدة في التنجيزات البرمجيّة والعتاديّة، ورقم الفرع لتحويل خلط الأعمدة، وسرعة النشر في طبقة التقليب الخطيّة.

6-1- أداء وخفّة وزن تحويل خلط الأعمدة:

يستخدم التحويل MixColumns() كثير الحدود المُثبَّت ذا المُعامِلات ('03', '01', '01', '02') ويُجري حساب البايت i من العمود c وفق العلاقة التالية (العمليّات على الدليل i هي موديولو 4):

$$s'_{i,c} = ('02' \bullet s_{i,c}) \oplus ('03' \bullet s_{i+1,c}) \oplus s_{i+2,c} \oplus s_{i+3,c}$$
$$s'_{i,c} = '02' \bullet s_{i,c} \oplus ('02' \oplus '01') \bullet s_{i+1,c} \oplus s_{i+2,c} \oplus s_{i+3,c}$$
$$s'_{i,c} = '02' \bullet (s_{i,c} \oplus s_{i+1,c}) \oplus s_{i+1,c} \oplus s_{i+2,c} \oplus s_{i+3,c}$$
$$s'_{i,c} = \text{xtime}\,(s_{i,c} \oplus s_{i+1,c}) \oplus s_{i+1,c} \oplus s_{i+2,c} \oplus s_{i+3,c}$$

بهدف التوصُّل إلى أقلّ عدد عمليّاتٍ مُمكِنٍ؛ يتمّ تعريف المُتحوِّل sum_c وحساب البايت i من العمود c وفق الخطوات التالية:

$$sum_c = s_{0,c} \oplus s_{1,c} \oplus s_{2,c} \oplus s_{3,c}$$
$$s'_{i,c} = \text{xtime}\,(s_{i,c} \oplus s_{i+1,c}) \oplus s_{i,c} \oplus sum_c$$

يُوضِّح الجدول 2 عدد عمليّات الإزاحة لليسار (SHL) وعمليّات XOR اللازمة لمُعالَجة كتلة الحالة باستخدام التحويل MixColumns().

الجدول (2) العمليّات اللازمة لمُعالَجة كتلة الحالة باستخدام التحويل MixColumns

الحساب المطلوب	العمليّات المُنفَّذة	العمليّات المُكافئة حسب قِيَم i,j	
		الحدّ الأدنى	الحدّ الأعلى
$s'_{i,c}$	xtime + 3 XOR	1 SHL + 3 XOR	1 SHL + 4 XOR
w'_c	4 xtime + 15 XOR	4 SHL + 15 XOR	4 SHL + 19 XOR
s'	16 xtime + 60 XOR	16 SHL + 60 XOR	16 SHL + 76 XOR

يستخدم التحويل InvMixColumns كثير الحدود المُثبَّت ذا المُعامِلات ('0B', '0D', '09', '0E') ويُجري حساب البايت i من العمود c وفق العلاقة التالية (العمليّات على الدليل i هي موديولو 4):

$$s'_{i,c} = ('0E' \bullet s_{i,c}) \oplus ('0B' \bullet s_{i+1,c}) \oplus ('0D' \bullet s_{i+2,c}) \oplus ('09' \bullet s_{i+3,c})$$
$$s'_{i,c} = ('08' \oplus '04' \oplus '02') \bullet s_{i,c} \oplus ('08' \oplus '02' \oplus '01') \bullet s_{i+1,c} \oplus ('08' \oplus '04' \oplus '01') \bullet s_{i+2,c} \oplus ('08' \oplus '01') \bullet s_{i+3,c}$$
$$s'_{i,c} = '08' \bullet (s_{i,c} \oplus s_{i+1,c} \oplus s_{i+2,c} \oplus s_{i+3,c}) \oplus '04' \bullet (s_{i,c} \oplus s_{i+2,c}) \oplus '02' \bullet (s_{i,c} \oplus s_{i+1,c}) \oplus s_{i+1,c} \oplus s_{i+2,c} \oplus s_{i+3,c}$$
$$s'_{i,c} = \text{xtime}(\text{xtime}(\text{xtime}\,(s_{i,c} \oplus s_{i+1,c} \oplus s_{i+2,c} \oplus s_{i+3,c})))$$

$$\oplus \text{xtime}(\text{xtime}(s_{i,c} \oplus s_{i+2,c}))$$
$$\oplus \text{xtime}(s_{i,c} \oplus s_{i+1,c})$$
$$\oplus s_{i+1,c} \oplus s_{i+2,c} \oplus s_{i+3,c}$$

بهدف التوصُّل إلى أقلّ عدد عمليّاتٍ مُمكِنٍ؛ يتمّ تعريف المُتحوِّلَين $sum_c, Xsum_c$ وحساب البايت رقم i من العمود c وفق الخطوات التالية:

$$sum_c = s_{0,c} \oplus s_{1,c} \oplus s_{2,c} \oplus s_{3,c}$$

$$Xsum_c = \text{xtime}(\text{xtime}(\text{xtime}(sum_c))$$
$$s'_{i,c} = Xsum_c \oplus \text{xtime}(\text{xtime}(s_{i,c} \oplus s_{i+2,c}))$$
$$\oplus \text{xtime}(s_{i,c} \oplus s_{i+1,c}) \oplus s_{i,c} \oplus sum_c$$

يُوضِّح الجدول 3 عدد عمليّات الإزاحة لليسار (SHL) وعمليّات XOR اللازمة لمُعالَجة كتلة الحالة باستخدام التحويل InvMixColumns.

الجدول (3) العمليّات اللازمة لمُعالَجة كتلة الحالة باستخدام التحويل InvMixColumns

الحساب المطلوب	العمليّات المُنفَّذة	العمليّات المُكافئة حسب قِيَم i,j	
		الحدّ الأدنى	الحدّ الأعلى
$s'_{i,c}$	3 xtime + 6 XOR	3 SHL + 6 XOR	3 SHL + 9 XOR
w'_c	15 xtime + 27 XOR	15 SHL + 27 XOR	15 SHL + 42 XOR
s'	60 xtime + 108 XOR	60 SHL + 108 XOR	60 SHL + 168 XOR

تُظهِر النتائج في الجدول 2 أنّ مُعالَجة كتلة الحالة باستخدام التحويل MixColumns() تتطلَّب إجراء العمليّات (16 SHL + 68 XOR) بشكلٍ وسطيّ، كما تُظهِر النتائج في الجدول 3 أنّ مُعالَجة كتلة الحالة باستخدام التحويل InvMixColumns() تتطلَّب إجراء العمليّات (60 SHL + 138 XOR) بشكلٍ وسطيٍّ وهذا يُكافئ حوالي 2.4 ضعف العمليّات المُستخدَمة في التحويل MixColumns.

يستخدم التحويل BinMixColumns() كثير الحدود المُثبَّت ذا المُعامِلات (1, 1, 1, 0) ويُجري حساب البايت i من العمود c وفق العلاقة التالية (العمليّات على الدليل i هي موديولو 4):

$$s'_{i,c} = s_{i+1,c} \oplus s_{i+2,c} \oplus s_{i+3,c}$$

بهدف التوصُّل إلى أقلّ عدد عمليّاتٍ مُمكِنٍ؛ يتمّ تعريف المُتحوِّل sum_c وحساب البايت رقم i من العمود c وفق الخطوات التالية:

$$sum_c = s_{0,c} \oplus s_{1,c} \oplus s_{2,c} \oplus s_{3,c}$$
$$s'_{i,c} = s_{i,c} \oplus sum_c$$

وبالتالي نستنتج أنّه تلزم عمليّة XOR واحدةٌ لحساب البايت $s'_{i,c}$، وتلزم /7/ عمليّات XOR لحساب العمود w'_c، وتلزم /28/ عمليّة XOR لمُعالَجة كتلة الحالة باستخدام التحويل BinMixColumns().

بالمُقارَنة بين عدد العمليّات المطلوبة في التحويلات السابقة؛ نستنتج أنّ زمن المُعالَجة باستخدام مصفوفة التحويل BinMixColumns() يبلغ حوالي 33% فقط من زمن المُعالَجة باستخدام مصفوفة التحويل MixColumns()، ويبلغ حوالي 14% فقط من زمن المُعالَجة باستخدام مصفوفة التحويل InvMixColumns.

يُمكِن المُقارَنة بين المصفوفات التحويلات السابقة بناءً على عدد XOR الذي يُعبِّر عن كلفة التنجيز العتاديّ كما هو مُقترَحٌ من قبل الباحِثَين Sim *et al.* (2015) حيث يُعرَّف عدد XOR لعنصرٍ في الحقل المُنتهي $GF(2^r)$ المُعرَّف على كثير الحدود غير القابل للاختزال $p(x)$ بأنّه عدد عمليّات XOR اللازمة لإجراء الضرب المعياريّ لعنصرٍ ما بهذا العنصر ويُشار له بـِ γ.

يتمّ حساب عدد عمليّات XOR المطلوبة لتنجيز صفٍّ كاملٍ M من مصفوفة النشر وفق العلاقة التالية (Sim *et al.*, 2015, 3):

$$XOR\ count\ of\ M = \sum_{i=0}^{K-1} \gamma_i + (n-1) * r$$

حيث أنّ K هو عدد العناصر في الصفّ، γ_i هو عدد XOR للعنصر رقم i في الصفّ، n هو عدد العناصر الغير صفريّة في الصفّ، r هو بُعد الحقل المُنتهي، وعندما تكون المصفوفة دائريّةً فإنّ عدد العمليّات الكلّي لتنجيز المصفوفة يساوي عدد XOR للصفّ مضروباً بعدد الصفوف.

يُبيِّن الجدول 4 نتائج مُقارَنة عدد XOR (عدد بوّابات XOR) للمصفوفات المُستخدَمة في تحويلات خلط الأعمدة MixColumns, InvMixColumns, BinMixColumns.

