**الجمهورية العربية السورية**

**المعهد العالي للعلوم التطبيقية والتكنولوجيا**

**قسم الاتصالات**

**العام الدراسي 2020/2019**

مشروع ماجستير

أعد لنيل درجة الماجستير في هندسة شبكات الاتصالات

# التواصل الآمن في المجموعات في شبكات إنترنت الأشياء

تقديم

**م. سامر المحمد**

إشراف

**د. محمد جنيدي**

**د. محمد الشايطة**

30/10/2019

**أهدي هذا العمل**

**إلى جريح أتمنى شفائه بسرعة.....وطني.**

**إلى مثلي وقدوتي في الحياة.....أبي.**

**إلى المنارة التي تنير دربي دوماً..... أمي.**

**إلى من يساندني ويدفعني إلى الأمام..... أخوتي وأخواتي.**

**إلى أهلي وأصدقائي.**

**الجبال قد تصمد أمام الزلازل لكنها لن تصمد أمام قطرات الماء التي تهطل بانتظام في هدوء وتكرار. فاجعل عملك مثل قطرات المياه حتى تتغلب على جميع العوائق. * (حكمة صينية قديمة)**

**كلمة شكر**

**أتقدم بالشكر إلى رئيس قسم الاتصالات الدكتورة شكري مقداد رئيس قسم الاتصالات. كما أخص بالشكر كل من الدكتورين الدكتور محمد جنيدي والدكتور محمد الشايطة اللذان كانا المصباح الذي ينير دربي في هذه الأطروحة من خلال الملاحظات والتوجيه المستمر.**

**سامر المحمد**

# الملخص


شهدت شبكة الإنترنت نمواً سريعاً خلال العقود الثلاثة الماضية، حيث تطورت إلى نموذج جديد يسمى "إنترنت الأشياء"، في إنترنيت الأشياء يتم توصيل جميع الأجهزة الإلكترونية بالشبكة العالمية. ومن أهم احتياجات التواصل في هذه الشبكة العالمية المستقبلية هو ضمان أمن المعلومات المتبادلة فيها.

يعد Datagram Transport Layer Security (DTLS) بروتوكولًا فعليًا لضمان الاتصال الآمن من طرف إلى طرف في إنترنت الأشياء. ومع ذلك، هناك أيضًا حاجة ملحة ومتزايدة إلى وجود اتصال جماعي آمن وفعال وذلك بسبب طبيعة بيئة إنترنت الأشياء التي تكون مقيدة بالموارد في كثير من الأحيان. لقد تم بالفعل اقتراح تعديل على برتوكول DTLS ليدعم الاتصال متعدد البث، لكن حماية الردود(الاستجابات) على طلبات البث المتعدد لم يتم معالجتها بالكامل حتى الآن. علاوة على ذلك، لا يوجد أي تطبيق متاح للعموم لهذا التعديل على بروتكول DTLS.

لقد طبقنا النهج القائم على DTLS للاتصال المتعدد البث باستخدام نظام التشغيل Contiki. وقمنا أيضًا بتطبيق نهج لحماية الاستجابة الفعالة وقمنا بتحليله وتعزيزه في هذه الرسالة.

أخيرًا، قمنا بتقييم النهج التي نتبعها وغيرها من الأساليب على منصة محاكاة للأجهزة المقيدة وهي محاكيِ Cooja المضمن في بيئة Contiki من حيث متطلبات الذاكرة وأداء الاتصالات واستهلاك الطاقة. وتقيم النهج الذي اتبعناه على أساس النتائج التي تم الحصول عليها.

## أسماء البرتوكولات الواردة في البحث

- UDP: User Datagram Protocol
- 6LoWPAN: IPv6 over Low -Power Wireless Personal Area Networks
- IPv6: Internet Protocol version 6
- DTLS: Datagram Transport Layer Security
- Coap: Constrained Application Protocol
- HTTP: HyperText Transfer Protocol
- TLS : Transport Layer Security
- TCP: Transmission Control Protocol
- MAC: Message authentication code
- HMAC: Keyed-Hashed Message Authentication Code
- PRF: Pseudo Random Function
- *PSK:* Phase Shift Keying
- AES: Advanced Encryption Standard
- AEAD: Authenticated Encryption with Associated Data
- CCM: Cipher Block Chaining-Message Authentication Code
- ECDHE: Ephemeral Elliptic Curve Diffie-Hellman
- ECDSA: Elliptic Curve Digital Signature Algorithm
- GSAKMP: Group Secure Association Key Management Protocol
- CBC: Cipher Block Chaining
- ESP: Encapsulating Security Payload
- IKEv2: Internet Key Exchange
- GSPD: Group Security Police Database
- DoS: denial-of-service
- TCP : Transmission Control Protocol
- REST: Representational state transfer

## الكلمات المفتاحية

- اتصال جماعي آمن
- بث متعدد
- إنترنت الأشياء
- اتصال أحادي البث
- برتوكول
- استجابة
- المصافحة

# المحتويات

# قائمة الأشكال

# قائمة الجداول

الفصل الأول

# نظرة عامة عن الموضوع

## 1.1. مقدمة:

من الملحوظ نمو الإنترنت بسرعة خلال الفترة الأخيرة، حيث يتطور من شبكة صغيرة محددة المهام إلى شبكة عنكبوتية ضخمة تضم جميع أنواع الأجهزة الإلكترونية الممكنة [1] . ويمكننا القول أن المجتمع الحديث يتحرك نحو وصل كل شيء وكل شخص بالإنترنت[2]. هذه البيئة التكنولوجية المستقبلية يطلق عليها اسم "إنترنت الأشياء".

**إن إنترنت الأشياء: هو نموذج لشبكة عالمية مترابطة تربط الأشياء التي يمكنها جمع المعلومات من البيئة والتفاعل مع المحيط الفيزيائي المحيط بها وتوفير خدمات مختلفة** [3]. غالبًا ما يكون الكائن أو الشيء، جهازًا ذكيًا مدمجًا مع إلكترونيات أو برمجيات أخرى، أو مع أجهزة الاستشعار والاتصال. وكمثال على هذه الأجهزة يمكن أن تكون على شكل: عدادات ذكية أو مستشعرات طبية أو شبكة ذكية ما أو أنظمة إضاءة[4] . ويمكن أن تكون أيضًا أكثر تقليدية مثل أجهزة الكمبيوتر المحمولة والهواتف الذكية والحواسب الشخصية .كونها متصلة بالإنترنيت، ويمكنها التواصل وتبادل البيانات مع بعضها البعض أو مع أي نوع من أنواع عقد التحكم.

وكثيراً ما تكون الأجهزة في إنترنت الأشياء ذات طبيعة محدودة الإمكانات والموارد، ولا سيما قدرة البطارية المحدودة وقدرات المعالجة، وبالتالي يجب أن تفي الاتصالات بالعديد من المتطلبات التي تعد فيها الكفاءة (الفعالية) واحدة من أهم الاحتياجات.

ومن أجل زيادة الفعالية لابد من استخدام التواصل الجماعي، وخاصة نهج الإرسال المتعدد[5]. يمكن استخدامه في الحالات التي تحتاج فيها العقدة إلى تسليم رسالة متطابقة إلى عدة مستلمين. على سبيل المثال يمكن أن يكون تبديل الإضاءة الخارجية، عندما يتم نقل رسالة "تبديل" متطابقة إلى عدة لمبات ذكية. بالإضافة إلى ذلك، في بعض الأحيان يتوقع المرسل تلقي ردود من بعض المستمعين من مجموعة متعددة الإرسال، على سبيل المثال، في حالة تطلب فيها عقدة التحكم معلومات درجة الحرارة أو الرطوبة من مستشعرات مثبتة في مبنى وتنتظر تلقي البيانات الضرورية فقط.

وهناك حاجة أخرى هامة لإنترنت الأشياء، هي إمكانية الاتصال الآمن من طرف إلى طرف .يتطلب هذا في المقام الأول وجود أنظمة تشفير خفيفة الوزن لحماية البيانات الحساسة، ونظم إدارة مفاتيح مناسبة، والتوثق المتبادل لأطراف الاتصالات[6] . وفي بعض الحالات تكون الحماية ضرورية، على سبيل المثال، البيانات التي تقدمها المستشعرات الحيوية عن التركيب البكتيري للمنتج المستخدم لضمان الجودة المطلوبة في الصناعة الغذائية. تعتبر البيانات في هذه الحالة سرية بشكل واضح لأن انتشارها غير المسيطر قد يضر بسمعة الشركة المنتجة للغذاء. حسابات التشفير عادة ما تكون ذات مهام عالية الدقة، لذا يجب عند تصميم الحلول الأمنية لإنترنت الأشياء أخذ القيود على السلامة في الاعتبار.

قد يؤدي إنشاء جلسة واحدة آمنة متعددة البث مع مرسل واحد ومستمعين متعددين بدلاً من إنشاء جلسات بث أحادي منفصلة فيما بينهم إلى زيادة كفاءة الاتصال بشكل كبير لأن إنشاء جلسة الاتصال يمكن أن يكون الجزء الأكثر استهلاكًا للموارد في العملية بأكملها. علاوة على ذلك، هناك فرق كبير بالنسبة لمرسل يقوم بإجراء تشفير مرة واحدة فقط لرسالة الإرسال المتعدد بدلاً من القيام بها بشكل متكرر لكل رسالة أحادية إلى أعضاء مجموعة الإرسال المتعدد. بالإضافة إلى ذلك، يمكن إعادة استخدام نفس مفتاح التشفير في جلسة آمنة متعددة الإرسال لحماية رسائل الاستجابة، وذلك بسبب تجنب إنشاء جلسات منفصلة لها.

في هذه البحث، نقدم حل من أجل تأمين اتصالات متعددة البث آمنة في إنترنت الأشياء التي على حد علمنا، لا وجود لها حتى الآن في المصادر المفتوحة، وتصميم وتنفيذ آلية حماية لاستجابات أحادية الإرسال على طلبات الإرسال المتعدد، ودمج الآلية في تنفيذ البث المتعدد وتقييم تجريبي للحلول المقدمة.

## 1.2. بعض الخلفيات والمحفزات

تتألف البنية العامة لإنترنت الأشياء من عدة طبقات حيث تكون كل طبقة مسؤولة عن وظائف مختلفة[7]. الطبقة الفيزيائية أو طبقة الإدراك مسؤولة عن تعريف الأشياء وجمع المعلومات. يمكن استخدام بروتوكولات المعيار 802.15.4 في هذه الطبقة[8]. حيث تقوم طبقة الشبكة، التي تتكون من بروتوكولات مثل بروتوكول الإنترنت الإصدار السادس(IPv6)[9]، وبروتوكول مخطط البيانات(UDP)[10]، المستخدم بمعالجة وإرسال المعلومات التي تم الحصول عليها من الطبقات العليا.

وهناك آلية للضغط ( 6LoWPAN)[11] بين الطبقة الفيزيائية وطبقة الشبكة التي تسمح بنقل حزم الـ IPv6 في الشبكات اللاسلكية منخفضة الطاقة عن طريق تقليل حجم رؤوس IPv6 و UDP بشكل كبير. وأخيرًا، توفر طبقة التطبيقات خدمات مختلفة حسب الحاجة. وعادةً ما يُستخدم بروتوكول التقييد (CoAP) [12]، في طبقة التطبيق لإنترنت الأشياء. وهو بروتوكول نقل قياسي للويب يستخدم بين العقد في الشبكات المقيدة. يمكن أن نقول إن CoAP هو إصدار خفيف من (HTTP) [13] نظرًا لأنه يوفر وظائف مماثلة لـ HTTP بما في ذلك التفاعل بين (الطلب / الاستجابة) بين عقد التطبيقات والعقد التي تتضمن الخدمات. الذي سنشرح عنه بشكل مفصل لاحقاً.

ولدى إنترنت الأشياء بالفعل مجموعة واسعة من التطبيقات في مجالات متعددة بما في ذلك النقل، والبيئة الذكية، والمجالات الشخصية والاجتماعية، وأنظمة الأمن [14]. سيؤدي توصيل المزيد والمزيد من الكائنات داخل الشبكة إلى زيادة كبيرة في كميات المعطيات التي تنتقل بين العقد، وسيتطلب ذلك

مساحات تخزينً أكبر للبيانات [15]. من أجل التعامل مع هذا التحدي، يجب تبني حلول لاستخدام أكثر فعالية لعرض النطاق الترددي الحالي.

تعتبر اتصالات المجموعة، وخاصة البث المتعدد (IP multicast) أحد هذه الحلول. ومع ذلك، يجب تكييف الـ CoAP وفقًا لذلك لتحقيق اتصال فعال متعدد البث في الشبكات منخفضة الطاقة. تم تقديم طريقة لاستخدام برتوكول CoAP كبرتوكول نقل مع الإرسال المتعدد[16]. إن من إحدى الميزات التي يمتلكها إنترنت الأشياء هو زيادة الاعتماد على شبكات الطاقة المنخفضة والفقد (LLN) والتي تكون في بعض الأحيان الخيار الوحيد لبعض نطاقات التطبيق. أحيانا يجب استخدام بروتوكول نقل غير موثوق، مثل UDP لزيادة فعالية النقل أو متطلبات تفرضها التطبيقات كالسرعة وغيرها، وهناك برتوكولات أخرى تعالج مسألة ضمان التسليم في الطبقات العليا مثل (CoAP) ومع ذلك، فإنه هناك صعوبة عندما تطلب مسألة تأمين البيانات بين تطبيقات الاتصال.