الجدول (4) عدد XOR لمصفوفات تحويلات خلط الأعمدة

	مُعامِلات الصف	عدد XOR للصف	عدد XOR للمصفوفة
مصفوفة التحويل MixColumns	('03', '01', '01','02')	(11+0+0+3) +(4-1)∗8 = 38	152
مصفوفة التحويل InvMixColumns	('0B', '0D', '09','0E')	(26+23+17+ 20) +(4-1)∗8 = 110	440
مصفوفة التحويل BinMixColumns	(1, 1, 1, 0)	(0+0+0+0) +(3-1)∗8 = 16	64

تُظهِر نتائج مُقارَنة عدد بوّابات XOR المُبيَّنة في الجدول 4 أنّ كلفة التنجيز العتاديّ لمصفوفة التحويل BinMixColumns هي بحدود 42% من كلفة التنجيز العتاديّ لمصفوفة التحويل MixColumns وبحدود 15% من كلفة التنجيز العتاديّ لمصفوفة التحويل InvMixColumns.

تَنسجم الاستنتاجات السابقة على مستوى كلفة التنجيز العتاديّ وزمن المُعالَجة مع نتائج الدراسات العمليّة التي أجراها الباحثون Banik *et al.* (2015) مع وجود اختلافاتٍ تتعلَّق بأمثليّة التنجيز، يُوضِّح الجدول 5 مُقارَنةً أعدّها الباحثون بين ميزات المصفوفة المُستخدَمة في التحويل BinMixColumns والمصفوفة المُستخدَمة في التحويل MixColumns (Banik *et al.*, 2015, 10).

الجدول (5) مُقارَنة ميزات المصفوفتين في التحويلين MixColumns(), BinMixColumns

رقم الفرع	زمن التأخير	مساحة الرقاقة	
5	0.68 ns	104 GEs	مصفوفة النشر في التحويل MixColumns
4	0.37 ns	48 GEs	مصفوفة النشر في التحويل BinMixColumns

نستنتج ممّا سبق؛ أنّ تحويل خلط الأعمدة الثنائيّ في الخوارزميّة BMC-AES يُوفِّر مَردوداً أفضل وكلفةً أقلّ من تحويل خلط الأعمدة/ خلط الأعمدة العكسيّ في الخوارزميّة AES الذي يستخدم عمليّات xtime.

6-2- رقم الفرع لتحويل خلط الأعمدة:

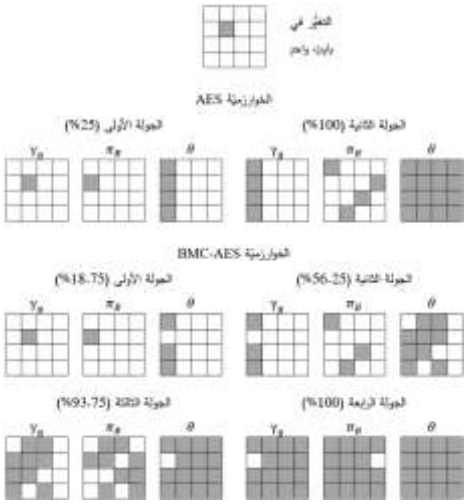
تُوفِّر مصفوفة النشر المُستخدَمة في تحويل خلط الأعمدة الثنائيّ في الخوارزميّة BMC-AES خصائص نشرٍ أقلّ مُقارنةً بمصفوفة النشر المُستخدَمة في تحويل خلط الأعمدة في الخوارزميّة AES، يتمّ التعبير عن خصائص النشر لتحويل خلط الأعمدة باستخدام رقم الفرع $B(\theta)$.

تُقدِّم مصفوفة النشر المُستخدَمة في تحويل خلط الأعمدة الثنائيّ في الخوارزميّة BMC-AES رقم فرعٍ $B(\theta) = 4$ أصغر مُقارَنةً مع رقم الفرع الأعظميّ $B(\theta) = 5$ الذي تُقدِّمه مصفوفة النشر المُستخدَمة في تحويل خلط الأعمدة في الخوارزميّة AES، والذي يؤثِّر بشكلٍ مباشرٍ بالمُقاوَمة ضدّ هجمات تحليل الشيفرة الخطيّ والفرقيّ وهجمات الإشباع وغيرها، وفق ما تمّ شرحه في أمان الخوارزميّة ضدّ الهجمات المعروفة.

6-3- سرعة النشر في الخوارزميّة:

تتعلَّق سرعة النشر في خوارزميّة التشفير بخصائص النشر لطبقة التقليب الخطيّة، وتُقاس سرعة النشر بعدد الجولات التي يستغرقها المُشفِّر لتحقيق النشر الكامل؛ بحيث تتأثَّر جميع البايتات في الخرج بجميع بايتات الدخل (Banik *et al.*, 2015, 10)، تتعلَّق سرعة النشر في طبقة التقليب الخطيّة بكلٍّ من تحويل خلط البايتات ضمن الأعمدة وتحويل نشر البايتات بين الأعمدة (إزاحة الصفوف).

يؤدّي رقم الفرع المنخفض للمصفوفة القريبة من MDS المُستخدَمة في تحويل خلط الأعمدة الثنائيّ في الخوارزميّة BMC-AES إلى سرعة نشرٍ أقلّ مُقارَنةً بمصفوفة MDS المُستخدَمة في تحويل خلط الأعمدة في الخوارزميّة AES. يُوضِّح الشكل 8 أنّ سرعة النشر في الخوارزميّة AES أفضل منها في الخوارزميّة BMC-AES، حيث أنَّه يتحقَّق النشر الكامل في نهاية الجولة الثانية للخوارزميّة AES؛ في حين أنّه لا يتحقَّق إلّا في نهاية الجولة الرابعة من الخوارزميّة BMC-AES؛ وهذا يعني تأخيراً بمِقدار جولَتين.



الشكل (8) سرعة النشر في الخوارزميّتين BMC-AES وAES

كـان بالإمكـان اسـتخدام تحويـل تبـديل مَواقِـع مُرافِـقٍ لتحويل خلط الأعمدة الثنائيّ أكثر كفاءةً من حيث سرعة النشر، ولكن تمّ الحفاظ على تحويل إزاحة الصفوف طالما أنّه يُحقِّق النشر الأمثليّ لبايتات الأعمدة مع ضمان بنيةٍ مُوحَّدةٍ للخوارزميّة من أجل جميع أحجام الكتل المُمكِنة، خصوصاً وأنّه تمَّت إضافة جولَتَين إضافيَّتين كهامش أمانٍ ضدّ تحليل الشيفرة الفرقيّ والخطيّ، وبالتالي سيتمّ تعويض بطء سرعة النشر عن طريق زيادة عدد الجولات.

6-4- مُقارَنة الخوارزميّة BMC-AES مع تنجيزات AES المُخصَّصة:

يُمكِن تنجيز الخوارزميّة AES بطرقٍ عديدةٍ تُوفِّر مُقايَضةً بين حجم الذاكرة والمَردود لتكون مناسبةً للاستخدام في تطبيقاتٍ مُعيَّنة، يتركّز الاختلاف بين التنجيزات في مدى اختصار العمليّات الحسابيّة التي يتمّ إجراؤها في تحويل الجولة والاستغناء عنها مُقابِل استخدام عمليّات البحث في الجداول (LUT) الجاهزة.

أ- التنجيزات المُخصَّصة على مُعالِجات 8 بت:

يُمكِن تمثيل الحقل المُنتهي $GF(2^8)$ باستخدام مُولِّدٍ (g) وتعريف العمليّات الحسابيّة باستخدام قوىالمُولِّد، بناءً على ذلك تقوم بعض التنجيزات ببناء جدول الأسّ $(Exptable)$ الذي يُمثِّل قوى المُولِّد في الحقل المُنتهي $GF(2^8)$ حيث أنّ:

$$Exptable[i] = g^i \quad , 0 \le i < 255$$

وبناء جدول اللّوغاريتم $(Logtable)$ الذي يُحقِّق:

$$Logtable[Exptable[i]] = i$$

ويتمّ تنجيز عمليّات الضرب المعياريّ في تحويل خلط الأعمدة/ خلط الأعمدة العكسيّ بين أيّة قيمتَين في الحقل المُنتهي $GF(2^8)$ كعمليّات جمع موديولو 255 وبحثٍ في الجدولَين $Logtable$ و$Exptable$ كما يلي:

$$mul(x,y) = Exptable[(Logtable[x] + Logtable[y]) \; mod \; 255]$$

بما أنّ مُعامِلات تحويل خلط الأعمدة تكون ثابتةً فإنّه يُمكِن استخدام النتيجة المُقابِلة لها من جدول اللّوغاريتم بصورةٍ مباشرة، وبالتالي فإنّ عمليّة الضرب بمُعامِلٍ أكبر من '01' تتطلَّب عمليَّتَي LUT وعمليّة جمعٍ موديولو 255، وبالتالي فإنّ تحويل خلط الأعمدة العكسيّ ينطوي على ثماني عمليّات LUT في حين أنّ تحويل خلط الأعمدة ينطوي على أربع عمليّات LUT فقط؛ فإنّ زمن تنفيذ تحويل خلط الأعمدة العكسيّ يزيد عن ضعف زمن تنفيذ تحويل خلط الأعمدة.

يُوفِّر هذا التنجيز الحماية ضدّ هجمات التوقيت بسبب عدم وجود تعليماتٍ شرطيّةٍ؛ ولكنّه يستهلك حجماً ذاكريّاً إضافيّاً بمِقدار 256 بايت من أجل كلّ من جدولَي الأسّ واللّوغاريتم ويُوفِّر مَردوداً أقلّ كون الوصول إلى الذاكرة أبطأ من تنفيذ عمليّة الإزاحة وعمليّة XOR الشرطيّة في التنجيز المَبني على عمليّات xtime.

كما تمّ شرحه في الفقرة 6-1؛ يُوفِّر تحويل خلط الأعمدة الثنائيّ في الخوارزميّة BMC-AES مَردوداً أفضل وكلفةً أقلّ من تحويل خلط الأعمدة/ خلط الأعمدة العكسيّ في الخوارزميّة AES الذي يستخدم عمليّات xtime؛ وبالتالي فهو حتماً يُوفِّر مَردوداً أفضل وكلفةً أقلّ مُقارَنةً بالتنجيز الذي يستخدم جداول اللّوغاريتم والأسّ.

ب- التنجيزات المُخصَّصة على مُعالِجات 32 بت:

باعتبار أنّ المصفوفة a هي مصفوفة الحالة على دخل تابع الجولة والمصفوفة e هي مصفوفة الحالة على خرج تابع الجولة والمصفوفة k هي مصفوفة مفتاح الجولة؛ يُمكِن صياغة ناتج تحويل الجولة في المُشفِّر باعتبار j هو رقم العمود والعمليّات عليه هي موديولو 4 كما يلي:

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} \mathrm{S}[a_{0,j}] \\ \mathrm{S}[a_{1,j+c1}] \\ \mathrm{S}[a_{2,j+c2}] \\ \mathrm{S}[a_{3,j+c3}] \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix}$$

يُمكِن تعريف الجداول $\mathrm{T}_0 .. \mathrm{T}_4$ التي يحتوي كلٌّ منها على 256 كلمةٍ مُكوَّنةٍ من 4 بايت كما يلي:

$$\mathrm{T}_0[a] = \begin{bmatrix} \mathrm{S}[a]\bullet 02 \\ \mathrm{S}[a] \\ \mathrm{S}[a] \\ \mathrm{S}[a]\bullet 03 \end{bmatrix} \quad \mathrm{T}_1[a] = \begin{bmatrix} \mathrm{S}[a]\bullet 03 \\ \mathrm{S}[a]\bullet 02 \\ \mathrm{S}[a] \\ \mathrm{S}[a] \end{bmatrix}$$

$$\mathrm{T}_2[a] = \begin{bmatrix} \mathrm{S}[a] \\ \mathrm{S}[a]\bullet 03 \\ \mathrm{S}[a]\bullet 02 \\ \mathrm{S}[a] \end{bmatrix} \quad \mathrm{T}_3[a] = \begin{bmatrix} \mathrm{S}[a] \\ \mathrm{S}[a] \\ \mathrm{S}[a]\bullet 03 \\ \mathrm{S}[a]\bullet 02 \end{bmatrix}$$

يُمكِن صياغة ناتج تحويل الجولة في المُشفِّر AES باستخدام الجداول $\mathrm{T}_4 .. \mathrm{T}_0$ باعتبار j هو رقم العمود والعمليّات عليه هي موديولو 4 كما يلي:

$$e_j = \mathrm{T}_0[a_{0,j}] \oplus \mathrm{T}_1[a_{1,j+c1}] \oplus \mathrm{T}_2[a_{2,j+c2}] \oplus \mathrm{T}_3[a_{3,j+c3}] \oplus k_j$$

يُمكِن استخدام جداول مُماثلةٍ في فاكّ التشفير، علماً أنّه لا يُمكِن الاستغناء عن الجدول S في المُشفِّر والجدول S^{-1} في فاكّ التشفير كون الجولة الأخيرة لا تحتوي على تحويل خلط الأعمدة/ خلط الأعمدة العكسيّ.

إنّ التنجيز السابق لخوارزميّة AES يجعل زمن التشفير وفكّ التشفير مُتساويَين كونه ينطوي على نفس العدد من عمليّات LUT. بالمُقارَنة مع التنجيز المَبني على عمليّات xtime؛ يُوفِّر التنجيز السابق الحماية ضدّ هجمات التوقيت، ومَردوداً أفضل كونه يختصر العمليّات الحسابيّة المطلوبة لضرب المُعامِلات بشكلٍ كامل، ولكنّه يستهلك حجماً ذاكريّاً إضافيّاً بمِقدار KB 4 في كلٍّ من المُشفِّر وفاكّ التشفير.

يُمكِن مُلاحَظة أنّ الجدول T_i ينتج عن تدوير كلمات الجدول T_{i-1} المُكوَّنة من 4 بايت أي أنّ $\mathrm{T}_i[a] = \mathrm{Rotbyte}[\mathrm{T}_{i-1}[a]]$، وبالتالي يُمكِن استخدام جدولٍ واحدٍ (T_0) وتوفير حجم KB 3 مُقابِل استخدام ثلاث عمليّات تدويرٍ لكلمات 4 بايت من أجل كلّ عمودٍ في الجولة الواحدة، وبالتالي يُمكِن صياغة ناتج تحويل الجولة في المُشفِّر باستخدام الجدول T_0 كما يلي:

$$e_j = k_j \oplus \mathrm{T}_0[a_{0,j}] \oplus \mathrm{Rotbyte}(\mathrm{T}_0[a_{1,j+c1}] \oplus \mathrm{Rotbyte}(\mathrm{T}_0[a_{2,j+c2}] \oplus \mathrm{Rotbyte}(\mathrm{T}_0[a_{3,j+c3}])))$$

يُمكِن استخدام تنجيزٍ مُماثلٍ للخوارزميّةBMC-AES. بالمُقارَنة مع تنجيز الخوارزميّة AES السابق؛ تُوفِّر الخوارزميّة BMC-AES مَردوداً أعلى بنسبة 33% في عمليّات التشفير وفكّ التشفير، وهي تستهلك نصف حجم الذاكرة المطلوب لكلتا عمليَّتَي التشفير وفكّ التشفير كونها تستخدم نفس الجداول في عمليَّتَي التشفير وفكّ التشفير، مع ميزةٍ إضافيّةٍ هي إمكانيّة توليد الجداول المطلوبة بشكلٍ أبسط لاختصار حجم الكود في التنجيز.

7- نتائج التنجيز العمليّ للخوارزميّة BMC-AES:

تمّ استخدام مكتبات التشفير bouncycastle التي يدعمها نظام التشغيل أندرويد، تُوفِّر مكتبات التشفير bouncycastle استخدام خوارزميّات تشفيرٍ مختلفةٍ يُشار لها بمُحرِّكات التشفير (crypto engines)، كما وأنّها تدعم أنماط العمل المختلفة للمُشفِّرات الكتليّة وأنماط الحشو المختلفة بالإضافة إلى العديد من أساسيّات التشفير مثل توابع الاختزال وأكواد وثوقيّة الرسالة والعديد من خدمات التشفير اللّاتناظريّ.

يُتيح الإصدار 1.70 من مكتبات التشفير bouncycastle ثلاثة مُحرِّكات تشفيرٍ مختلفةً للخوارزميّة AES تُوفِّر مُقايَضةً بين سرعة التنفيذ وحجم الذاكرة المطلوب، وهي:

1- مُحرِّك التشفير AESFastEngine: يستخدم ثمانية جداولٍ مُكوَّنةٍ من 256 كلمةً بطول 4 بايت، تُستخدَم الجداول الأربعة $T_0 .. T_4$ بالإضافة إلى صندوق الاستبدال S في عمليّة التشفير. وتُستخدَم الجداول الأربعة $Tinv_0 .. Tinv_4$ بالإضافة إلى صندوق الاستبدال S^{-1} في عمليّة فكّ التشفير. تختصر تلك الجداول العمليّات الحسابيّة المطلوبة لحساب أعمدة مصفوفة الحالة؛ كما تمّ شرحه في الفقرة 6-4 (ب).

2- مُحرِّك التشفير AESLightEngine: يستخدم جدولَين مُكوَّنين من 256 كلمةً بطول 4 بايت. يُستخدَم الجدول T_0 بالإضافة إلى صندوق الاستبدال S في عمليّة التشفير، ويُستخدَم الجدول $Tinv_0$ بالإضافة إلى صندوق الاستبدال S^{-1} في عمليّة فكّ التشفير، يُوفِّر ذلك حجماً ذاكريّاً بمِقدار 3 KB في كلٍّ من المُشفِّر وفاكّ التشفير مُقارَنةً بمُحرِّك التشفير AESFastEngine؛ على حساب استخدام 12 عمليّة تدويرٍ لكلمات 4 بايت في كلّ جولة؛ كما تمّ شرحه في الفقرة 6-4 (ب).

3- مُحرِّك التشفير AESEngine: يستخدم جدولَين مُكوَّنَين من 256 بايت، هما جدول اللّوغاريتم (Logtable) وجدول الأسّ (Alogtable) المَبنيّان باستخدام المُولِّد $g = '03$، واللّذان يُستخدَمان في تحويل خلط الأعمدة/ خلط الأعمدة العكسيّ لتنفيذ عمليّات الضرب المعياريّ بين أيّ قيمتَين في الحقل المُنتهي $GF(2^8)$ كعمليّات جمعٍ موديولو 255 وبحثٍ في الجداول (LUT)؛ كما تمّ شرحه في الفقرة 6-4 (أ).

تمَّت مُقارَنة أداء الخوارزميّة BMC-AES على مستوى عمليّات التشفير/ فكّ التشفير مع كلٍّ من تنجيز المحرِّك AESFastEngine وتنجيز المحرِّك AESLightEngine في الفقرة 6-4 (ب)، وتمّ التوصّل إلى أنّ مَردود الخوارزميّة BMC-AES أعلى بنسبة 33% في عمليّات التشفير وفكّ التشفير مع استهلاك نصف حجم الذاكرة المطلوب في كلّ من المُشفِّر وفاكّ التشفير.

في الفقرة 6-1؛ تمّت مُقارَنة أداء تحويل خلط الأعمدة الثنائيّ في الخوارزميّة BMC-AES مع أداء تحويل خلط الأعمدة/ خلط الأعمدة العكسيّ في الخوارزميّة AES التي تستخدم عمليّات xtime والتوصُّل إلى أنّ زمن المُعالَجة في مصفوفة التحويل BinMixColumns يبلغ حوالي 33% من زمن المُعالَجة في مصفوفة التحويل MixColumns، ويبلغ حوالي 14% من زمن المُعالَجة في مصفوفة التحويل InvMixColumns.

في الفقرة 6-4 (أ)؛ تمّت مُقارَنة أداء تحويل خلط الأعمدة الثنائيّ في الخوارزميّة BMC-AES مع أداء تحويل خلط الأعمدة/ خلط الأعمدة العكسيّ في الخوارزميّة

AES التي تستخدم جداول اللّوغاريتم والأسّ، وتمّ التوصّل إلى أنّ المَردود حتماً سيكون أفضل مع توفير 512 بايت من حجم الذاكرة المطلوبة، ولكن لم يتمّ التوصُّل إلى التحسين النهائيّ في مَردود عمليّتَي التشفير وفكّ التشفير.