بروتوكول أمان طبقة النقل (TLS) [17]، هو معيار قياسي فعلي لأمن الاتصالات من طرف إلى طرف عبر الإنترنت في الوقت الحاضر. بالنظر في برتوكول (TLS) نجد أنه مصمم للعمل فوق بروتوكولات نقل موثوقة مثل برتوكول (TCP) [18]، مما يجعل استخدام البروتوكول في IoT إنترنيت الأشياء صعب وغير قابل للتحقيق. ولكن لحسن الحظ، يوجد حل لهذه المشكلة، من خلال استخدام برتوكول يدعى بروتوكول Datagram Transport Layer Security (DTLS) [19].

يعتمد برتوكول DTLS على برتوكول TLS في تصميمه ويوفر ضمانات أمنية مكافئة، ويتم تكييفه للتشغيل مع النقل غير الموثوق. حيث تم الاعتماد على تصميم الـ TLS في تصميم الـ DTLS قدر الإمكان لتجنب الإجراءات الأمنية غير الضرورية، وكان هناك عائقان رئيسيان في برتوكول TLS يمنعان استخدامه في إنترنت الأشياء والتي تم حلها في DTLS:

1. لا يسمح TLS بفك التشفير بشكل جزئي لأن إجراء فك التشفير يعتمد على رقم تسلسلي مشتق من رأس TCP للدفقة كاملة، بينما DTLS يحل هذه المشكلة عن طريق عدم تشفير كامل الدفق وتقديم أرقام تسلسل صريحة للرزم الجزئية تكون هذه الأرقام التسلسلية موجودة في رأسه.

2. إذا فقدت أي من رسائل المصافحة، فإن رسائل المصافحة تفقد الترتيب فيقوم بروتوكول TLS بكسر الاتصال لأنه يقوم على الوثيقة في الاتصال أي أن الرسائل يجب دائما تصل بنفس الترتيب وبدون ضياع، لحل هذه المشكلة، يسمح DTLS بمؤقت إعادة الإرسال ويتسامح مع عدم تلقي رسائل المصافحة بشكل متسلسل.

بشكل مماثل لتصميم الـ TLS، تم تصميم الـ DTLS بحيث يدعم الاتصال الأحادي في نموذج المرسل/المستمع. ومع ذلك، من المرغوب جدًا أن يكون DTLS قادرًا على تأمين اتصال CoAP عبر البث المتعدد لـ IPv6 نظرًا لأن اتصالات المجموعة عرضة أيضًا للهجمات المعتادة عبر الهواء.

ومن الأعمال البحثية التي خاضت في مجال تكييف الـ DTLS لتأمين تواصل المجموعة الآمن نذكر التالي، (Garcia-Morchon et al) اقترحا تكيّف طبقة سجل DTLS لتوفير الحماية المرغوبة [20]. وتتمثل الفكرة الأساسية في تعديل رأس الـ DTLS وبالتالي تجنب إعادة استخدام نفس أرقام التسلسل من قبل مرسلين مختلفين. من خلال تعيين معرفات عشوائية فريدة لكل مرسل، وعلاوة على ذلك، يجب على كل مستمع أن يتابع هذه المعرفات لضمان وصولها بشكل آمن بالإضافة إلى سلامة الرسالة وسريتها. ومع ذلك، في طريقة (Garcia-Morchon et al) لمناقشة حماية طلبات الإرسال المتعدد، لا تعتبر هذه الطريقة من الأساليب الفعالة لتأمين رسائل الاستجابة الأحادية المستقبلة من المستمعين. لأنها تقترح إنشاء جلسات أمان أحادية البث منفصلة وفي هذه الطريقة يتم إفساد جميع الجهود لزيادة

كفاءة الاتصال من خلال عملية توحيد المعلمات الأمنية، التي تم التوصل إليها عن طريق إنشاء جلسة آمنة واحدة متعددة البث بدلاً من جلسات بث أحادية متعددة.

وقد اقترح Tiloca في الآونة الأخيرة نهج واحد فعال من حماية رسائل الاستجابة [21]. لقد اقترح أنه يمكن إعادة استخدام نفس مجموعة الموارد الرئيسية لجلسة الإرسال المتعدد من أجل تأمين رسائل استجابة البث الأحادي. ويتطلب الأمر تغير في رأس DTLS لهذه الرسائل بطريقة تسمح بإدخال معرف فريد لكل مجموعة بث متعدد بحيث يمكن للمرسل تمييز إجابات البث الأحادي من مجموعات بث متعدد .ومع ذلك، فقد اكتشفنا أن بعض ميزات هذا الأسلوب قد تؤثر بشكل كبير على مستوى الأمان في الاتصال. نناقش المشكلات المحتملة بالتفصيل في الفصل الثالث، ونصمم حلاً محسناً يخفف المشكلات المذكورة سابقاً ويلبي متطلبات الفعالية في نفس الوقت.

وأخيرًا، البيئة التي نستخدمها لإدارة الأجهزة المقيدة هي نظام التشغيل Contiki OS المفتوح المصدر [22]، والذي تم تصميمه للاستخدام مع إنترنت الأشياء. يمكن أن تعمل مع مجموعة من الأجهزة اللاسلكية منخفضة الطاقة ولديها مجتمع نشط من المطورين. علاوة على ذلك، يتضمن نظام تشغيل الكونتيكي المكتبات الخاصة بـ CoAP و DTLS من أجل الاتصال الأحادي. نحن سنستخدمها كقاعدة لتنفيذ دعم البث المتعدد.

## 1.3. إشكالية البحث:

تدور مشكلة البحث حول موضوع إيجاد حل آمن، من اجل التواصل في المجموعات في شبكات إنترنت الأشياء وبروتوكول التطبيقات المقيّدة (CoAP) هو بروتوكول تطبيقات إنترنت متخصص للأجهزة المقيدة، كما هو محدد في RFC 7252. حيث لا يوجد حاليًا أي تطبيق لآلية الاتصال المتعدد الآمن لبروتوكول تطبيقCoAP، يقوم تطبيق بروتوكول أمان DTLS المستخدم مع CoAP في إنترنت الأشياء في الوقت الحالي فقط بدعم من الاتصال الأحادي.

والإجابة الإجابة عن الأسئلة التالية، هل يمكن تطبيق اتصال متعدد آمن في إنترنت الأشياء وكيف يمكن تصميمه وتنفيذه؟ علاوة على ذلك، لا يوجد أي نهج متفق عليه بشأن تأمين استجابات أحادية البث لطلبات الإرسال المتعدد. لذلك، كيف يمكن حماية الاستجابات لطلبات الإرسال المتعدد على نحو متساو وكيف يمكن تقييم هذه الصلاحية؟

### 1.4. الهدف من الرسالة:

في هذه الرسالة، سنقوم بالعمل من أجل اتصالات متعددة البث آمنة في إنترنت الأشياء، التي على حد علمنا، لا وجود لها حتى الآن في المصادر المفتوحة، وتصميم وتنفيذ آلية لحماية الاستجابات أحادية الإرسال على طلبات الإرسال المتعدد، ودمج الآلية في تنفيذ البث المتعدد وتقييم تجريبي للحلول المقدمة. ويمكن تلخيص الهدف من هذا البحث بالتالي:

1. دراسة الاتصال ضمن إنترنيت الأشياء وبروتوكولات الأمن.
2. دراسة برمجة النظام الأساسي لإنترنيت الأشياء باستخدام Contiki os.
3. تصميم نهج محسّن لحماية رسائل الاستجابة؛
4. تقيم النتائج، وتوضح مزايا النهج المصمم في الاتصالات الجماعية بشكل عام من حيث متطلبات الذاكرة وأداء الاتصالات واستهلاك الطاقة.

الفصل الثاني

# الدراسة المرجعية

يقدم هذا الفصل المعلومات الأساسية اللازمة والمتعلقة بهذا الموضوع بما في ذلك العديد من التقنيات والبروتوكولات لضمان الفهم الكامل للمحتوى المعروض في هذه الأطروحة.

أولاً، نناقش شبكات إنترنت الأشياء المقيدة، هيكليتها والبروتوكولات المستخدمة. يصف القسم الثاني بروتوكول التطبيقات المقيدة (CoAP) والذي يستخدم عادة كبروتوكول للتطبيقات في إنترنت الأشياء. يتضمن الوصف محددات لبرتوكول (CoAP) في اتصالات المجموعة. وفي القسم الذي يليه نقوم بوصف بروتوكول أمن طبقة النقل (DTLS) وهو بروتوكول الأمان الفعلي المستخدم للنقل من طرف إلى طرف في إنترنت الأشياء. كما نقدم نهجًا قائما لتأمين اتصالات البث المتعدد استنادًا إلى بروتوكول (DTLS). ويتضمن القسم 2.3 منهجًا قائمًا للحماية الكافية لاستجابات البث الأحادي لطلبات الإرسال المتعدد. ويناقش القسم الأخير الأعمال ذات الصلة، مع التركيز بشكل أساسي على IPsec وتطبيقه في الاتصالات متعددة البث في إنترنيت الأشياء.

## 2. دارسة مرجعية عن إنترنيت الأشياء(IOT)

### 2.1. شبكات إنترنيت الأشياء IoT ذات الموارد المحدودة

واحدة من الأفكار الدافعة لإنترنت الأشياء هي ربط شبكات من الأجهزة التي قد يكون لها قدرات محدودة في الذاكرة والتخزين والمعالجة. وتسمى الشبكات التي تكون فيها معظم الأجهزة الموصولة ذات قدرات محدودة بالشبكات منخفضة الطاقة وشبكات الفقد (LLNs- Low-Power and Lossy Networks) [23]. يمكن أن يكون لـ (LLNs) عدة أشكال، يمكن أن تتكون من عدة أجهزة إلى آلاف الأجهزة، وفي

نفس الوقت، تتميز بمعدلات خسارة عالية، وعدم الاستقرار، ومعدلات بيانات منخفضة. ونتيجة لذلك، يجب تصميم التقنيات والحلول الخاصة بإنترنت الأشياء بغرض تقليل استهلاك الطاقة، وكمية المعلومات المرسلة عبر الهواء، وكمية البيانات المخزنة في الذاكرة. ويفرض هذا قيودًا على حجم الرزم المنقولة بدون تجزئة وأنماط الاتصال والقدرة على توفير تسليم موثوق للبيانات. لم يتم تحسين بروتوكولات الإنترنت التقليدية مثل HTTP و TCP للاتصالات منخفضة الطاقة بسبب الرؤوس المطولة والبيانات الوصفية المحمولة ومتطلبات الموثوقية[24]. ونتيجة لذلك، تم تصميم بروتوكولات محددة مناسبة لإنترنت الأشياء، وتم تكييفها لإنترنت الأشياء. قد تختلف البروتوكولات المستخدمة في طبقات إنترنت الأشياء حسب المهمة التي تؤديها أجهزة الشبكة. هنا، سنقدم بعض البرتوكولات التي يشيع استخدامها ويمكن أن يشار إليها كبرتوكولات أساسية في إنترنيت الأشياء[25] . وفي الشكل التالي نعرض البروتوكولات المستخدمة في مشروعنا للتنفيذ والتقييم.

| CoAP |
| --- |
| DTLS |
| UDP |
| IPv6 with 6LoWPAN |
| IEEE 802.15.4 MAC |
| IEEE 802.15.4 PHY |

| HTTP |
| --- |
| TLS |
| TCP |
| IP |
| IEEE 802.3 MAC |
| IEEE 802.3 PHY |

*الشكل 2- 1: بروتوكولات إنترنيت الأشياء IoT المستندة إلى CoAP (على اليسار) وبروتوكولات التقليدية المستندة إلى الويب (على اليمين).*

يصور الشكل مكدس بروتوكولات إنترنيت الأشياء IoT على اليسار ومكدس بروتوكولات الإنترنت التقليدي مع الطبقات الموافقة على اليمين. في مكدس إنترنت الأشياء الذي تم تصويره، يعرف IEEE

802.15.4 PHY و IEEE 802.15.4 MAC [8]، طبقة التحكم في النفاذ والطبقة الفيزيائية بالترتيب، التي تعتمد عليها معظم تقنيات إنترنت الأشياء [24]. لقد تم تصميم عائلة بروتوكولات IEEE 802.15.4 بهدف توفير الطاقة واخذ موضوع زيادة الكفاءة والفعالية في الاعتبار حتى يسمح للأجهزة بالانتقال بين حالات الطاقة المنخفضة وتجنب الطرق المكلفة مثل الإرسال والاستقبال واستماع القنوات [26]. بينما تسهل الطبقات العليا دمج الشبكات اللاسلكية منخفضة الطاقة مع إنترنت الأشياء.

غالباً ما يشار IPv6 over Low -Power Wireless Personal Area Networks (6LOWPAN) في طبقات بروتوكولات إنترنيت الأشياء إلى IPv6 عبر الشبكات اللاسلكية منخفضة الطاقة [27]. وهي طبقة تكيف مستقلة. فهو يحل مسألة كيفية تحميل أطر IPv6 في 802.15.4 وكيفية تنفيذ وظائف اكتشاف الجيران. تتم معالجة المسألة الأولى من خلال تعريف آلية التجزئة وإعادة التجميع، بالإضافة إلى إدخال ضغط الرأس إلى طبقات الداتا(data link) والشبكة وطبقة النقل عن طريق إزالة المعلومات المكررة في طبقة الداتا[28] . ويدعم 6LoWPAN وظيفة لكل من تعيين عناوين الإرسال الأحادي والبث المتعدد وهو أمر مهم لموضوع عملنا.

ويستثني في إنترنت الأشياء عند ربط المليارات من الأجهزة ومعالجتها تقنيات NAT عند البوابات[29] . لذلك، تم استبعاد IPv4 من إنترنت الأشياء، وأعتمد على IPv6 كبروتوكول IP مستعمل فقط على مستوى الشبكة لضمان مساحة عناوين كبيرة، حيث يتمتع IPv6 بمزايا اكثر من IPv4 مثل توجيه أكثر كفاءة ومعالجة حزم أكثر فعالية وتحسينات في الأمن.