بناءً على ما سبق؛ سيتمّ إجراء مُقارَنةٍ عمليّةٍ للأداء بين الخوارزميّة BMC-AES وتنجيزَين للخوارزميّة AES، الأول هو مُحرِّك التشفير AESEngine الذي يستخدم جدولَي اللّوغاريتم والأسّ في تحويلَي خلط الأعمدة وخلط الأعمدة العكسيّ، وسنُشير له بمُحرِّك التشفير AES-Logtable، والثاني هو مُحرِّك التشفير AESEngine المُعدَّل بحيث يستخدم عمليّات xtime الأمثليّة في تحويلَي خلط الأعمدة وخلط الأعمدة العكسيّ؛ وفق ما تمّ شرحه في الفقرة 6-1، وسنُشير له بمُحرِّك التشفير AES-xtime.

تشمل المُقارَنة مع التنجيزَين السابقَين تقييم الأداء من حيث مَردود عمليّات تشفير وفكّ تشفير الكتلة وذلك باستخدام قياسات زمن مُعالَجة الكتلة، بالإضافة إلى مُقارَنة خصائص النشر والخلط على خرج المُشفِّر من حيث أثر الانهيار[7] ومعيار الانهيار الصارم (SAC)[8] بين النصّ الصريح والنصّ المُشفَّر من جهةٍ؛ وبين المفتاح والنصّ المُشفَّر من جهةٍ أخرى.

7-1- مَردود عمليّات التشفير وفكّ التشفير:

لتقييم تأثير التعديل على الخوارزميّة AES بشكلٍ دقيقٍ؛ تمّ تعديل مُحرِّك التشفير AESEngine.class بما يقتصر على روتين خلط الأعمدة ()MixColumn وروتين خلط الأعمدة العكسيّ ()InvMixColumn وزيادة عدد الجولات وعدد مفاتيح الجولات بمِقدار 2 ضمن الروتين ()generateWorkingKey وبالتالي التوصُّل إلى الخوارزميّة BMC-AES، في حين تمّ التوصُّل إلى التنجيز AES-xtime بالتعديل على روتين خلط الأعمدة ()MixColumn وروتين خلط الأعمدة العكسيّ ()InvMixColumn فقط.

يستخدم تطبيق التشفير مُحرِّك التشفير بنمط الحشو والتخزين المُؤقَّت الذي يستخدم نمط الحشو PKCS7 داخليّاً، ويعمل في نمط تسلسل كتل التشفير CBC (Cipher Block Chaining)[9]؛ علماً أنّ اختيار نمط عملٍ أو نمط حشوٍ مُختلفَين لا يُؤثِّر على النتائج كون قياسات الزمن تتمّ على أضيق نطاقٍ مُمكِنٍ ضمن مُحرِّك التشفير أي ضمن روتين تشفير الكتلة ()encryptBlock وروتين فكّ تشفير الكتلة ()decryptBlock.

سيتمّ إجراء القياسات التي تخصّ الأداء على ثلاث منصّاتٍ مختلفةٍ:

- المنصّة الأولى هي حاسوبٌ شخصيٌّ من طراز HP Pavilion 15 Notebook PC ذو مُعالِج Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz (4 CPUs)، وذاكرة وصولٍ عشوائيٍّ DDR3 8GB يعمل عليه نظام التشغيل Windows 10 Pro 64-bit.
- المنصّة الثانية هي جهازٌ مَحمولٌ من طراز Samsung Galaxy Note 2 (GT-N7100) ذو مُعالِج

[7] أثر الانهيار (Avalanche effect): وهو يعني أنّ تغيُّر بتٍّ واحدةٍ من الدخل يجب أن ينتج عنه تغيُّرٌ في العديد من بتّات الخرج.

[8] معيار الانهيار الصارم (SAC): هو إصدارٌ أكثر صرامةً من معيار أثر الانهيار، والذي ينصّ على أنّ أيّ بتّ خرجٍ j يجب أن تتغيّر باحتمال (50%) عندما يتمّ عكس أيّة بتّ دخلٍ واحدةٍ i من أجل جميع قِيَم i, j.

[9] هو أحد أوضاع عمل المُشفِّر الكتليّ، يُوفِّر وضع العمل CBC الأمان عند تشفير عدّة كتلٍ باستخدام نفس المفتاح؛ عن طريق جمع (chaining) كتل النصّ الصريح مع كتل النصّ المُشفَّر السابقة قبل إدخالها إلى المُشفِّر، يتطلَّب وضع العمل CBC وجود شعاعٍ ابتدائيٍّ (Initial Vector) لجمعه مع أوّل كتلة نصٍّ صريح، لا يلزم أن يكون الشعاع الابتدائيّ سرّيّاً ولكن يجب أن يكون غير مُتوقَّع (Dworkin, 2001, 10).

Samsung Exynos Quad 4412 1.6 GHz بمعماريّة 4x ARM Cortex-A9 الذي لا يدعم تسريع AES (ARMv8 Cryptography Extensions)، وذاكرة وصولٍ عشوائيٍّ 2GB LPDDR2، يعمل عليه نظام التشغيل أندرويد إصدار 4.3.

– المنصّة الثالثة هي جهازٌ محمولٌ من طراز Samsung Galaxy J7 Pro (SM-J730F/DS) ذو مُعالِج Samsung Exynos Octa 7870 1.6 GHz بمعماريّة 8x ARM Cortex-A53 الذي يدعم تسريع AES (ARMv8 Cryptography Extensions)، وذاكرة وصولٍ عشوائيٍّ 3GB LPDDR3، يعمل عليه نظام التشغيل أندرويد إصدار 9.

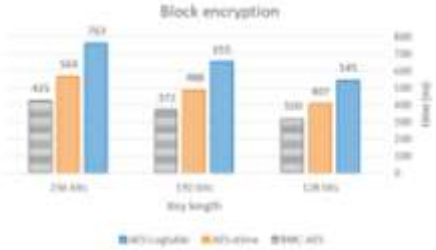
تمّ تطوير برنامجٍ اختباريٍّ يستقبل ملفاً بحجمٍ لا يقلّ عن /30.7 MB/ بما يُوفِّر أكثر من /2,000,000/ عيِّنةً اختباريّةً من كتل النصّ الصريح والمُشفَّر، يقوم البرنامج بتشفير كتل الملف ومن ثمّ فكّ تشفيرها تلقائيّاً وحساب الزمن الوسطيّ

على منصّة الحاسوب الشخصيّ؛ تمّ تنفيذ البرنامج الاختباريّ مئة مرّةٍ من أجل كلّ مُحرِّك تشفيرٍ باستخدام أطوال المفاتيح (128, 192, 256) بت، ثمّ حساب المُتوسِّط الحسابيّ لأدنى خمسين نتيجةً والتوصُّل إلى أزمنة عمليّات تشفير/ فكّ تشفير الكتلة؛ كما هو مُبيَّنٌ في الجدول 6 ومُوضَّحٌ في الشكل 9 والشكل 10.

تُشير قياسات الزمن في الجدول 6 إلى أنّ الخوارزميّة BMC-AES تُوفِّر مَردوداً مُحسَّناً بنسبة 70% في عمليّة التشفير ونسبة 145% في عمليّة فكّ التشفير مُقارَنةً بالتنجيز AES-Logtable، ومَردوداً مُحسَّناً بنسبة 27% في عمليّة التشفير ونسبة 96% في عمليّة فكّ التشفير مُقارَنةً بالتنجيز AES-xtime، علماً أنّ هذه النسب تزداد مع زيادة طول المفتاح (بسبب زيادة عدد الجولات التي يتمّ إجراؤها)؛ كما هو مُبيَّنٌ في الجدول 7.

الجدول (6) زمن تشفير/ فكّ تشفير كتلةٍ باستخدام أطوال مفاتيحٍ مُختلفةٍ على منصّة حاسوبٍ شخصيّ

time (ns)	128 bit key		192 bit key		256 bit key	
	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
AES-Logtable	545	785	655	944	763	1097
AES-xtime	407	626	488	748	569	874
BMC-AES	320		372		425	



الشكل (9) زمن تشفير كتلةٍ باستخدام أطوال مفاتيحٍ مُختلفةٍ على منصّة حاسوبٍ شخصيّ

الشكل (10) زمن فكّ تشفير كتلةٍ باستخدام أطوال مفاتيحٍ مُختلفةٍ على منصّة حاسوبٍ شخصيّ

الجدول (7) تحسُّن المَردود باستخدام الخوارزميّة BMC-AES على منصّة حاسوبٍ شخصيّ

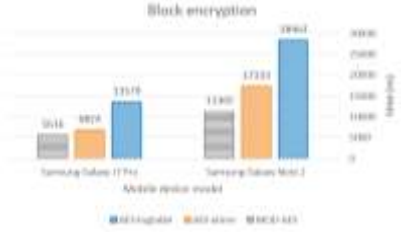
throughput improvement	128 bit key		192 bit key		256 bit key	
	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
AES-Logtable	%70	%145	%76	%154	%80	%158
AES-xtime	%27	%96	%31	%101	%34	%106

على منصّتَي الأجهزة المَحمولة؛ تمّ تنفيذ البرنامج الاختباريّ خمسين مرّةً من أجل كلّ مُحرِّك تشفيرٍ باستخدام مفتاحٍ بطول 128 بت، ثمّ حساب المُتوسِّط الحسابيّ لأدنى عشرين نتيجةً والتوصُّل إلى أزمنة عمليّات تشفير/

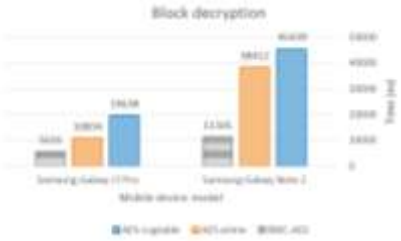
فكّ تشفير الكتلة؛ كما هو مُبيَّنٌ في الجدول 8 ومُوضَّحٌ في الشكل 11 والشكل 12.