وتعتبر طبقة النقل في إنترنت الأشياء المسؤولة عن عملية توصيل الرسائل بين التطبيقات. والـ UDP هو بروتوكول موجه ولا يوفر ضمانات لتسليم الرسائل والحماية المضاعفة. ومع ذلك، فإنه أصبح مفيداً في LLNs لأن بروتوكولات النقل الموثوقة يمكن أن تشبع بسرعة وصلات النطاق الترددي المنخفض،

ونتيجة لذلك، يجعلها غير فعالة للغاية. علاوة على ذلك، يرفع الفعالية من خلال تقليل التعقيد وحجم البروتوكول والآليات المستخدمة فيه.

إن أعلى طبقتين هما طبقتي الشبكة والنقل في مكدس إنترنيت الأشياء، حيث يتوضع بروتوكول الأمان (DTLS) وبروتوكول التطبيق(CoAP) في هاتين الطبقتين. وبما أن هذه البروتوكولات مهمة للغاية لعمل هذه الأطروحة، فإننا سنناقشها بالتفصيل في الأقسام التالية.

## 2.2. بروتوكول التطبيقات المقيّدة (CoAP)

بروتوكول التطبيقات المقيّدة CoAP [12]، هو بروتوكول طلب / استجابة، مصمم خصيصًا للاستخدام في الشبكات منخفضة الطاقة وشبكات الفقد(LLNs). يمكن ترجمته بسهولة إلى HTTP للتكامل مع الويب التقليدي ولكن بالإضافة إلى ذلك يوفر دعم البث المتعدد.

وهنا ملخص للخصائص الرئيسية التي تعالجهاCoAP [24]:

- بروتوكول ويب يحقق متطلبات M2M في البيئات المقيدة.
- يعتمد على الـ UDP مع إمكانية تحقيق موثوقية اختيارية لطلبات البث الأحادي.
- تبادل الرسائل غير متزامن.
- انخفاض التعقيد وحجم الرؤوس والتغليف بين الطبقات.
- يدعم URIو نوع المحتوى.
- بسيط من حيث القدرات التخزينية المؤقتة.
- اكتشاف الموارد الاختياري.
- يُلزم باستخدام مخطط(DTLS) لأمن طبقة النقل.

يشبه نموذج لـ CoAP نموذج المرسل / المستمع الخاص بـ HTTP ومع ذلك، في الاتصال من آلة إلى آلة، تعمل كلتا الطرفيتين عادة كعملاء وخوادم. يمكن تقسيم بنية CoAP بشكل منطقي إلى طبقتين.

**طبقة الرسائل:** تتعامل مع الاتصالات غير المتزامنة عبر UDP من خلال توفير الموثوقية والتسلسلية.

**طبقة (الطلب / الاستجابة):** تعمل على تلبية طلبات الاستجابات ودلالاتها. تتحكم طبقة الرسائل في تبادل الرسائل عبر الشبكة. تشترك الطلبات والردود في تنسيق الرسالة نفسه مع طول رأس ثابت يبلغ 4 بايت، وخيارات ممكنة للمعطيات. تحمل كل رسالة معرّف رسالة يمثل أساسًا للكشف عن التكرارات وموثوقية اختيارية. وعادةً ما يتم زيادته بمقدار معرف واحد لكل رسالة جديدة حتى يتم تجنب إعادة استخدام المعرف لفترة من حياة الرسالة.

هناك أربعة أنواع من الرسائل فيCoAP :

- Confirmable (CON): الرسائل التي توفر الموثوقية، يجب الرد على رسالة CON بـ ACK، وإلا يتم إعادة إرسالها بعد انتهاء دورة حياة الاطار، يجب أن يحمل الإقرار نفس معرف الرسالة ليتم قبوله.

- Non-confirmable (NON): هذه الرسائل لا تحتاج إلى تسليم بشكل موثوق. ومع ذلك، فإنها لا تزال تحمل معرِّف الرسائل، لذا يوجد اكتشاف التطابق للمعرف.

- Acknowledgement (ACK): رسائل تؤكد استلام رسالةCON، كما يمكنهم أيضًا الرد على الطلب.

- RESET (RES): يتم إرسالها إذا تعذرت معالجة رسالة مستلمة.

تحمل رسالة CoAP دلالات الطلب والرد التي هي رمز الطريقة أو رمز الاستجابة ومعلومات اختيارية مثل معرف URI أو نوع المعطيات، يمكن أن تحتوي الرسائل أيضًا على حقل الرمز المميز حيث يكون الرمز المميز عشوائيًا لكل رسالة بخلاف "معرف الرسالة"، يتم استخدام الرموز المميزة لمطابقة الاستجابات للطلبات بشكل مستقل عن الرسائل السابقة حيث قد تصل خارج الترتيب أو تضيع دون سابق إنذار، يوصى بشدة باستخدام هذا الحقل من قبل المعيار لتوفير مستوى أمان أساسي على الأقل على مستوىCoAP. يوضح الشكل 2.2 تنسيق رسالة CoAP.

<table>
<tr><td>Version (2)</td><td>Type (2)</td><td>Token Length (4)</td><td>Code (8)</td><td>Message ID (16)</td></tr>
<tr><td colspan="5">Token (if any)</td></tr>
<tr><td colspan="5">Options (if any)</td></tr>
<tr><td>0xFF</td><td colspan="4">Payload (if any)</td></tr>
</table>

*الشكل 2- 2: تنسيق الرسالة CoAP مع عدد البتات المقابلة بين القوسين*

حيث نلخص حقول رسالة COAP كالتالي:

1. **الإصدار (النسخة):** 2 بت وتكون عدد صحيح بدون إشارة يشير إلى رقم الإصدار للCoAP ، يجب أن تقوم التطبيقات بتعيين هذه المواصفة وملئ الحقل 1أو (01) بالتمثيل الثنائي للإصدار الحالي. ويتم حجز القيم الأخرى للإصدارات المستقبلية. يجب تجاهل الرسائل ذات أرقام الإصدارات غير المعروفة.
2. **النوع:** 2 بت وتكون عدد صحيح بدون إشارة يشير إلى ما إذا كانت هذه الرسالة من النوع القابل للتأكيد (0) أو غير قابل للتأكيد (1) أو إقرار (2) أو إعادة ضبط (3).

3. **طول الرمز:** عدد صحيح بدون إشارة ويتكون من 4 بت .يشير إلى طول حقل الرمز المميز بطول متغير (0–8 بايت). الأطوال 9–15 محجوزة، ويجب عدم إرسالها، ويجب معالجتها كخطأ في تنسيق الرسالة.
4. **الشفرة:** عدد صحيح بدون إشارة ويتكون من 8 بت، مقسّم إلى فئات حيث تكون أول 3 بت (البتات الأكثر أهمية) و5 بت (البتات الأقل دلالة)، موثقة كـ "c.dd" حيث يكون "c" عبارة عن رقم من 0 إلى 7 لـ يكون الحقل الفرعي الأول 3 بت و "dd" رقمين من 00 إلى 31 للحقل الفرعي الثاني 5 بت. يمكن للشفرة أن تشير إلى طلب بالقيمة (0) أو استجابة نجاح بالقيمة (2) أو استجابة خطأ المرسل بالقيمة (4) أو استجابة خطأ في المستمع بالقيمة (5) .(يتم حجز كافة قيم الفئات الأخرى) كحالة خاصة، يشير الرمز 0.00 إلى رسالة فارغة.
5. **معرف الرسالة**: عدد صحيح يتكون من 16 بت في ترتيب بايتات الشبكة. تستخدم للكشف عن ازدواجية الرسائل ومطابقة الرسائل من نوع (ACK) / (RES) إلى رسائل من النوع قابل للتأكيد / غير مؤكدة. يتم تحديد قواعد إنشاء معرف الرسالة والرسائل المتطابقة بشكل مماثل لـHTTP ، ويوفر CoAP أربع طرق أساسية:GET وPUT وPOST وDELETE.

نوضح استخدامها فيما يلي:

- GET لاسترداد المعلومات من الخادم.
- PUT تقوم بتحديثات أو إنشاء موارد على الخادم.
- POST تقوم بإنشاء مورد جديد أو تحديث هدف واحد.
- DELETE حذف مورد معين.

يمكن أن تكون الردود أما نجاح أو خطأ عميل (يعني أن العميل قد تكبد بعض الأخطاء)، أو خطأ داخلي في الخادم عندما يتعذر على الخادم تنفيذ الطلب. اختلاف آخر لـCoAP عن HTTP هو أنه

يدعم أرسال طلبات إلى مجموعة الإرسال المتعدد IP. يتم اكتشاف ذلك بواسطة عدة دلائل لإرسال CoAP، سنناقش ذلك في القسم الفرعي التالي.

### 2.2.1. اتصالات المجموعة في CoAP

يشير الاتصال الجماعي في الـ CoAP إلى علاقة (واحد إلى العديد) بين العقد الطرفية. وعلى وجه التحديد، يمكن لعميل CoAP (الحصول أو تعيين) الموارد على العديد من الخوادم باستخدام برتوكول CoAP من خلال IP multicast، تم توحيد هذه التقنية مؤخرًا بواسطة فريق (IETF) [16].

في حالة إرسال طلب بث متعدد الإرسال، يجب أن يكون نوع الرسالة من النوع غير قابلة للتأكيد Non-confirmable وتوجيهها إلى عنوان IP متعدد البث بدلاً من عقدة CoAP طرفية. قد يتجاهل المستمع طلب الإرسال المتعدد، خاصة إذا لم يكن لديه أي شيء للإجابة عليه، وإذا كان المستمع لا يريد الإجابة، فلا ينبغي عليه الأرسال، على أية حال، لا يتم إرسال الرد على الفور، بل اختيار فترة من الوقت عندما ينوي المستمع الرد. ثم يختار المستمع نقطة عشوائية خلال هذه الفترة الزمنية ويرسل استجابة أحادية البث خاصة به. يتم وضع وقت الانتظار الأساسي لتجنب التصادمات بين استجابات أحادية البث للمستمعين المختلفين. إذا لم يتم استخدام هذا النهج، فقد تصبح التصادمات محتملة وممكنة للغاية، خاصة عندما يصل عدد العقد إلى عدد كبير مثل مائة. أخيرًا، عندما يحصل المرسل على استجابة أحادية الإرسال لطلب الإرسال المتعدد، يجب عليه التحقق فقط من مطابقة الرمز المميز وليس معرف الرسالة.

ويتم تصنيف الأجهزة في الشبكة مثل (LLN) إلى دورين، (1) مرسل و (2) مستمع. قد يكون لدى أي عقدة أحد هذه الأدوار أو كلا الأدوار. يحدد التطبيق (التطبيقات) الذي يتم تشغيله على جهاز بشكل أساسي هذه الأدوار بواسطة استدعاءات الدوال التي يتم تنفيذها على مكدس IP الخاص بالجهاز.

من حيث المبدأ، لا يتطلب المرسل أو المستمع أي إجراءات وصول أو مصادقة مسبقة لإرسال أو الاستماع إلى رسالة بث متعدد [RFC5374] يقوم المرسل إلى مجموعة البث المتعدد IPv6 بتعيين وجهة الحزمة إلى عنوان IPv6 الذي تم تخصيصه للإرسال المتعدد IPv6 يصبح الجهاز مستمعًا عن طريق "الانضمام" إلى مجموعة البث المتعدد IPv6 المحددة بالتسجيل مع جهاز توجيه الشبكة، مما يشير إلى نيته في تلقي الحزم المرسلة إلى مجموعة البث.يمكن لأي جهاز من حيث المبدأ أن يقرر الاستماع إلى أي عنوان بث متعددIPv6 هذا يعني أيضًا أن التطبيقات على الأجهزة الأخرى لا تعرف، أو لا يتم إعلامها، عندما ينضم مستمعون جدد إلى الشبكة.يمكن العثور على المزيد من التفاصيل حول الاتصال المتعدد لبروتوكول IPv6 و CoAP في [I-D ietf-core-groupcomm] لا ننوي في هذا المشروع تعديل أي من بروتوكولات توجيه الاتصالات الأساسية أو الإرسال المتعدد.

## 2.3. برتوكول Datagram Transport Layer Security (DTLS)

### 2.3.1. نظرة عامة عن DTLS

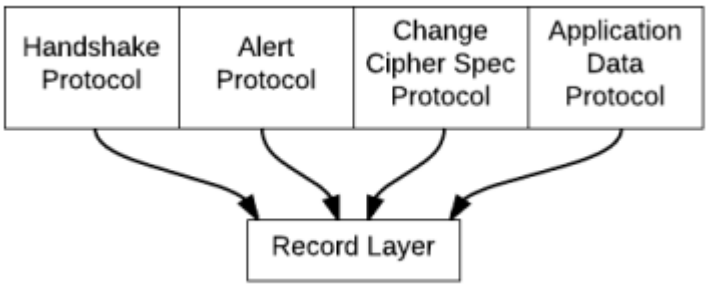
كما يوحي الاسم، فإنDatagram Transport Layer Security [19]، هو بروتوكول أمان اتصالات الشبكة. يمكّن طرفين من إنشاء جلسة آمنة وحماية تبادل البيانات عبر الاتصال المتناظر.