الجدول (8) زمن تشفير/ فكّ تشفير كتلةٍ باستخدام مفتاح 128 بت على منصّتَي جهازٍ مَحمول

time (ns)	Samsung Galaxy		Samsung Galaxy J7 Pro	
	Enc.	Dec.	Enc.	Dec.
AES-Logtable	28,463	45,439	13,579	19,638
AES-xtime	17,333	38,412	6,824	10,834
BMC-AES	11,305		5,616	



الشكل (11) زمن تشفير كتلةٍ باستخدام مفتاح 128 بت على منصّتَي جهازٍ مَحمول



الشكل (12) زمن فكّ تشفير كتلةٍ باستخدام مفتاح 128 بتعلى منصّتَي جهازٍ مَحمول

الجدول (9) تحسُّن المَردود باستخدام الخوارزميّة BMC-AESعلى منصّتَي جهازٍ مَحمول

time (ns)	Samsung Galaxy		Samsung Galaxy	
	Enc.	Dec.	Enc.	Dec.
AES-Logtable	%152	%302	%142	%250
AES-xtime	%53	%240	%22	%93

بمُقارَنة نتائج تحسُّن المَردود في الجدول 9 مع مُقابِلاتها في الجدول 7؛ يُمكِن ملاحظة ما يلي:

- بما يخصّ التنجيز AES-Logtable على منصَّتَي الجهاز المَحمول؛ تُوفِّر الخوارزميّة BMC-AES تحسُّناً أكبر في المَردود مُقارَنةً بالنتائج التي تمّ التوصُّل إليها على منصّة الحاسوب الشخصيّ، وذلك يرتبط بالبطء النسبيّ لزمن الوصول إلى الذاكرة من نوع (LPDDR2, LPDDR3) في الأجهزة المَحمولة مُقارَنةً بزمن الوصول إلى الذاكرة (DDR3) في الحاسوب الشخصيّ.
- بما يخصّ التنجيز AES-xtime على منصَّتَي الجهاز المَحمول؛ تُوفِّر الخوارزميّة BMC-AES تحسُّناً في المَردود بنسبٍ مختلفة، إذ يكون تحسُّن المَردود على منصّة الجهاز المَحمول J7 Pro قريباً من تحسُّن المَردود على منصّة الحاسوب الشخصيّ؛ كون معماريّة مُعالِج الجهاز تدعم تسريع AES، في حين يكون تحسُّن المَردود على منصّة الجهاز المَحمول Note 2 أكبر من تحسُّن المَردود على منصّة الحاسوب الشخصيّ، كون معماريّة مُعالِج الجهاز لا تدعم تسريع AES، ممّا يؤدّي إلى تراجُعٍ في مَردود التنجيز AES-xtime مُقارنةً بمَردود الخوارزميّة BMC-AES.

يُبيِّن الجدول 10 مُقارَنةً نهائيّةً بين الخوارزميّة BMC-AES والتنجيز AES-Logtable والتنجيز AES-xtime على المنصّات الثلاثة السابقة من حيث مَردود عمليّات التشفير وفكّ التشفير باستخدام مفتاحٍ بطول 128 بت. حيث تمّ حساب المَردود مُقدَّراً بواحدة $(Mbps)$ من خلال

قياسات زمن معالجة الكتلة (T) الوارِدة في الجدولين 6-8 وفق العلاقة:

$$Throughput(Mbps) = \frac{10^9}{1024 * 8 * \text{T}\ (ns)}$$

الجدول (10) مُقارنة مَردود التشفير وفكّ التشفير باستخدام مفتاح 128 بت على المنصّات المختلفة

$throughput$ $(Mbps)$	Samsung Galaxy Note 2		Samsung Galaxy J7 Pro		HP Pavilion 15 Notebook PC	
	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
AES-Logtable	4.29	2.69	8.99	6.22	223.98	155.50
AES-xtime	7.04	3.18	17.89	11.27	299.93	195.00
BMC-AES	10.80		21,74		381.47	

كعيِّناتٍ اختباريّةٍ عشوائيّة، تمّ توليد الملف الاختباريّ عن طريق تشفير أيّ ملفٍّ (يُحقِّق الحدّ الأدنى للحجم المطلوب) باستخدام أيّ مفتاحٍ بنمط العمل CBC ثم إعادة تشفيره باستخدام مفتاحٍ آخرَ مختلف، ممّا يُنتِج ملفّاً شبه عشوائيٍّ يُمكِن استخدامه كملفٍّ اختباريّ.

يقوم البرنامج بقراءة كتلة نصٍّ صريحٍ (PT) من الملف الاختباريّ وحساب كتلة النصّ المُشفَّر المُقابِلة لها (CT) وتوليد مجموعة كتل النصّ الصريح [newPT] الناتجة عن تغيير بتٍّ واحدةٍ وحساب كتل النصّ المُشفَّر المُقابِلة لها [newCT].

يتمّ حساب أثر الانهيار من أجل كلّ كتلة نصٍّ صريحٍ (PT) عن طريق حساب المُتوسِّط الحسابيّ لمسافات هامينغ[10] بين كتلة النصّ المُشفَّر (CT) وكتل النصّ المُشفَّر [newCT]؛ كما هو مُوضَّحٌ في مثال الجدول 13. يتمّ حساب أثر الانهيار النهائيّ بأخذ المُتوسِّط الحسابيّ لأثر الانهيار لجميع كتل الملف الاختباريّ كما هو مُوضَّحٌ في مثال الجدول 14؛ وفق مُخطَّط البرنامج المُوضَّح في الشكل 15.

الشكل (13) مَردود التشفير وفكّ التشفير باستخدام مفتاح 128 بت على منصّة حاسوبٍ شخصيّ

الشكل (14) مَردود التشفير وفكّ التشفير باستخدام مفتاح 128 بت على منصّة جهازٍ مَحمول

7-2- خصائص النشر:

لمُقارَنة خصائص النشر على خرج المُشفِّر من حيث أثر الانهيار بين النصّ الصريح والنصّ المُشفَّر؛ تمّ تطوير برنامجٍ اختباريٍّ يستقبل مفتاح تشفيرٍ ثابتاً "SECRET_1SECRET_2"، وملفّاً بحجمٍ يزيد عن 15.25 MB/ لتوفير /1,000,000/ كتلة نصٍّ صريحٍ

[10] تُعبِّر مسافة هامينغ (Hamming Distance) بين كلمتين مُتساويتَي الطول عن عدد المَواقع التي تكون فيها قِيَم البت مُختلِفةً بين الكلمتين.

تُشير النتائج باستخدام /1,000,000/ كتلة نصٍّ صريحٍ أنّ تغيُّر بتٍّ واحدةٍ من كتلة النصّ الصريح على دخل الخوارزميّة AES يُؤدّي إلى تغيُّر (64.0003) بت من كتلة النصّ المُشفَّر على خرجها، ويُقابِلها تغيُّر (63.9997) بت على خرج الخوارزميّة BMC-AES، ممّا يعني أنّ أثر الانهيار في كلتا الخوارزميّتين جيّدٌ بحيث أنّ تغيُّر بتٍّ واحدةٍ من كتلة النصّ الصريح يؤدّي إلى تغيُّر 50% من بتّات كتلة النصّ المُشفَّر.

لمُقارَنة خصائص النشر على خرج المُشفِّر من حيث أثر الانهيار بين المفتاح والنصّ المُشفَّر؛ تمّ تطوير برنامجٍ اختباريٍّ يستقبل كتلة نصٍّ صريحٍ ثابتةً "PlainT_1PlainT_2"، وملفّاً بحجمٍ يزيد عن /15.25 MB/ لتوفير /1,000,000/ مفتاح تشفيرٍ بطول 128 بت كعيِّناتٍ اختباريّةٍ عشوائيّة، يتمّ حساب أثر الانهيار النهائيّ عبر مجموعات المفاتيح وفق مُخطَّط البرنامج المُوضَّح في الشكل 16.

تُشير النتائج باستخدام /1,000,000/ مفتاحاً أنّ تغيُّر بتٍّ واحدةٍ من المفتاح على دخل الخوارزميّة AES يُؤدّي إلى تغيُّر (64.00087) بت من كتلة النصّ المُشفَّر على خرجها، ويُقابِلها تغيُّر (64.00086) بتّ على خرج الخوارزميّة BMC-AES، ممّا يعني أنّ أثر الانهيار في كلتا الخوارزميّتين جيّدٌ بحيث أنّ تغيُّر بتٍّ واحدةٍ من المفتاح يُؤدّي إلى تغيُّر 50% من بتّات كتلة النصّ المُشفَّر.

7-3- خصائص الخلط:

لمُقارَنة خصائص الخلط على خرج المُشفِّر باستخدام معيار الانهيار الصارم (SAC) بين النصّ الصريح والنصّ المُشفَّر؛ تمّ تطوير برنامج حساب احتماليّة تغيُّر بتّ الخرج j من كتلة النصّ المُشفَّر عند تغيُّر بتّ الدخل i من كتلة النصّ الصريح ونُشير له بالاسم probability(i, j, g)، يستخدم البرنامج مفتاح تشفيرٍ ثابتاً "SECRET_1SECRET_2" وملفّاً بحجمٍ يزيد عن /15.25 MB/ لتوفير /1,000,000/ كتلة نصٍّ صريحٍ كعيِّناتٍ اختباريّةٍ عشوائيّة، ويستقبل رقم بتّ الدخل (i) ورقم بتّ الخرج (j) كمُدخَلات.

يقوم البرنامج بقراءة كتل النصّ الصريح [PT] من الملف الاختباريّ وحساب كتل النصّ المُشفَّر المُقابِلة لها [CT] ومن ثمّ يقوم بقلب البت i في كتل النصّ الصريح [PT] للحصول على كتلٍ جديدةٍ [newPT] وحساب كتل النصّ المُشفَّر المُقابِلة لها [newCT]، وحساب احتمال تغيُّر البت j بين الكتل [CT] والكتل [newCT] المُقابِلة لها؛ وفق مُخطَّط البرنامج المُوضَّح في الشكل 17.