يهدف بروتوكول DTLS، كبروتوكول أمني، إلى تحقيق الأهداف الأمنية الأربعة –**السرية وسلامة الرسائل والمصادقة على الأقران وعدم التنصل.** للقيام بذلك، يستخدم البروتوكول العمليات ذات الصلة بمصادقة الأقران والتشفير وإنشاءMAC، برتوكولات DTLS مرنة في التصميم وتدعم طرق مختلفة للقيام بكل من هذه العمليات. يتعين على الطرفين المتصلين التأكد من أنهما يدعمان نفس الخوارزميات ويتفقان على استخدامهما. كما يحتاج كلاهما أيضًا إلى إنشاء مفاتيح مطابقة للاستخدام مع الخوارزميات المحددة. هذا هو المعروف باسم التفاوض على الجلسة ويتحقق من خلال عملية تسمى المصافحة. نتيجة هذه العملية هي جلسة خاصة آمنة بين الطرفين المتصلين. ونشرح في الملحق كيف تتم عملية المصافحة بشكل مفصل ومخطط الحالة للبرتوكول.

### 2.3.2. مخطط أمن طبقة النقل (DTLS)

مخطط أمن طبقة النقل (DTLS) [19]، هو البروتوكول الأمني الذي يقوم على نقل كتل البيانات فوق برتوكولات غير موثوق بها مثل UDP. ينشأ قناة آمنة بين التطبيقات عند المرسل وعند المستمع الذي يوفر خصوصية الاتصالات. يمكن تقسيم بنية DTLS إلى طبقتين.

- يمكن الإشارة إلى **الطبقة الأولى** على أنها طبقة المصافحة وتتضمن بروتوكولات Handshake و Alert و Cipher Spec Change.
- في حين تتم الإشارة إلى **الطبقة الثانية** على أنها طبقة التسجيل (Record layer) ولا تحتوي إلا على بروتوكول التسجيل كما هو مبين في الشكل التالي.

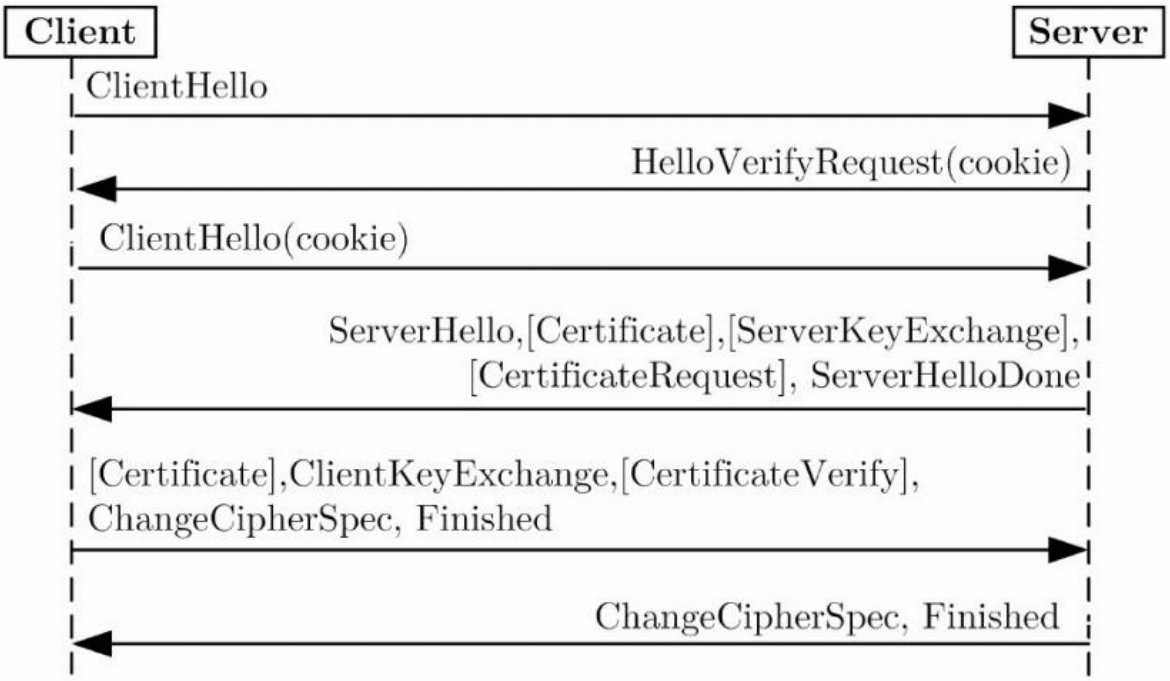



*الشكل 2- 3: البرتوكولات الجزئية لبرتوكول الـDTLS*

الطبقة الأولى مسؤولة عن إنشاء واستئناف جلسات آمنة والتي تشمل التفاوض على التشفير المناسب وتبادل البيانات لتوليد مفاتيح الجلسات. وتكون طبقة التسجيل مسؤولة عن وظائف معالجة البيانات الواردة والصادرة.

في حين يمكن وصف بروتوكولات التنبيه وتغيير الشفرة على أنها بروتوكولات داعمة للإشارات فيما يخص الأخطاء والانتقال إلى التشفير المناسب، ويقوم بروتوكول المصافحة بأداء وظيفة رئيسية في كل

من التوثيق والتفاوض على خوارزميات التشفير والتوقيع الرقمي والضغط. ويتم تبادل البيانات في DTLS كما يبين الشكل التالي:




*الشكل 2- 4: تبادل البيانات في DTLS*

تشبه عملية المصافحة هنا عملية المصافحة في برتوكول طبقة النقل الآمنة (TLS) [30]، ولكن DTLS يقدم العديد من الميزات للتكيف من أجل أن يعمل مع بروتوكول نقل غير موثوق به.

**أولاً**، تتم إضافة عملية تبادل ملفات تعريف الارتباط (cookie) للوقوف ضد هجمات DoS. **ثانيًا**، تضيف DTLS تعديلات على رأس المصافحة لمعالجة فقدان الرسالة وإعادة ترتيبها وتجزئة الرسائل. وهو مدعوم بآلية إعادة إرسال بسيطة. حيث يحدث تعديل في الرأس الرئيسي، بحيث يتم تعيين لكل رسالة مصافحة رقم تسلسلي محدد داخل تلك المصافحة. نتيجة لذلك، يمكن تلقي الرسائل خارج الترتيب، ووضعها في قائمة الانتظار ومعالجتها عندما يكون دورها. إذا لم يتم تلقي رسالة ذات رقم تسلسلي متوقع أثناء الفترة المخصصة، تتم إعادة الإرسال. نتائج المصافحة هي تحقيق لبعض معايير الأمان

بما في ذلك سر رئيسي جديد يتم استخدامه بشكل أكبر لإنشاء مجموعة المفاتيح، وذلك باستخدام تابع عشوائي بدائي متفق عليه ((Pseudo Random Function (PRF)).

إن برتوكول الـ DTLS يعيد استخدام PRF المحدد في TLS والذي يستند على بناء يعرف باسم

Keyed-Hashed Message Authentication Code (HMAC) [31]. هذا البناء هو المعيار الحالي الذي يعتمد في عملية الهاش للمفاتيح مما يعني تطبيق دالة هاش على نص باستخدام مفتاح سري. يمكن استخدام دالات التهشير المختلفة كأساس لـ HMAC، على الرغم من أن SHA256 [32] هو الأفضل. هناك ستة بارامترات للكتابة والقراءة تم إنشاؤها من السر الرئيسي:

1. client write MAC key
2. server write MAC key
3. client write encryption key
4. server write encryption key
5. client write IV
6. server write IV

حيث يتم استخدام مجموعة المعلمات الخاصة بالمرسل لكتابة العمليات بواسطة المرسل ولعمليات القراءة بواسطة المستمع والعكس بالعكس. إما الـ client write IVأو server write IVيتم إنشاؤها فقط من أجل الـ nonce في التوثق المصادق عليه (authenticated ciphers).

ويعمل بروتوكول السجل على حماية بيانات التطبيق باستخدام مفاتيح الجلسة التي تم إنشاؤها أثناء المصافحة. ويتم نقل جميع بيانات DTLS بأطر جزئية التي لا يمكن معالجتها إلا عندما يكون الإطار بأكمله متاحًا. من أجل تجنب التعامل مع التجزئة، يتطلب DTLS تسجيل البيانات في مخطط بيانات واحد. يعرض الشكل 2.5 نسق سجل DTLS.

| Content type<br>1 Byte | Version<br>2 Bytes | Epoch<br>2 Bytes | Sequence number<br>6 Bytes | Length<br>2 Bytes | Ciphertext | MAC |
|---|---|---|---|---|---|---|

*الشكل 2- 5: نسق سجل الـDTLS*

يتم استخدام رقم الحقبة (epoch) في العقد للتمييز بين الجلسات مع حالات تشفير مختلفة. تؤدي سلسلة رقم الحقبة ورقم التسلسل في DTLS وظائف متطابقة كرقم التسلسل في TLS ويتم استرجاعه من الرأس الخاص بطبقة النقل. ومع ذلك، على عكس TLS، يتم إرسال أرقام الحقبة والتسلسل بشكل واضح في عنوان DTLS من أجل توفير الموثوقية في حالة فقدان الحزمة أو إعادة الترتيب أو التكرار. يمكن استخدام تسلسل الأعداد من أجل حماية إعادة الإرسال، وفي حساب الـ MAC في نمط تشفير العداد (counter mode ciphers). في البداية، يتم تحقيق حماية إعادة الإرسال من خلال تذكر الحقبة وتسلسل الأرقام من الأطر الواردة. إذا كانت الرسالة التي تتم معالجتها تحتوي على سلسلة من الأرقام التي ظهرت من قبل، يتم تجاهل هذه الرسالة. ويتم أيضًا تضمين التسلسل في جزء من الإطار الذي تتم مصادقته لتحقيق ضمان التكامل في نمط تشفير العداد، كما يتم ربط رقم الحقبة ورقم التسلسل مع IV للمستمع أو المرسل المشتقة من مجموعة المفاتيح لإنشاء nonce الذي يتم استخدامه كقيمة بدائية للتهيئة في التشفير الكتلي. تتبع عملية التشفير في الـ DTLS مبدأ MAC-then-encrypt. ويمكن تعريف خطوات تشفير الرسائل كما يلي:

1. يتم حساب HMAC للبيانات والرأس

Tag = HMAC (kmac, Header + Data)

2. ثم يتم تمديد سلسلة الرأس والبيانات و Tag

Padded = Padding (Header + Data + Tag)

3. يتم تشفير البيانات الممددة باستخدام مفتاح التشفير واختيار عشوائي للـ IV

Encrypted = Encrypt ([k enc, new random IV], Padded)

4. ثم يتم إرفاق الرأس إلى النص المشفر ويتم إرسال الرسالة الناتجة إلى طبقة النقل.

عندما تتلقى عقدة اطر بيانات، فإنها تفك تشفير البيانات، ثم يتم التحقق من بنية الحشو ويتحقق من تهشير المفتاح كما أنه من المهم ملاحظة أن التشفير التدفقي محظور في DTLS من أجل تحقيق استقلالية التشفير / فك التشفير.

### 2.3.3. اختيار خوارزمية DTLS وإنشاء مفتاح

يتمثل الهدف النهائي لـ DTLS في اتفاق الطرفين المتصلين على الخوارزميات لاستخدامها في تشفير الرسائل وحساب MAC وكذلك لإنشاء مفاتيح مطابقة على كلا الجانبين لهذه العمليات. في رسالة ClientHello، يوفر المرسل قائمة من مجموعات التشفير ciphersuits وفي رسالة ServerHello يضمن المستمع مجموعة التشفير المفضلة لديه من تلك القائمة. بهذه الطريقة يتفق الأقران على خوارزميات مدعومة بالتبادل لاستخدامها في حماية الرسائل خلال الجلسة. يتم إنشاء مفاتيح التشفير من خلال تبادل 3 قيم عشوائية –قيمة عشوائية للمرسل بطول 32 بايت، وقيمة عشوائية للمستمع بطول 32 بايت وسر شبه رئيسي 48 بايت. يتم تبادل القيمتين السابقتين عبر رسائل hello المرسل والمستمع وليست سرية. يتم حماية الأخير من خلال نظام تبادل المفاتيح. تُستخدم القيم الثلاث لإنشاء سر رئيسي طوله 48 بايت .يتم استخدام هذه القيمة، بدورها، مع المرسل وأرقام عشوائية للمستمع لتوليد أكبر قدر ممكن من الموارد الأساسية التي تتطلبها الخوارزميات المحددة للاستخدام مع إنترنيت الأشياء المذكورة سابقاً. يتم استخدام دالة عشوائية زائفة (PRF) لإنشاء السر الرئيسي والمفاتيح.

### 2.3.4. مكونات تطبيقات DTLS

يتكون تنفيذ DTLS عادة من عدة مكونات، وهي:

- مصدر العشوائية

تعتمد عملية توليد المفاتيح في DTLS وقوة المفاتيح الناتجة بشكل كبير على استخدام الأرقام العشوائية. إذا كانت الأرقام العشوائية قابلة للتنبؤ، فستكون للمفاتيح المولدة خصائص أمان ضعيفة. هذا هو السبب في أن تطبيقات DTLS يجب أن توفر طريقة لتوليد بايت عشوائي غير متوقع.

- خوارزميات التشفير

يحتوي هذا المكون على شفرات متناظر وغير متناظر (مفتاح عام) تستخدم للتشفير، وتوثيق الأقران وتبادل المفاتيح .يجب أن يشتمل هذا المكون أيضًا على وظائف لتمكين استخدام هذه التشفير في المخططات ذات الصلة بإصدار DTLS وعمله ومجموعة التشفير المدعومة.

- مخطط الحالةDTLS

هذا المكون مسؤول عن معالجة عملية المصافحة. وهو يتناول انتقالات الحالة وعمليات الإرسال وإعادة الإرسال.