يقتضي استيفاء معيار الانهيار الصارم (SAC) أن يتمّ التحقُّق من أنّ احتمال تغيُّر البتّ j يساوي 50% من أجل جميع قِيَم (i, j) المُمكِنة، ولذلك قمنا بتوسيع برنامج حساب الاحتماليّة probability(i, j, g) ليُجري الحسابات الاحتماليّة من أجل جميع قِيَم (i, j) بما يُعادل (128 × 128 تجربةً). يقوم البرنامج بحساب احتمالات جميع الحالات مع الاحتفاظ بالنتيجة ذات الانحراف الأكبر (maxDeviation) عن القيمة 50% وحفظ الدليلَين (i, j) المُقابِلَين في المُتحوِّلَين (i_md, j_md)؛ وفق مُخطَّط البرنامج المُوضَّح في الشكل 18.

تُشير النتائج باستخدام /100,000/ كتلة نصٍّ صريحٍ أنّه من أجل جميع قِيَم (i, j) فإنّ تغيُّر البتّ i من كتلة النصّ الصريح على دخل كلتا الخوارزميّتين AES, BMC-AES يُؤدّي إلى تغيُّر البتّ j من كتلة النصّ المُشفَّر على خرجها باحتمالٍ قريبٍ من 50%، وأنّ استخدام /1,000,000/ كتلة نصٍّ صريحٍ يجعل الاحتمال

أقرب إلى 50% وبالتالي يكون المعيار SAC مُحقَّقاً؛ كما هو مُبيَّنٌ في الجدول 11.

الجدول (11) نتائج معيار SAC بين النصّ الصريح والنصّ المُشفَّر

	BMC-AES	**AES**
زوج البتّات (i_md, j_md) عبر 100,000 عيّنة	(111, 38)	(109, 122)
احتماليّة تغيُّر j_md بتغيُّر i_md عبر 100,000 عيّنة	%49.269	%49.351
احتماليّة تغيُّر j_md بتغيُّر i_md عبر 1,000,000 عيّنة	%49.9039	%49.9443

لمُقارَنة خصائص الخلط على خرج المُشفِّر باستخدام معيار الانهيار الصارم (SAC) بين المفتاح والنصّ المُشفَّر؛ تمّ تطوير برنامج حساب احتماليّة تغيُّر بتّ الخرج j من كتلة النصّ المُشفَّر عند تغيُّر بتّ الدخل i من المفتاح ونُشير له بالاسم probability(i, j, g)، يستخدم البرنامج كتلة نصٍّ صريحٍ ثابتةً "PlainT_1PlainT_2" وملفّاً بحجمٍ يزيد عن /15.25 MB/ لتوفير /1,000,000/ مفتاحاً كعيِّناتٍ اختباريّةٍ عشوائيّة، ويستقبل رقم بتّ الدخل (i) ورقم بتّ الخرج (j) كمُدخَلات.

يقوم البرنامج بقراءة المفاتيح [Key] من الملف الاختباريّ وتشفير كتلة النصّ الصريح الثابتة للحصول على كتل النصّ المُشفَّر المُقابلة لها [CT]، ومن ثمّ يقوم بقلب البتّ i من المفاتيح [Key] للحصول على مفاتيح جديدةٍ [newKey] وحساب كتل النصّ المُشفَّر المُقابِلة لها [newCT]، وحساب احتمال تغيُّر البتّ j بين الكتل [CT] والكتل [newCT] المُقابِلة لها؛ وفق مُخطَّط البرنامج المُوضَّح في الشكل 19.

يقتضي استيفاء المعيار SAC أن يتمّ التحقُّق من أنّ احتمال تغيُّر البتّ j يساوي 50% من أجل جميع قِيَم (i, j) المُمكِنة، ولذلك قمنا بتوسيع برنامج حساب الاحتماليّة probability(i, j, g) ليُجري الحسابات الاحتماليّة من أجل جميع قِيَم (i, j) بما يُعادل (128 × 128 تجربةً)، يقوم البرنامج بحساب احتمالات جميع الحالات مع الاحتفاظ بالنتيجة ذات الانحراف الأكبر (maxDeviation) عن القيمة 50% وحفظ الدليلَين (i, j) المُقابِلَين في المُتحوِّلَين (i_md, j_md) ؛ وفق نفس مُخطَّط البرنامج المُوضَّح في الشكل 18.

تُشير النتائج باستخدام /100,000/ مفتاحاً أنّه من أجل جميع قِيَم (i, j) فإنّ تغيُّر البتّ i من المفتاح على دخل كلتا الخوارزميّتين AES, BMC-AES يُؤدّي إلى تغيُّر البتّ j من كتلة النصّ المُشفَّر على خرجها باحتمالٍ قريبٍ من 50%، وأنّ استخدام /1,000,000/ مفتاحاً يجعل الاحتمال أقرب إلى 50% وبالتالي يكون المعيار SAC مُحقَّقاً؛ كما هو مُبيَّنٌ في الجدول 12.

الجدول (12) نتائج معيار SAC بين المفتاح والنصّ المُشفَّر

	BMC-AES	AES
زوج البتّات (i_md, j_md) عبر 100,000 عيّنة	(125, 105)	(121, 54)
احتماليّة تغيُّر j_md بتغيُّر i_md عبر 100,000 عيّنة	%50.676	%50.594
احتماليّة تغيُّر j_md بتغيُّر i_md عبر 1,000,000 عيّنة	%50.0718	%50.1342

8- مُناقَشة النتائج وتحليلها:

تُظهِر مُقارَنة الخوارزميّة BMC-AES مع الخوارزميّة AES أنّ الخوارزميّة BMC-AES تُوفِّر أداءً أفضل، حيث أنّ زمن المُعالَجة في التحويل BinMixColumns أقلّ بحوالي 67% من زمن المُعالَجة في التحويل MixColumns، وهو وأقلّ بحوالي 86% من زمن المُعالَجة في التحويل InvMixColumns.

كما أنّ كلفة التنجيز العتاديّ –من حيث عدد بوّابات XOR أو عدد البوّابات المُكافئة GEs– للتحويل BinMixColumns أقلّ بحوالي 58% من كلفة التنجيز العتاديّ للتحويل MixColumns، وأقلّ بحوالي 85% من كلفة التنجيز العتاديّ للتحويل InvMixColumns، وهو ما ينعكس بشكلٍ إيجابيٍّ على زمن الاستجابة؛ حيث أنّ زمن التأخير في مصفوفة التحويل BinMixColumns يكون أقلّ بحوالي 46% من زمن التأخير في مصفوفة التحويل MixColumns.

يتجلّى انخفاض زمن المُعالَجة في الخوارزميّة BMC-AES مُقارَنةً بالخوارزميّة AES على شكل تحسُّنٍ في المَردود بنسبة 27% في عمليّة التشفير ونسبة 96% في عمليّة فكّ التشفير (بالحدّ الأدنى) على منصّة الحاسوب الشخصيّ، وعلى شكل تحسُّنٍ في المَردود بنسبة 22% في عمليّة التشفير ونسبة 93% في عمليّة فكّ التشفير (بالحدّ الأدنى) على منصّات الأجهزة المَحمولة التي تدعم مُعالِجتها تسريع AES. في حين يكون المَردود أفضل بنسبة 53% في عمليّة التشفير ونسبة 240% في عمليّة فكّ التشفير على منصّات الأجهزة المَحمولة التي لا تدعم مُعالِجتها تسريع AES، وبالتالي تُعَدّ الخوارزميّة BMC-AES أكثر مُلاءمةً للاستخدام في الأجهزة المَحمولة مُقيَّدة الموارد.

وفي التنجيزات المُخصَّصة على مُعالِجات 32 بت؛ يكون مَردود الخوارزميّة BMC-AES على مستوى عمليّات التشفير/ فكّ التشفير أعلى بنسبة 33% من مَردود الخوارزميّة AES، مع استهلاك نصف حجم الذاكرة المطلوب لكلتا عمليَّتَي التشفير وفكّ التشفير.

تكون سرعة النشر في الخوارزميّة AES أفضل منها في الخوارزميّة BMC-AES حيث يتحقَّق النشر الكامل في نهاية الجولة الثانية للخوارزميّة AES، في حين أنّه لا

يتحقَّق إلّا في نهاية الجولة الرابعة من الخوارزميّة BMC-AES؛ وهذا يعني تأخيراً بمِقدار جولتين يتمّ تعويضه عن طريق زيادة عدد الجولات بمِقدار جولتين.

تُوفِّر الخوارزميّة BMC-AES خصائص نشرٍ جيّدةً مُماثِلةً للخوارزميّة AES من حيث أثر الانهيار بين النصّ الصريح والنصّ المُشفَّر من جهةٍ وبين المفتاح والنصّ المُشفَّر من جهةٍ أخرى؛ حيث أنّ تغيُّر بتٍّ واحدةٍ من كتلة النصّ الصريح أو المفتاح يُؤدّي إلى تغيُّر 50% من بتّات كتلة النصّ المُشفَّر. كما أنّها تُوفِّر خصائص خلطٍ جيّدةً مُماثِلةً للخوارزميّة AES من حيث تَحقُّق معيار الانهيار الصارم (SAC) بين النصّ الصريح والنصّ المُشفَّر من جهةٍ وبين المفتاح والنصّ المُشفَّر من جهةٍ أخرى.

تُبدي الخوارزميّة BMC-AES مُقاوَمةً لتحليل الشيفرة الخطيّ والفرقيّ وبهامش أمانٍ لا يقلّ عن أربع جولاتٍ حسب حجم الكتلة المُستخدَم، وهي تُبدي مُقاوَمةً ضدّ هجمات الإشباع وبهامش أمانٍ لا يقلّ عن ستّ جولات.