- وظائف ذات صلة بالرسالة

يشير هذا المكون إلى الوظيفة المسؤولة عن توليد الرسائل وتسجيلها ومعالجتها، بالإضافة إلى الرقم التسلسلي والمعالجة الدورية. يتم معالجة تجزئة الرسائل وضغطها وتشفيرها هنا.

- وظيفة التبادل الرئيسية

اعتمادًا على إجراءات تبادل المفاتيح المدعومة، قد تكون هناك حاجة إلى وظائف إضافية. على سبيل المثال، إذا تم استخدام RSA كطريقة للمصادقة وتبادل المفاتيح، فسيتم تضمين مفتاح RSA في شهادة. ستكون هناك حاجة إلى وظائف إضافية لمعالجة هذه الشهادة والتحقق منها.

### 2.3.5. استخدام الـ DTLS و CoAP في أنترنيت الأشياء

في المنشور الخاص بـ "بروتوكول التقيد بالطلب(CoAP)"، يحدد IETF استخدام DTLS لتأمين CoAP وعلاوة على ذلك، تناولت الوثيقة استراتيجيات التبادل الرئيسية الثلاث –المفاتيح المشتركة مسبقًا والمفتاح عام والشهادات.

وفي الـ RFC يرشح PSK–TLS مع AES 128– CCM 8 و TLS ECDHE ECDSA مع AES 128 CCM 8 كبرامج تشفير إلزامية لتنفيذ مفاتيح مشترك مسبقا، والمفتاح العمومي والشهادات، على التوالي. تستفيد مجموعات التشفير هذه من حماية طبقة السجل المدمجة والفعالة باستخدام AES 128 في وضع CCM وفقط 8 بايت MAC. يؤدي تحديد وضع التشفير CCM إلى تقليل حجم مادة المفتاح المطلوبة لكل اتصال حيث يتم استخدام مفتاح واحد فقط لكل من التشفير وإنشاءMAC. يوجد تحليل إضافي أكثر تفصيلاً لعملية التشفير في مشروع "TLS / DTLS Profiles for Internet of Things". [33]

IETF يستكشف تبادل رسائل المصافحة والعمليات ذات الصلة لكليهما. وعلاوة على ذلك، يتطرق المشروع إلى استكشاف للمتطلبات والخطط الإضافية التي ينطوي عليها استخدام أي من مجموعات التشفير. بالإضافة إلى ذلك، يتضمن توصيات تتعلق بجوانب محددة من استخدام مجموعات التشفير وكذلك اقتراحات لاستخدام رسائلhello بشكل عام ولكل مجموعة تشفير لاستخدامها في البيئات المقيدة.

لقد تم تصميم CoAP كبرتوكول مكافئ لـ HTTP في البيئات المقيدة. ويوفر وظائف مماثلة، ولكن أكثر إحكاما. تم تصميم البروتوكول لدعم الوظائف التي تعزز كفاءة الاتصال، مثل الإرسال المتعدد.

لقد تم اختيار DTLS لتنفيذه كبروتوكول أمني لتنفيذه مع تطبيقات CoAP بواسطةIETF وقد تم تعريفه لاستخدامه في بيئات مقيدة وتمت التوصية باستخدام مخططات الحالة لتقليل حجم تنفيذ البروتوكول وأثره في كل من الإرسال والحسابات الزائدة في الشبكة لسوء الحظ، فإن معيار DTLS لا يدعم الإرسال المتعدد أصلاً.توجد طرق أخرى لتأمين الإرسال المتعدد [34] [35] [36] [37]، ولكنها غير مناسبة للاستخدام في البيئات المقيدة، كما هو موضح في [38]راجع المقدمة. علاوة على ذلك، ونظراً لطبيعة البيئات المقيدة، المنطقة المستهدفة لـCoAP، فمن المستحسن أن يتم استخدام بروتوكول أمان واحد لجميع اتصالات الشبكة، بحيث يظل حجم التنفيذ ضئيلاً. لذلك، هناك جهد مستمر في تطوير مخططات عمل لتكييف DTLS بحيث يمكن استخدامها لحماية عمليات الإرسال المتعدد. تم تشكيل مجموعة عمل خاصة في IETF لتنفيذ هذه المهمة. يتم تقديم الحالة الحالية لعملهم في مسودة IETF أمان متعدد قائم على DTLS في بيئات مقيدة ويوفر العديد من المفاهيم المفيدة لمزيد من التطوير في هذا المجال.

أولاً، يتم تحديد العديد من المصطلحات والأدوار والكتل الأساسية للعمل في هذا المجال كما يلي:

- **وحدة التحكم بالمجموعة group controller :**كيان، تتمثل مسؤوليته في إنشاء مجموعات متعددة البث، وتوليد وتوزيع، وكذلك تحديث الاتحادات الأمنية بين أعضاء المجموعة المرخص لهم.
- **المرسل:** جهاز / كيان يرسل البيانات إلى مجموعة بث متعدد .اعتمادا على مجموعة الإرسال المتعدد، يمكن أن يكون هناك واحد إلى 50 مرسلاً في تطبيقات تعتمد على CoAP.
- **المستمع:** جهاز/ كيان، يستمع إلى عنوان IP متعدد الإرسال ويتلقى رسائل واردة.
- **مجموعة الأمان** (SA): هي السياسات ومفاتيح التشفير / فك تشفير توفر وتوجه خدمات الأمان لحماية النقل في الشبكات التي تتبع هذه السياسة. تتكون SA من ثلاثة مكونات، وهي:
  - المعرف: معرفات للمرسل والمستمعين في مجموعة الإرسال المتعدد.
  - السياسة: مجموعات التشفير ومدة حياة مجموعة الإرسال المتعددة المحددة.

- موارد القفل: تستخدم لتوليد المفاتيح.

- **مجموعة أمان المجموعة**(GSA) : مجموعة منSAs، والتي تحدد بشكل جماعي كيفية تأمين الاتصالات في مجموعة الإرسال المتعدد. يتم تزويد أعضاء المجموعة بـ GSA عند انضمامهم إلى المجموعة عن طريق GC.
- **الموارد الأساسية:** نفس المفهوم الخاص بـ SA، تم توسيعه ليشمل جميع SAs المعنية لمجموعة الإرسال المتعدد فقط.
- **حالة اتصال المجموعة**:
هي بنية بيانات بما في ذلك الزوج {Epoch، *Truncated Sequence Number* } والموارد الأساسية المشتقة المحتملة. يمكن لعضو المجموعة تخزين حالات الاتصال المتعدد للمجموعة.
- طلب المجموعة: رسالة الإرسال المتعدد المرسلة بواسطة عقدة المرسل إلى كافة عقد المستمع في المجموعة.
- استجابة المجموعة: رسالة أحادية الإرسال مرسلة بواسطة عقدة المستمع كاستجابة لطلب مجموعة.

### 2.3.6. متطلبات الأمان

متطلبات الأمان التي هي خارج نطاق هذه المشروع والتي يفترض أن تتحقق بالفعل:

1. إنشاء GSA: يجب استخدام آلية آمنة لتوزيع موارد المفاتيح وسياسات أمان الإرسال المتعدد ومعلمات الأمان لأعضاء مجموعة الإرسال المتعدد .يجب إنشاء GSA بواسطة وحدة تحكم المجموعة (التي تدير مجموعة الإرسال المتعدد) بين أعضاء المجموعة. يمكن لجهاز التوجيه أو مستمع بعيد لعب دور وحدة تحكم المجموعة. ومع ذلك، فإن إنشاء GSA يقع خارج نطاق هذه المشروع، من المتوقع أن يشتمل أي نشاط في IETF مخصص لتصميم نظام إدارة مفتاح عام لشبكة LLN على هذه الميزة ويعتمد على [RFC3740] و [RFC4046]و[RFC4535]

2. مجموعة تشفير أمان بيانات الإرسال المتعدد ciphersuite: يجب على جميع أعضاء المجموعة استخدام نفس مجموعة التشفير لحماية أصالة وتكامل وسرية رسائل البث المتعدد. مجموعة التشفير هو جزء من GSA. عادةً ما تكون الأصالة أكثر أهمية من السرية في LLNs. لذلك يجب أن يدعم بروتوكول أمان بيانات البث المتعدد المقترح مجموعات التشفير على الأقل مع MAC فقط (تشفير NULL) ومجموعات التشفير AEAD [RFC5116]. كما يجب دعم مجموعات التشفير الأخرى التي تم تعريفها من أجل أمان سجل البيانات في DTLS.
3. الأمن الأمامي: يجب ألا يكون للأجهزة التي تغادر المجموعة حق الوصول إلى أي جمعية أمان للمجموعات في المستقبل. هذا يضمن أن الجهاز العضو السابق لا يمكنه متابعة فك تشفير البيانات السرية التي يتم إرسالها في المجموعة. كما أنه يضمن أن هذا الجهاز لا يمكنه إرسال بيانات مشفرة و / أو تكامل محمية بعد أن يغادر المجموعة. يجب تعريف آلية تحديث جمعية أمان المجموعة كجزء من نظام الإدارة الرئيسي.
4. السرية التراجعية: يجب ألا يتمكن أي جهاز جديد ينضم إلى المجموعة من الوصول إلى أي من جمعية أمان المجموعات القديمة .يضمن ذلك عدم تمكن جهاز عضو جديد من فك تشفير البيانات المرسلة قبل انضمامه إلى المجموعة .يجب أن يضمن مخطط الإدارة الرئيسي تحديث جمعية أمان المجموعة لضمان السرية السابقة.

متطلبات الأمان التي يلزم تحقيقها بواسطة الحل الموضح في هذا البحث:

- طوبولوجيا الاتصال المتعدد: نحن نعتبر كلا من 1 إلى N (مرسل واحد مع مستمدين متعددين) طوبولوجيا الاتصالات. إن طوبولوجيا الاتصالات 1 إلى N هي أبسط سيناريوهات الاتصال الجماعي الذي يخدم احتياجات LLN النموذجية. على سبيل المثال، في حالة استخدام أتمتة الإضاءة البسيطة، يكون المفتاح هو الكيان الوحيد المسؤول عن إرسال الأوامر إلى مجموعة

من أجهزة الإضاءة. في حالات استخدام أتمتة الإضاءة الأكثر تقدمًا، ستكون هناك حاجة إلى طوبولوجيا اتصالN إلىM.

- يجب أن تدعم حلول الأمان أحجام المجموعات النموذجية المذكورة في مسودة "الاتصال الجماعي من أجل CoAP " [I-D.ietf-core-groupcomm]. حجم المجموعة هو مزيج من عدد المرسلين والمستمعين في مجموعة ذات تداخل محتمل (يمكن أن يكون المرسل أيضًا مستمعًا ولكن ليس بالضرورة أن يكون دائمًا). في حالات استخدام LLN، يكون عدد المرسلين (عادةً أجهزة التحكم) أصغر بكثير من عدد المستمعين (الأجهزة التي يتم التحكم فيها). يغطي حل الأمان الذي يدعم من 1 إلى 50 من المرسلين حجم المجموعة المطلوبة لمعظم حالات الاستخدام بهذا المشروع. يجب أن يكون العدد الإجمالي لأجهزة المجموعة ضمن نطاق من 2 إلى 100 جهاز. يجب تقسيم المجموعات الأكبر من هذه إلى مجموعات متعددة مستقلة صغيرة مثل تجميع أضواء لمبنى لكل طابق.
- سرية بيانات البث المتعدد: يجب تشفير الرسائل المتعددة، حيث أن بعض أوامر التحكم عند إرسالها بشكل واضح يمكن أن تشكل مخاطر خصوصية غير متوقعة على مستخدمي النظام .
- الحماية من هجوم إعادة أرسال بيانات الإرسال المتعدد: يجب ألا يكون من الممكن إعادة أرسال رسالة بث متعدد لأن هذا من شأنه تعطيل أرسال الاتصال الجماعي.
- مصادقة مجموعة البيانات المتعددة ونزاهتها: من الضروري التأكد من أن رسالة الإرسال المتعدد نشأت من عضو في المجموعة وأن الرسائل لم يتم العبث بها من قبل مهاجمين ليسوا أعضاءً. يُستخدم مفتاح المجموعة المتعدد البث المعروف لجميع أعضاء المجموعة لتوفير أصالة رسائل الإرسال المتعدد على سبيل المثال، استخدام رمز مصادقة الرسائل(MAC )، هذا يفترض أن جميع أعضاء المجموعة الآخرين موثوق بهم بعدم العبث برسالة الإرسال المتعدد.

### 2.3.7. أمن الإرسال المتعدد القائم على DTLS

النسخة المعيارية أعلاه من بروتوكول DTLS مناسبة فقط للاتصال الأحادي. ومع ذلك، أدت الحاجة إلى أمن الإرسال المتعدد في البيئات المقيدة إلى تطوير نسخته المعدلة لتواصل المجموعة. اقترحت مجموعة عمل IETF DTLS في بيئة مقيدة (DICE) [20]. يلبي البروتوكول المقدم في هذا الحل متطلبات الأمان الخاصة بسرية البيانات، وحماية إعادة الأرسال، ومصادقة المجموعة، وسلامة المعلومات للمجموعة ضمن الحجم وطوبولوجيا الاتصال المتعدد الإرسال المحددة للمجموعة. وسنلقي الضوء بشكل مفصل على الحلول المقدمة في مجال تأمين الاتصال الجماعي وحماية الاستجابات.

#### *2.3.7.1. حل مجموعة عمل IETF DTLS في بيئة مقيدة (DICE)*

حيث يقترح الحل المقدم في [20] أن يتم إنشاء جلسة مجموعة DTLS دون إجراء عملية مصافحة منتظمة. بدلاً من ذلك، يقوم جهاز تحكم للمجموعة بتوزيع معلمات ارتباط أمان المجموعة (GSA) [39]، التي تتكون من معلمات الأمان بين أعضاء المجموعة للإرسال المتعدد. وتتضمن معلمات الأمان مجموعة المفاتيح، مجموعة التشفير المستخدمة، خوارزمية MAC وخوارزمية الضغط، سواء كان نهاية الاتصال مرسل أو مستمع، وقيم اختيارية أخرى. كما هو الحال في جلسة DTLS التقليدية، وتتكون حالات الكتابة والقراءة الحالية لـ GSA من ستة عناصر رئيسية التي ذكرناها سابقاً في الفقرة 2.4.2 يمكن إنشاؤها من المفتاح الرئيسي، ويمكن القيام بعملية توزيع المفاتيح باستخدام مخططات إدارة مفاتيح موحدة للشبكات المقيدة مثل GSAKMP [40]، وبالتالي يشترك جميع أعضاء المجموعة في نفس موارد أمان المجموعة بحيث يمكن للأجهزة المرسلة أن تعنون رسائل الإرسال المتعدد إلى أجهزة الاستماع. **وتجدر الإشارة إلى أنه في ظل هذه الظروف**، لا يمكن ضمان صحة المصدر لرسائل الإرسال المتعدد المرسلة داخل المجموعة.

يشتمل تنسيق سجل DTLS التقليدي على حقول رقم الحقبة والتسلسل التي تضمن التحديث وتوفير الكشف على تكرار الرسائل المرسلة. رقم الحقبة يتم تثبيته بواسطة مصافحة DTLS لجلسة معينة ويتم تهيئة رقم التسلسل إلى 0 ويتم زيادته بمقدار واحد لكل أطار جديد يرسله المرسل.

**في السيناريو حيث يقوم العديد من المرسلين بتقديم نفس GSA ومشاركتها**، قد يكون **لديهم نفس أرقام التسلسل** لرسالتهم الخاصة بالبث المتعدد. وكنتيجة لذلك، قد يتم تجاهل الرسائل بواسطة عقد الاستماع كرسائل مكررة. علاوة على ذلك، تعريف DTLS من أجل استخدام خوارزميات تشفير التوثق (AEAD) مثل AES-CCM [41] و AES-GCM [42] والتي تستخدم **رقم شبه عشوائي (nonce)** لتحقيق أمن التشفير. يمكن كسر هذا الأمان تمامًا في حالة إعادة استخدام الرقم شبه العشوائي (nonce) ووجود مفتاح واحد ضمن المجموعة وهذا ممكن الحدوث في سيناريو المرسلين المتعددين لأن جميع المرسلين يطبقون نفس المفتاح على جميع رسائل الإرسال المتعدد المتبادلة داخل المجموعة. في هذه الأنواع من التشفير، يتم استخدام مفتاح واحد للتشفير والتوثيق. ففي AESCCM، يتم توفير التشفير باستخدام خوارزمية التشفير المتقدم (AES) في وضع العداد [43]، ويتم ضمان التوثيق باستخدام Cipher Block Chaining (CBC) MAC.

**يمكن أن يكون الحل المحتمل للتحديات** التي تمت مناقشتها هو مزامنة جميع مرسلي المجموعة لتجنب إعادة استخدام أرقام التسلسل. ومع ذلك، هذه مهمة صعبة، وخاصة في الشبكات المقيدة.

**ويتمثل النهج الثاني** الذي يتم اعتماده في تضمين معرّف هوية مرسل فريد في رقم التسلسل، مما يوفر لكل مرسل مساحة ترقيم منفصلة غير متداخلة. يتم ذلك بواسطة وحدة التحكم في المجموعة التي تقوم بتعيين معرف مرسل فريد لكل جهاز إرسال في المجموعة. يحصل جميع أعضاء المجموعة على قائمة بمعرفات المرسلين وهم أكثر قدرة على فرز رسائل الإرسال المتعدد المرسلة. كما يبين الشكل 2-5 نسق إطار DTLS للإرسال المتعدد مع تعين معرف للمرسل.

| Content type 1 Byte | Version 2 Bytes | Epoch 2 Bytes | Sender ID 1 Byte | Trunc_ seq_number 5 Bytes | Length 2 Bytes | Ciphertext | MAC |
|---|---|---|---|---|---|---|---|

*الشكل 2- 6: نسق سجل الـ DTLS في الإرسال المتعدد.*

### 2.3.7.2. حماية الردود الأحادية على رسائل البث المتعدد

تقول مجموعة عمل Dice [20]، أنه يجب تأمين رسائل الاستجابة الأحادية الفردية، وتقترح حمايتها من خلال إنشاء جلسة DTLS منفصلة لكل مستمع مرسل متعدد الإرسال. هذا له العديد من العيوب. أولا، إما أن المرسل أو المستمع يجب أن يبدأ مصافحة DTLS؛ أو كلاهما يجب أن يؤدي المصافحة ويستهلكا موارد مقابلة لحماية البيانات التي تنتمي إلى جلسة أخرى. علاوة على ذلك، يجب على المرسل تنفيذ كمية من عمليات المصافحة مساوية لعدد أعضاء مجموعة الإرسال المتعدد. وفقًا لمعيار الاتصال الجماعي في LLNs [44]، قد يصل عدد أعضاء مجموعة البث المتعدد الذين يصلون بشكل آمن إلى مائة، لذا، إذا كان مرسل الإرسال المتعدد جهازًا مقيَّدًا ويجب عليه إجراء المصافحة مع كل عضو في المجموعة، قد يستنزف موارده بالكامل ويمنع الاتصال تمامًا. بالإضافة إلى ذلك، لا يُطلب من المرسل معرفة جميع مستمعي الإرسال المتعدد في المجموعة مسبقًا نظرًا لأنها مهمة وحدة التحكم في المجموعة. هذا يعني أن المرسل لا يستطيع القيام بالمصافحة مقدمًا، وأنه قادر على تنفيذها فقط عندما يطلب المستمعون ما قد يحدث في فترة زمنية قصيرة جدًا بعد الحصول على طلبات الإرسال المتعدد. ومع ذلك، وفقا لتوصيف DTLS، إن المرسل هو الذي يجب عليه بدء عملية المصافحة، وليس المستمع. وتحدث مشكلة في أدوار الاتصال عندما يصبح مرسل CoAP ليكون مستمع DTLS والعكس بالعكس.

لحل المشكلات المذكورة أعلاه، يقترح Tiloca [21] امتدادًا للمخطط الوارد في القسم 2.3.7.1. الهدف من التمديد هو إعادة استخدام معلمات الجلسة وموارد تجميع المفاتيح لتأمين رسائل الاستجابة. النهج هو على النحو التالي. عند إنشاء مجموعة الإرسال المتعدد، تقوم وحدة تحكم المجموعة بتعيين معرف

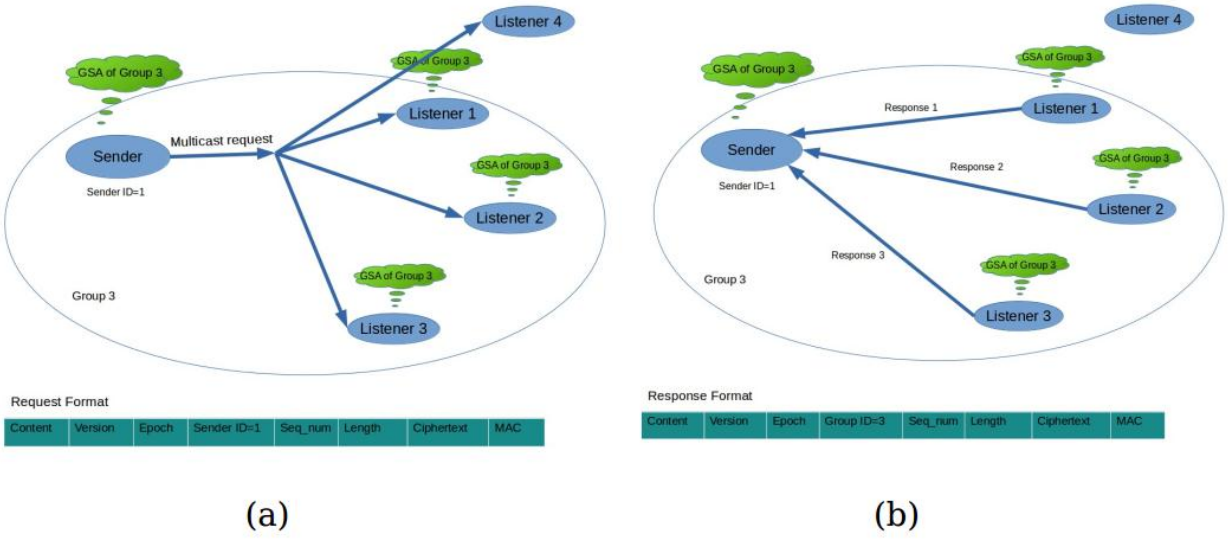
مجموعة لها وتوفر القيمة لكافة أعضاء المجموعة. سيتم تضمين معرف المجموعة في رأس DTLS لرسائل استجابة البث الأحادي بدلاً من معرف المرسل، لذلك في تنسيق السجل الناتج يسبق رقم التسلسل المقتطع. يتم تقديم تنسيق رسالة استجابة البث الأحادي في الشكل 2.6.

| Content type<br>1 Byte | Version<br>2 Bytes | Epoch<br>2 Bytes | Group ID<br>1 Byte | Trunc_ seq_number<br>5 Bytes | Length<br>2 Bytes | Ciphertext | MAC |
|---|---|---|---|---|---|---|---|

*الشكل2-1 : تنسيق سجل استجابة DTLS أحادية الإرسال لطلب الإرسال المتعدد*

يحتفظ المستمعون بجدول حيث يرتبط كل معرف مجموعة بمعرف مناظر للمرسل في مجموعة الإرسال المتعدد. إذا استلم المستمع طلبًا متعددًا محميًا ويعتزم إرسال استجابة أحادية مرة أخرى، يسترد المستمع معرف مجموعة مقترنًا بمعرف المرسل من الطلب ويقوم بإعداد استجابة للتنسيق المعروض في الشكل 2.6، باستخدام Server write encryption key وServer write MAC key و Server write IV عند استقبال حزمة الإرسال الأحادي، يقوم مرسل الإرسال المتعدد بفحص ما إذا كانت هناك جلسة DTLS أحادية الإرسال مع العقدة المحددة على منفذ محلي تم استلام الحزمة منه. إذا كانت هناك جلسة من هذا القبيل، تتم معالجة الحزمة وفقًا لـ DTLS التقليدي؛ وبخلاف ذلك، يفترض مرسل الإرسال المتعدد أن هذه الاستجابة أحادية الإرسال لطلبات الإرسال المتعدد التي تم إرسالها مسبقًا وتقوم بمعالجة الحزمة كسجل للنسق الوارد في الشكل 2.6. يسترد المرسل قيمة معرف المجموعة ويبحث في جدول الاتصال الخاص به لمفاتيح الجلسة والمفاتيح المقابلة. في حالة العثور على الاتصال، يتم فك السجل، ويتم التحقق من MAC ويتم التحقق من رقم التسلسل لاكتشاف فيما إذا كان مكرر أم لا.

يوضح الشكل 2.7 سيناريو محتمل مع مرسل مجموعة واحد وثلاثة مستمعين للمجموعة.




*الشكل 2- 7: سيناريو تواصل المجموعة مع طلب الإرسال المتعدد (أ) والاستجابات الأحادية (ب)*

في السيناريو، تتكون المجموعة من مرسل واحد وثلاثة مستمعين يتشاركون جميعًا معلمات الأمان كجزء من GSA. كما يشترك المستمع رقم 4 في عنوان الإرسال المتعدد حتى يمكنه استقبال الطلبات ولكنه لا يمتلك على موارد المفتاح المشترك، وبالتالي لا يمكن فك تشفير الطلبات. نلاحظ إن تنسيقات الطلب والاستجابة تختلف فقط في حقل المعرف ID حيث يضمن المرسل حقل معرف المرسل، في حين أن المستمعين يضمنون معرف المجموعة.

## 2.4. أعمال ذات صلة

في هذا المشروع، نستخدم DTLS لتوفير أمن الاتصالات في إنترنت الأشياء، وبناء الحل لأمن الاتصالات متعدد الإرسال على أساس DTLS. ومع ذلك، توجد نهوج أخرى لأمن الاتصالات من طرف إلى طرف، والتي نناقش أحدها في هذا القسم. كما نوضح سبب كون الحل القائم على DTLS أفضل في ظل ظروف هذا المشروع.

### 2.4.1. بروتوكول أمن الإنترنت(IPsec)

بروتوكول أمان الإنترنت (IPsec) [45]، هو بنية أمان مستخدمة على نطاق واسع لتوفير قناة آمنة من نظير إلى نظير عبر الإنترنت. إنها بالأحرى تتكون من ثلاثة بروتوكولات مستقلة.

يوفر برتوكول(IP Encapsulating Security Payload) (ESP) [46] السرية ومصادقة المصدر وسلامة حمولة رزم IP. ويستخدم (IP Authentication Header)(AH) [47] لتوفير المصادقة وضمان سلامة الحمولة و رأس الحزمة. كلا البروتوكولين أيضا يوفر حماية ضد هجمات إعادة الإرسال ويمكن استخدامها في أنماط مختلفة، مع بعض الميزات. علاوة على ذلك، يمكن دمج البروتوكولات معًا للمصادقة على الرأس والحمولة وفي عملية التشفير، ولكن نادرًا ما يتم استخدام هذا النهج بسبب تعقيد التنفيذ والكلفة. البروتوكول الثالث، تبادل المفاتيح (Internet Key Exchange) (IKEv2) [48]، هو المسؤول عن إجراء التوثيق المتبادل وتأسيس موارد القفل ومعلمات الأمان الأخرى. يشار إلى هذه المعلمات مجتمعة باسم رابطة الأمن (SA). يمكن تنفيذ IPsec في المضيف، غالباً ما يكون البرتوكول أكثر تضافراً مع نظام التشغيل (OS)، أو مع جهاز توجيه [49].