بخلاف الخوارزميّة AES؛ تتمتَّع الخوارزميّة BMC-AES بميزة أمان التنجيز ضدّ هجمات التوقيت حيث تستغرق جميع التحويلات المُستخدَمة في الخوارزميّة وقتاً ثابتاً مُستقلاً عن قيمة المفتاح أو قيمة المُعامِلات، كما أنها تتمتَّع بميزة أمان التنجيز ضدّ هجمات تحليل القدرة البسيط (SPA) كونها تستخدم تسلسلاً ثابتاً من التعليمات.

كما في الخوارزميّة AES؛ يُمكِن توفير الحماية ضدّ هجمات تحليل القدرة الفرقيّ (DPA) عند تنجيز الخوارزميّة عن طريق تطبيق إجراءات حماية التعليمات الفرديّة (مثل تقنيّة مُوازَنة الحمل وتقنيّة تقنيع المُعامِلات) وتطبيق إجراءات إلغاء التزامن.

بناءً على ما سبق؛ تُوفِّر الخوارزميّة BMC-AES أداءً أفضل وكلفةً أخفّ قياساً بالخوارزميّة AES مع الحفاظ على الميزات المرغوبة وفق ما يلي:

- تحسين الأداء من حيث مَردود عمليّات التشفير وفكّ التشفير، وتخفيض كلفة التنجيزات البرمجيّة من حيث حجم الكود بسبب تبسيط تحويل خلط الأعمدة.
- تحسين الأداء من حيث سرعة الاستجابة واستهلاك الطاقة، وتخفيض كلفة التنجيزات العتاديّة من حيث عدد البوّابات المُكافئة (GEs).
- مُلاءمة الاستخدام في نظم تشفير التخزين في أجهزة أندرويد بسبب التحسين الكبير في زمن فكّ التشفير بشكلٍ مستقلٍّ عن توافُر دعم تسريع AES في وحدات المُعالَجة المركزيّة للأجهزة.
- التأثير البسيط على خفّة إجرائيّة إعداد المفتاح والناتج عن توليد مفتاحَي جولة إضافيَّين.
- الحفاظ على هامش أمانٍ مُناسِبٍ ضدّ جميع الهجمات المعروفة، وتحسين مُقاوَمة الهجمات على التنجيز.
- الحفاظ على ميزات الخوارزميّة AES الأساسيّة المرغوبة من حيث البساطة والتنوّع في التنجيزات على المنصّات المُختلِفة ودعم أحجام كتلةٍ إضافيّةٍ 192 ,256 بت.

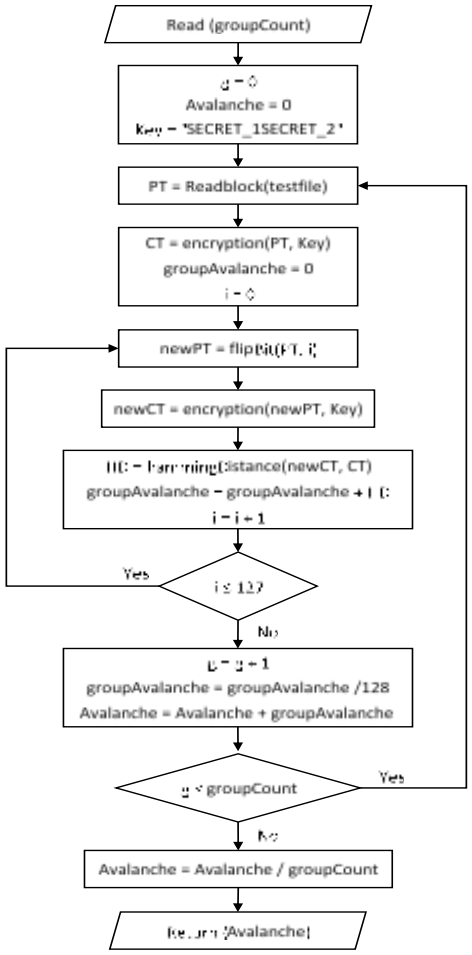
ضمن سياق اختبار الخوارزميّة على منصّات أندرويد؛ تمّ تعديل مُحرِّك التشفير AESEngine في مكتبات التشفير bouncycastle ذات الإصدار 1.70 وجرى تضمينها في تطبيق أندرويد، يُمكِن استخدام المكتبة المُعدَّلة في برامج التشفير الخاصّة وتطبيقات التراسُل الآمِن وغيرها من التطبيقات التي تستخدم التشفير

التناظريّ، ولكنّها غير مُتاحةٍ على مستوى النظام لجميع التطبيقات.

تشمل آفاق التطوير توسيع مجال استخدام الخوارزميّة BMC-AES عن طريق تضمينها في مكتبات التشفير bouncycastle أو Conscrypt[11] التي يُقدِّمهما نظام التشغيل أندرويد كمُزوِّدات أمانٍ تُوفِّر واجهات برمجة التطبيقات (APIs) التي تُتيح وظائف التشفير المختلفة لجميع التطبيقات في نظام أندرويد.

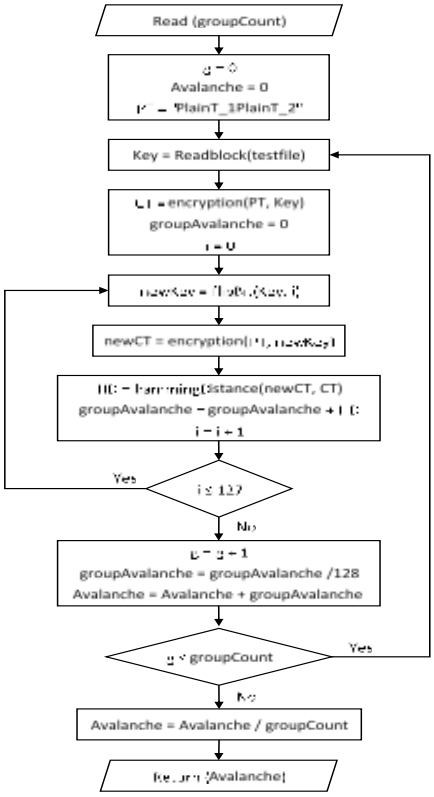
كما يُمكِن استخدام الخوارزميّة BMC-AES ضمن موديول النواة dm-crypt المدعوم في أندرويد 10 والإصدارات الأقدم أو موديول النواة dm-default-key المدعوم اعتباراً من أندرويد 11 والمُستخدَمَين في نظم تشفير التخزين في أندرويد.

الملحق

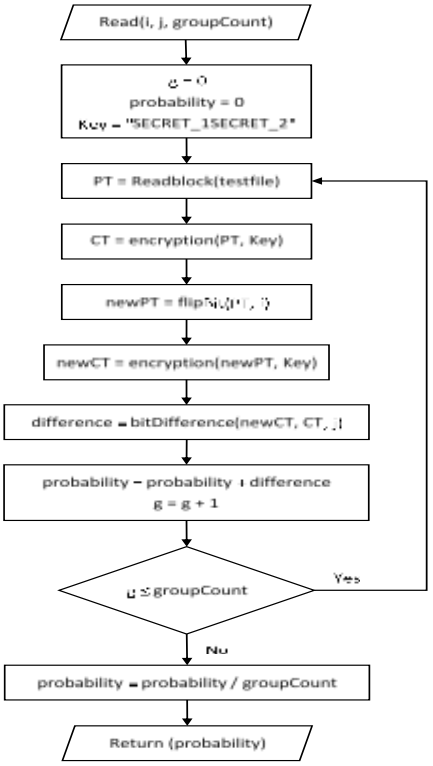


الشكل (15) مُخطَّط حساب أثر الانهيار بين النصّ الصريح والنصّ المُشفَّر

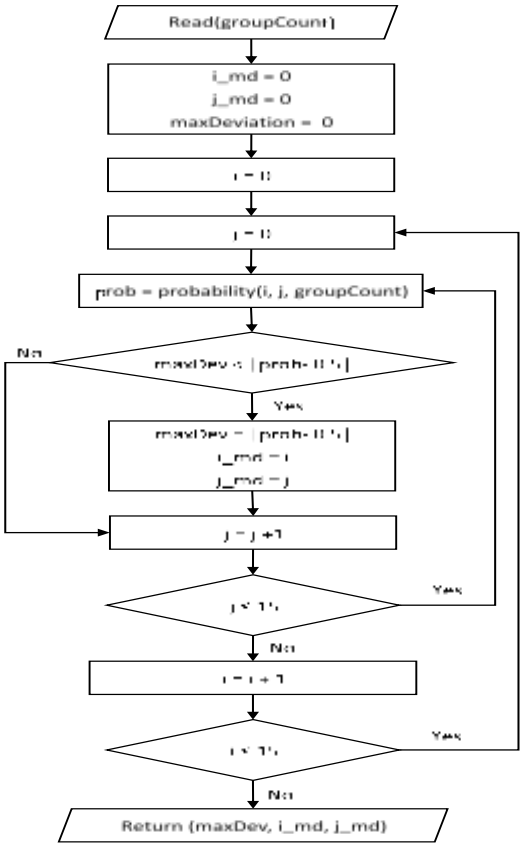
[11] يتوفَّر تنجيز الخوارزميّة AES ضمن مُزوِّد الأمان bouncycastle في نظام أندرويد 8.1 والإصدارات الأقدم، تمّ إيقاف العمل بتنجيز الخوارزميّة AES في مُزوِّد الأمان bouncycastle لحساب مُزوِّد الأمان Conscrypt الذي يُتيح تنجيز الخوارزميّة AES في نظام أندرويد 9 والإصدارات الأحدث.