إن برتوكول IPsec، الذي يعمل على طبقة الشبكة، يقوم على ضمان سلامة ونزاهة رؤوس طبقة النقل، والتي لا يمكن إجراؤها باستخدام بروتوكولات تعمل فوق طبقة النقل مثل DTLS. كونه مهم بالنسبة لعملنا، تم تمديد IPsec لاستخدامه في البث المتعدد IP [50]. وبشكل خاص، يجب أن يخصص IPsec عنوان IP محدد لبروتوكول الإنترنت في حالة الإرسال المتعدد كعنوان وجهة، ويجب الإشارة إلى كل من عنوان الوجهة والمصدر بشكل صريح من أجل التوجيه الصحيح. كما تقديم SA نماذج الدعم الخاصة بنوع الإرسال المتعدد فضلاً عن سياسات أمن قواعد البيانات للمجموعة (GSPD) لدعم كلا من الإرسال الأحادي وحزم الإرسال المتعدد الجديدة. وأخيرًا، يتم توفير المصادقة وسلامة المعلومات على مستوى المجموعة بمفتاح مصادقة مشترك.

على الرغم من العديد من المزايا، فإن IPsec لها أيضًا عدة عيوب، خاصة عند الاستخدام في إنترنت الأشياء. أولاً، يضيف عبئاً كبير إلى حزم IP وذلك لأنه يفرض رؤوس طويلة نوعا ما إلى الرزم، في حين تفرض إنترنت الأشياء قيوداً صارمة على حجم الرزمة. يأتي زيادة طول الرؤوس بشكل أساسي من الرؤوس الإضافية التي يجب تضمينها في كل مخطط بيانات لكل برتوكول من البرتوكولات الثلاث التي ذكرناها سابقاً المكونة للـ IPsec. من أجل تخفيف هذه المشكلة، [51] يقترح حلاً لدمج IPsec مع LoWPAN6، مما يعني إمكانية ضغط رأس IPsec بواسطة تقنيات LoWPAN6. عملياً، تم تنفيذ الحل وتم إظهار انخفاض ملحوظ في حجم الحزمة. ومع ذلك، فإن استهلاك الطاقة وزمن المعالجة لا تزالا مرتفعة إلى حد كبير.

علاوة على ذلك، فإن IPsec لا يقوم إلا بتأمين قناة أمنة إلى الآلة أو الجهاز، بينما يسمح DTLS بالتحكم في الوصول بشكل أكثر دقة من خلال إمكانية التواصل بين التطبيقات. جانب آخر يتعلق بأمن أسماء النطاقات (DNS) [52]. يسمح DTLS باستخدام اسم DNS بشكل آمن لتعريف مستمع، وهذا غير مدعوم في معظم تطبيقات IPsec. ويعد هذا أمر مهم لأن عنوان IP نادرًا ما يكون معروفًا مسبقاً لذا يجب إشراك DNS في المصادقة.

وأخيراً، تشجع (IETF) بشدة لاستخدام DTLS في مقترحاتها الأخيرة. وقد يساعد ذلك في توحيد معماريات الأمن في إنترنت الأشياء اعتماداً على DTLS، وبالتالي تجنب المشاكل المحتملة المتعلقة بالتشغيل البيني.

# تحليل الآليات الحالية والحل المقترح في تصميم آلية حماية الاستجابة

في هذا الفصل، نقوم بتحليل الآليات الحالية لحماية الاستجابة في الاتصالات الجماعية، واقتراح التحسينات عليها في حالة الإمكان. في التحليل، نوضح الأسباب الكامنة وراء نقاط الضعف المكتشفة وبيان الهجمات المحتملة والسيناريوهات المقابلة.

## 3. تحليل الآليات الحالية

تهدف الآلية المقترحة في [21] إلى إعادة استخدام الموارد من مجموعة الأمان (GSA) لحماية استجابات البث الأحادي لطلبات الإرسال المتعدد. وبالتالي، يمكن تجنب إنشاء جلسة DTLS منفصلة من عقدة إلى عقدة. تقدم الآلية معرّف المجموعة الذي سيتم تعيينه إلى كل GSA وتقترح تضمين هذا المعرّف كحقل جديد في رأس DTLS. إذا كان مرسل الإرسال المتعدد يتلقى رسالة أحادية من عقدة لا يحتفظ بها بجلسة DTLS من عقدة إلى عقدة، فإنه يفترض أنه يمكن أن يكون هناك جلسة DTLS للمجموعة، ويسترد لاحقاً بايت يناظر حقل معرف المجموعة. يتحقق مرسل الإرسال المتعدد من وجود GSA مع معرف مجموعة يساوي الرقم المسترجع. في حالة وجود مثل هذا GSA، يستخدم المرسل موارد المجموعة لفك تشفير الرسالة والتحقق منها.

ويركز النهج الموصوف في الغالب على تحقيق كفاءة اتصال عالية ويفعل ذلك بشكل جيد من خلال تجنب إنشاء جلسة آمنة جديدة حيث أننا ذكرنا سابقا أن عملية المصافحة هي العملية الأكثر استهلاكاً للموارد في بروتوكول DTLS. ومع ذلك، فقد اكتشفنا أن هذا النهج يغفل بعض الاعتبارات الأمنية،

وبالتالي يبدو عرضة للهجمات الضارة. علاوة على ذلك، نحن نصف الأسباب الكامنة وراء الهجمات الفعلية التي يمكن تنفيذها بنجاح.

### 3.1. نقاط الضعف الأمنية المكتشفة والشروط الضمنية

كما هو موضح في القسم 2.3.2، تتكون الموارد الأساسية في DTLS من ستة عناصر مشتقة من سر رئيسي. هي Client Write - Server Write MAC Key- Client Write MAC Key Server Write IV- Client Write IV- Server Write Encryption Key- Encryption key جميع هذه العناصر متمايزة ولا يمكن لخصم محتمل أن يستفيد من حقيقة أنها تأتي من نفس المصدر.

في سيناريو اتصال المجموعة الآمنة المستندة إلى DTLS التي اقترحها [20] [21]، يستخدم جميع المرسلين والمستمعين أو العملاء والخوادم في مصطلحات أخرى لإجراء عمليات الكتابة والقراءة نفس المعلمات. ووفقًا لـ [21]، لحماية رسائل الاستجابة، يقوم المستمعون بتضمين معرف المجموعة في رأس السجل ليتم التميز بين المجموعات عند المرسل. وعلى نقيض مرسلي الإرسال المتعدد الذين يحملون جميعًا معرّفات مرسل فريدة ويتم تضمينها في رأس السجل، يحتفظ المستمعون بنفس رقم معرف المجموعة لأن هناك معرّفًا واحدًا للمجموعة. كما هو موضح في الشكل 2.6، والان وبالعودة إلى بنية إطار الـDTLS وبملاحظة ان رأس سجل DTLS لاستجابات البث الأحادي لطلبات الإرسال المتعدد تشتمل على المحتوى وإصدار البروتوكول ورقم الحقبة ومعرف المجموعة ورقم التسلسل المقتطع وقيم طول البيانات.

من الواضح أنه من المرجح أن تكون هناك تطابق لحقول المحتوى، والإصدار، وحقل معرف المجموعة بالنسبة لجميع ردود المستمعين من نفس المجموعة. المحتوى هو بيانات التطبيق، الإصدار يمكن أن يكون 1.2 لـ DTLS. وبالوقت ذاته، تتم زيادة رقم الحقبة لكل حالة تشفير جديدة ويتم زيادة رقم التسلسل لكل سجل يتم إرساله. لكن، يتم تعيين كل من رقم الحقبة ورقم التسلسل **بالقيمة صفر في بداية اتصال**

**المجموعة** بحيث يبدأ كل مستمعي المجموعة بنفس القيم، وعلاوةً على ذلك، يتم زيادتها بطريقة موحدة. نتيجة لذلك، من المحتمل أن ينتهي المستمعون برؤوس متطابقة لسجلات الـ DTLS (لا يأخذ طول البيانات بعين الاعتبار لأن المرسل لا يملك أي معلومات مسبقة عنه ويقبل أي قيمة لطول البيانات).

علاوةً على ذلك، قد يؤدي مشاركة نفس المفاتيح والجزء الصريح من الشعاع الابتدائي IVs من قبل المستمعين إلى إعادة استخدام الزوج {Key، nonce} في أنظمة التشفير المصادقة. كل هذا يبدو أنه نقاط ضعف تؤدي إلى إمكانية تنفيذ الهجمات الموضحة أدناه.

### 3.1.1. إعادة الأرسال

عندما يتم إرسال رسالة باستخدام DTLS، فإنها مجزأة ومضغوطة اختياريًا وموقعة بمفتاح (HMAC) وأخيراً مشفرة. تتم العمليات بترتيب محدد. حيث يتم البدء في تطبيق موارد القفل في عملية حساب HMAC. ويتم إنشاء HMAC كالتالي:

HMAC (Server MAC write key, epoch+groupID+trunc_seq_number+ Content type+Protocol version+ Data length+ Data)

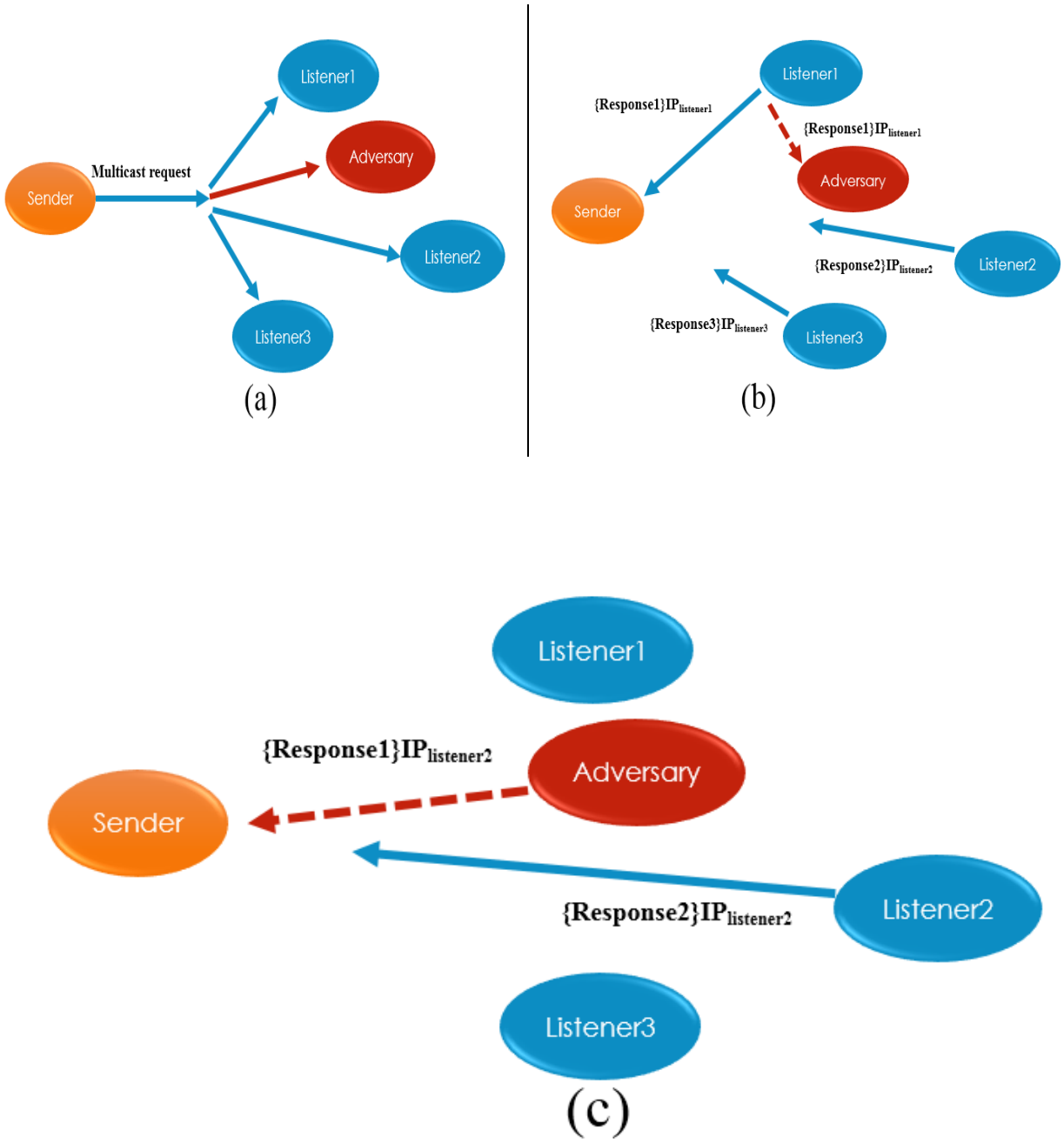
حيث تكون القيم من رقم الحقبة إلى طول البيانات رأسًا لطبقة السجل. كما ناقشنا أعلاه، يشترك المستمعون في نفس مفتاح Server MAC write key ومن المرجح أن ينتهي بهم الأمر باستخدام نفس أرقام التسلسل ورقم الحقبة. بعد ذلك، يقوم المستمعون بتشفير سلسلة من البيانات و HMAC والحشو مرة أخرى باستخدام مفتاح Server Write key متماثلة وقيمة جديدة عشوائية للـIV:

Encrypt ([Server write key, random IV], Header + Data + HMAC)

بالنظر إلى القيم المتطابقة، ما الذي سيحدث حتى لو اختلف كلاً من طول البيانات والبيانات للاستجابات من مستمعين مختلفين؟ الجواب هو أن سجل DTLS من مستمع واحد يمكن قبوله بنفس القدر كسجل DTLS صالح من مستمع آخر!