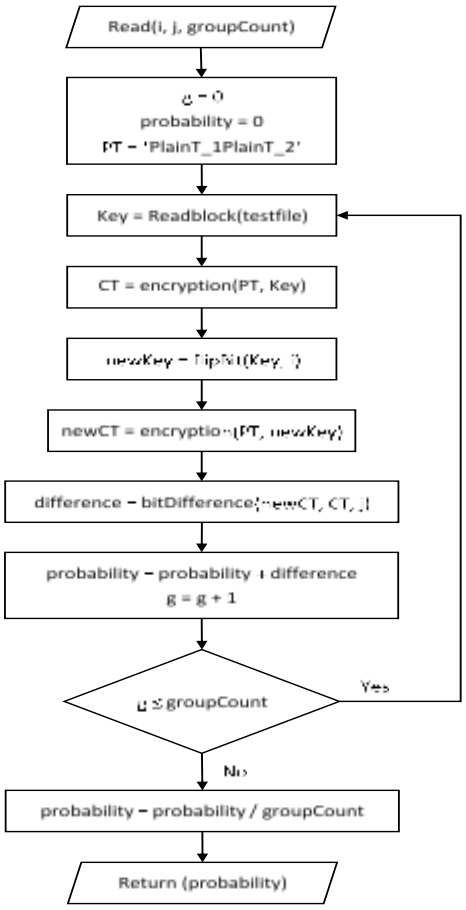
الشكل (16) مُخطَّط حساب أثر الانهيار بين المفتاح والنصّ المُشفَّر



الشكل (17) مُخطَّط حساب احتماليّة تغيُّر بتّ الخرج j بتغيُّر البتّ i من النصّ الصريح



الشكل (18) مُخطَّط التحقُّق من معيار SAC بين النصّ الصريح والنصّ المُشفَّر



الشكل (19) مُخطَّط حساب احتماليّة تغيُّر بتّ الخرج j بتغيُّر البتّ i من المفتاح

Group (3): Key = 32 5F 54 45 52 43 45 53 31 5F 54 45 52 43 45 53 PT = CF 56 73 31 A3 BE EC 1C 6C 38 16 74 73 07 1F 6E CT = (BMC–AES) BE 98 C0 0D F2 D8 CB C7 8D 14 4D 01 05 46 A7 36 (AES) 3D A6 1C C9 0D 54 B6 54 C4 F3 01 2B CA A5 8C E1			
flip bit	(newPT) Plaintext after flipping bit (i)	(newCT) the corresponding ciphertext $\frac{\text{BMC–AES}}{\text{AES}}$	HD $\frac{\text{BMC–AES}}{\text{AES}}$
0	CF 56 73 31 A3 BE EC 1C 6C 38 16 74 73 07 1F 6F	4E 53 5E E3 8A E8 FD 78 3D AA 80 25 98 4F 99 2F 2E 13 0C 86 0D 94 C5 3B 33 A0 D1 59 5A F9 D2 34	$\frac{68}{61}$
1	CF 56 73 31 A3 BE EC 1C 6C 38 16 74 73 07 1F 6C	4E F0 A4 8E EE 97 E9 82 08 05 5F 74 0F D9 7A B0 5A A0 29 91 73 3B 01 D8 0C 34 2A 49 0D 5F 0D 2C	$\frac{55}{68}$
2	CF 56 73 31 A3 BE EC 1C 6C 38 16 74 73 07 1F 6A	3A 1E 6F A0 34 53 C3 29 EC 98 DF 97 7E BA 72 B5 0F CE 05 50 98 0F D0 48 12 21 FF B5 14 E9 78 8B	$\frac{64}{68}$
⋮	⋮	⋮	⋮
125	EF 56 73 31 A3 BE EC 1C 6C 38 16 74 73 07 1F 6E	DC 0F 34 EC 1E BC A1 BA 3A EA F1 ED 20 07 95 BA 38 CE EC 5D 5B 41 43 EA 44 5B 42 B4 F2 CE 5B 01	$\frac{69}{61}$
126	8F 56 73 31 A3 BE EC 1C 6C 38 16 74 73 07 1F 6E	24 2E FC 2A 09 7A FF 62 27 99 F5 67 19 CD 6B 0C 4A BA C6 27 5E 74 7E 5D 83 AC 9D 6F 67 D7 EC CD	$\frac{65}{61}$
127	4F 56 73 31 A3 BE EC 1C 6C 38 16 74 73 07 1F 6E	8C D0 B6 E4 91 9C C4 FE 6F 54 01 D0 BD 64 7E 6A 03 02 C0 C5 C8 F1 D2 90 C7 AF BE 76 90 3C C2 5F	$\frac{56}{65}$
Group avalanche effect = Average([HD]) = $\frac{64.84375\ (\text{BMC–AES})}{62.7578125\ (\text{AES})}$			

الجدول 13: مثالٌ على حساب أثر الانهيار من أجل كتلة نصٍّ صريح

Key = 32 5F 54 45 52 43 45 53 31 5F 54 45 52 43 45 53			
Group	(PT) Plaintext of group	(CT) Ciphertext of group BMC-AES / AES	Ava BMC-AES / AES
1	FE 8B 2B 73 49 AB EE 8B 3B 2A CE F7 58 6E E1 52	92 13 D5 E1 C8 19 2D 1D B8 B6 D9 A2 C2 FF FA 8C / 6A DB A1 F8 64 EF A1 AB D0 A3 FE 1B 91 58 90 58	64.21875 / 64.1328125
2	6F 42 05 37 C5 B9 74 59 B5 9B 07 C7 B0 92 C3 5A	FE DC 97 4B B1 4B 62 13 EC AC 9B B1 99 51 FD 32 / 82 31 8D 2E 1C D8 A3 43 B1 A2 D4 31 4A 47 48 42	65.125 / 63.734375
3	CF 56 73 31 A3 BE EC 1C 6C 38 16 74 73 07 1F 6E	BE 98 C0 0D F2 D8 CB C7 8D 14 4D 01 05 46 A7 36 / 3D A6 1C C9 0D 54 B6 54 C4 F3 01 2B CA A5 8C E1	64.84375 / 62.7578125
4	2A D5 3A 8E 36 0A 46 80 44 68 7A 7A E4 A3 75 9C	15 81 2D AA 5C 79 33 7D 72 BA 21 0C 92 E3 B3 86 / 4D B5 5F 5D AE 58 FD EF CC 40 EF F8 8C A9 37 36	63.21875 / 64.2421875
5	F1 2D 1F 5A 62 E6 44 FC 02 FC 28 33 78 C0 0B 38	5C 4C 8D 67 61 2C BF 5E 7C 6B 83 B1 E5 C0 24 E6 / 49 50 4A CF AF 95 32 44 66 87 6B 48 F3 18 22 3F	63.796875 / 63.5703125
6	A3 F0 B2 C8 01 BD 60 06 11 D3 C6 72 35 F7 00 C9	3D 92 F1 3B C6 5A DC 47 EB 73 9F 6B 07 C4 9E 57 / 63 44 AF 92 DD 25 21 63 91 57 AD 0B FB A8 B1 8B	64.5390625 / 63.8359375
7	E3 59 60 C8 82 75 F1 2F 47 24 BF 39 F0 0A DF 62	0E 2C 09 01 1F AF 99 AF A9 87 10 F0 1F 76 0D 8F / 0A CA 5D B8 F8 70 AD 9F FD 4B 75 2D 77 BC 98 AF	64.453125 / 63.96609375
Avalanche effect = Average([Ava]) = 64.3136161 (BMC-AES) / 63.74776786 (AES)			

الجدول 14: مثالٌ على حساب أثر الانهيار باستخدام عدّة كتل نصٍّ صريح

التمويل: هذا البحث ممول من جامعة دمشق وفق رقم التمويل (501100020595).

References

[1] Abdulgader, A., Ismail M., Zainal N., & Idbeaa T. (2015). **Enhancement of AES Algorithm Based on Chaotic Maps and Shift Operation for Image Encryption.** Journal of Theoretical and Applied Information Technology, Vol. 71, No. 1, pp. 1-12.

[2] AOSP- Docs- Security (Updated in 06/06/2022). **Enabling Adiantum.** Retrieved in 10/06/2022. https://source.android.com/security/encryption/adiantum.

[3] AOSP- Docs- Security (Updated in 06/06/2022). **Encryption.** Retrieved in 10/06/2022. https://source.android.com/security/encryption.

[4] AOSP (2021). **Android Enterprise Security White Paper**, Updated April 2021

[5] Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., et al. (2015). **Midori: A Block Cipher for Low Energy.** International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT), Springer, Berlin, Heidelberg, LNCS (9453) pp. 411-436.

[6] Crowley, P., & Biggers, E. (2018). **Adiantum: length-preserving encryption for entry-level processors.** IACR Transactions on Symmetric Cryptology, Vol. 2018, No. 4, pp. 39–61.

[7] Daemen, J. & Rijmen, V. (2002). **The Design of Rijndael: AES- The Advanced Encryption Algorithm.** Springer, Berlin, Heidelberg.

[8] Dworkin, Morris. (2001). **Recommendation for Block Cipher Modes of Operation, Methods and Techniques.** NIST Special Publication 800-38A.

[9] Gamido, H. V., Sison, A. M., & Medina, R. P. (2018). **Modified AES for Text and Image Encryption.** Indonesian Journal of Electrical Engineering and Computer Science, Vol. 11, No. 3, pp. 942-948.

[10] Sim, S. M., Khoo, K., Oggier, F., & Peyrin, T. (2015). **Lightweight MDS Involution Matrices (Full version).** Cryptology ePrint Archive: Report 2015/258, Original Publication (with minor differences): proceedings of International Workshop on Fast Software Encryption (FSE), Springer, Berlin, Heidelberg, LNCS (9054) pp. 471-493.

[11] Stallings, William. (2017). **Cryptography and Network security Principles and Practice.** Seventh edition global edition, Pearson Education.

[12] StatCounter. **Mobile Operating System Market Share Worldwide.** retrieved in 01, 06, 2022. https://gs.statcounter.com/os-market-share/mobile/worldwide/.

[13] StatCounter. **Tablet Operating System Market Share Worldwide.** retrieved in 01, 06, 2022. https://gs.statcounter.com/os-market-share/tablet/worldwide/.

[14] Wadi, S. M. & Zainal, N. (2014). **High Definition Image Encryption Algorithm Based on AES Modification.** Wireless Personal Communications, Vol. 79، pp. 811–829.

[15] Wenceslao, F. V. (2018). **Enhancing the Performance of the Advanced Encryption Standard (AES) Algorithm Using Multiple Substitution Boxes.** International Journal of Communication Networks Information Security, Vol. 10, No. 3، pp. 496-500.