**بشكل أساسي، يعرّف المرسل المستمع بواسطة عنوان IP المصدر ورقم المنفذ لحزمة واردة.** ومع ذلك، لا تتم حماية عنوان IP أو رقم المنفذ بواسطة رأس DTLS نظرًا لأن DTLS يعمل أعلى في مكدس البروتوكول. علاوة على ذلك، **لا يتضمن رأس DTLS أي تمييز بين أعضاء نفس المجموعة،** مما يسمح لأعداد التسلسل بالتداخل. ونتيجة لذلك، لن يكون المرسل قادراً على تحديد ما إذا كان سجل DTLS قادم بالفعل من هذا المستمع الدقيق.

دعنا نفكر في سيناريو بوجود مهاجم نشط قادر على اعتراض الحزم، واستبدال البيانات غير المحمية في طبقات النقل والشبكات والبيانات، وإعادة إرسال الحزم الناتجة. نقدم مثالاً لمجموعة من عميل واحد (مرسل متعدد الإرسال) وثلاثة خوادم (مستمعين) يتواصلون بشكل آمن باستخدام النهج الموضح أعلاه لحماية الاستجابة. إذا كان بعض مستمعي المجموعة قد بدأوا العمل في نفس اللحظة، فمن المرجح أن يحتفظوا بنفس القيم لأرقام التسلسل لأن طلبات الإرسال المتعدد تصل في كل لحظة إلى جميع العقد التي تستمع إلى عنوان الإرسال المتعدد. إذا كان مستمعو المجموعة قد بدأوا التسلسل من قيم مختلفة، يمكن للمهاجم تخزين استجابات عقدة واحدة واستخدامها عندما يكون نفس رقم التسلسل على عقدة أخرى. قد يقوم المهاجم بعد ذلك باعتراض رسالة استجابة من عقدة واحدة وتغير عنوان IPثم يقوم بإعادة إرسال هذه الرسالة كإجابة من عقدة أخرى. سيتم قبول الرسالة من قبل المرسل باعتبارها صالحة بسبب الأسباب الموضحة أعلاه. هذا هو المعروف باسم هجوم إعادة الإرسال التي أجريت من خلال IP مخادع. ولكن كيف يؤدي المهاجم هذا النشاط الخبيث؟ هناك ما لا يقل عن سيناريوهين محتملين للهجوم تم توضيحها في الشكل 3.1 والشكل 3.2.

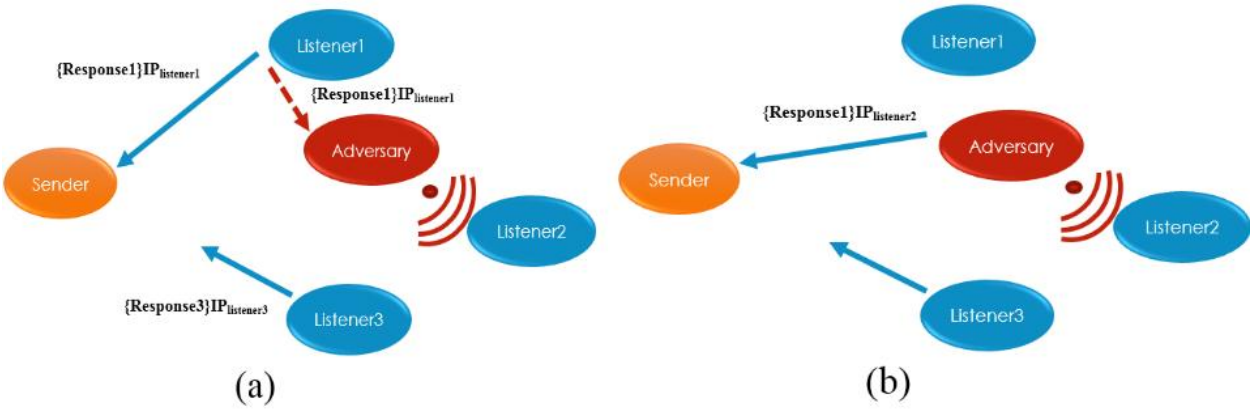



*الشكل 3- 1: إعادة الهجوم على آلية حماية الرد مع إعادة الإرسال السريع: (أ) يرسل المرسل طلب الإرسال المتعدد؛ (ب) يرسل المستمعون الردود، ويعترض الخصم رد المستمع 1؛ (ج) يعيد الخصم إرسال الرد مع عنوان IP المخادع للمستمع 2*

في السيناريو الفرعي الأول الموضح في الشكل 3.1، يرسل المرسل طلب الإرسال المتعدد إلى أعضاء المجموعة الذين يستمعون إلى عنوان متعدد الإرسال محدد. يتلقى المهاجم هذا الطلب مع أعضاء

المجموعة لأنه يستمع أيضًا على عنوان الإرسال المتعدد. وبهذه الطريقة يمكنه معرفة أن أعضاء المجموعة سيرسلون الردود قريبًا. دعونا نفترض أن المستمع 1 يجيب أولاً بسبب مسافة أقرب إلى المرسل أو تأخير استجابة أصغر في حالة وجود تأخير عشوائي في الإجابة التي يخضع لها المستمعون. يعترض المهاجم هذه الاستجابة من المستمع 1 (من السهل إيقاف الإرسال اللاسلكي) ويعيد إرسالها بسرعة مع عنوان IP للمستمع 2 الذي لم يرد بعد أو لا يزال رده على الطريق. إذا كان المهاجم قادرًا بشكل سريع على تجاوز رسالة من المستمع 2، فسيتم قبول رسالته المخادعة والمعادة ردًّا من هذا المستمع. وسيتلقى مرسل الإرسال المتعدد أيضًا الاستجابة الحقيقية ولكنه سيتجاهلها نظرًا لأنها تحتوي على رقم تسلسل أقل من الاستجابة التي تم قبولها للتو، وبالتالي فهي رسالة قديمة.

يفترض السيناريو الفرعي الثاني الموضح في الشكل 3.2 أن المهاجم لديه القدرة على إجراء التشويش على عقدة محددة، ومنع استقبال عقدة وعمل الإرسال على مستوى الراديو، أو اعتراض طريق العقدة فعليًا ومنع وظائفها، على سبيل المثال، اعتراض العقدة بمادة تحجب أو تمتص موجات الراديو. في هذه الحالة لا يحتاج المهاجم حتى أن يكون سريعًا في إعادة الرد لأنه لن يكون هناك إجابة من العقدة الشرعية على الإطلاق. إذا قام الخصم بحجب وظيفتي الاستقبال والإرسال عن المستمع 2 تمامًا، سوف يستمر هذا التأثير المدمر للهجوم حتى عندما يتم كشفه. لان المستمع 2 لم يكن على علم بطلبات الإرسال المتعدد، ولم يحاول إرسال الردود، وبالتالي لم يتم زيادة رقم التسلسل، في حين أن مرسل الإرسال المتعدد قد تلقى ردودًا صالحة مع عنوان IP لهذا المستمع وزاد من عداد رقم التسلسل. ونتيجة لذلك، عندما يبدأ المستمع في إرسال الردود بعد توقف الحظر، ستكون أرقام تسلسله قديمة، وبالتالي سيتم تجاهل الردود. وسيستمر الوضع بنفس الطريقة وتجاهل الردود الجديدة حتى يصبح مقدار الرقم التسلسلي مساوي رقم التسلسل للردود التي تم صياغتها من قبل المهاجم عندها يصبح العداد المتزايد لرقم التسلسل بالنسبة للمستمع مساويا لعداد المرسل.




*الشكل 3- 2: هجوم إعادة الإرسال على آلية حماية الاستجابة مع التشويش أو الاختراق الفيزيائي للعقد: (أ) يقوم العدو باعتراض استجابة من المستمع 1وفي نفس الوقت ينفذ التشويش على المستمع 2؛ (ب) يعيد الخصم إرسال الرد مع عنوان IP المخادع للمستمع 2*

لتقدير الضرر المحتمل الذي يمكن أن يسببه هذا الهجوم، دعنا نأخذ مثالاً على نظام أمان بسيط حيث يكون المرسل عبارة عن عقدة تحكم مركزية والمستمعين هم أجهزة استشعار الحركة مثبتة على طول جسم بعض الكائنات. يطلب المرسل بشكل دوري إذا كان هناك أي حركة بالقرب من أجهزة الاستشعار. يرد المستمعون بكلمة "نعم" أو "لا". يريد المتعدي إن يكون بالقرب من الكائن التي تحمل أحد المستشعرات. ثم يعيد إرسال الرد "بلا" بالنيابة عن مستشعر آخر على الطلبات كرد من أحدهم الذي تستقبله عقدة التحكم. ونتيجة لذلك، تعتقد عقدة التحكم أن العقدة التي تم عزلها لا تزال قيد التشغيل ولم تكن هناك أي تحركات من هذه العقدة أو المستشعرات.

### 3.1.2. إعادة استخدام Nonce في التشفير المصادق عليه

يوفر DTLS إمكانية استخدام التشفير المصادق عليه مع مجموعات البيانات المرتبطة (AEAD) مثل AES-CCM وAES-GCM التي توفر تشفيرًا موثوقًا به. أنه يوفر التشفير AES في وضع العداد مع CBC-MAC وMAC Galois على التوالي. ويأخذ كل من AES-CCM وAES-GCM قيمة nonce عشوائية كدخل من أجل عملية التشفير ولكن فقط جزء من nonce هو عشوائي لأنه يتم استخدامه كعداد في التشفير. وفقًا للقسم 3 في تحديد AES-CCM [41]، يتم إنشاء nonce ويدعى

CCMNonce لاستخدامه في AES-CCM وAES-GCM على أنه مزيج من قيمة Salt ورقم تسلسلي.

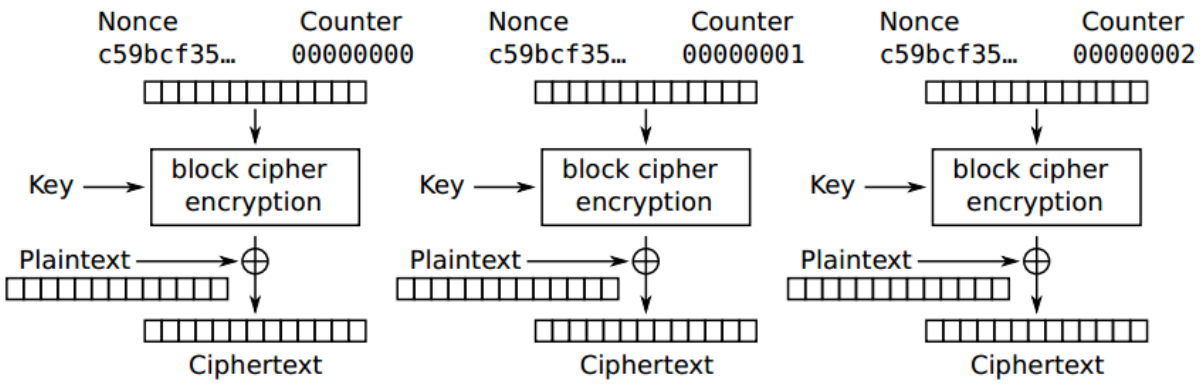
```
struct {
opaque salt[4];
opaque nonce_explicit[8];
} CCMNonce;
```

قيمة salt 4 بايت هي Client Write IV أو Server Write IV المخزنة في GSA.

يتم إنشاء 8 بايت nonce_explicit من epoch + sequence_number الذي يتم ترجمته إلى epoch + group_id + truncated_seq_number للمستمع في حالة آلية الحماية من الاستجابة المقترحة. عند تحليل CCMNonce للمستمع الناتج، يمكن ملاحظة أنه من المحتمل أن تكون متطابقة للمستمعين المختلفين في المجموعة! وذلك يعود لان المستمعين يشتركون في نفس Server Write IV من معلمات أمان المجموعة ويمكن أن يكون nonce_explicit هو نفسه بسبب وتيرة متماثلة لزيادة رقمي الحقبة والتسلسل التي تمت مناقشتها في القسم الفرعي 3.1.1 السابق.

الميزة الرئيسية للتشفير المصادق عليه هو استخدام مفتاح واحد لكل من المصادقة وتشفير البيانات. وبالتالي، يتم استخدام مفتاح Client Write IV Server Write IV فقط ولا يتم استخدام مفاتيح الـ MAC للمستمع والمرسل [41]، يفرض استخدام زوج منفصل لـ key/IV مرة واحدة فقط، وإلا قد يتم كسر أمان التشفير تمامًا.

يقوم كل من AES-CCM وAES-GCM بتشفير وفك تشفير البيانات باستخدام وضع العداد (CTR). لفهم التأثير الذي قد يكون لـ nonce reuse، دعنا نوضح كيف يعمل التشفير في وضع العداد:




*الشكل 3- 3: التشفير في وضع العداد (CTR)*

يمكن ملاحظة أنه يتم الحصول على كتلة من النص المشفر بشكل أساسي على الشكل التالي:

Ciphertext = Plaintext XOR E (Key, Counter)

هذا يعني أنه إذا تم تشفير رسالتين باستخدام نفس المفتاح والعداد، فإن XOR للنصوص المشفرة الخاصة بخوارزمية التشفير ستكون XOR من النصوص الأصلية! لذلك قد يتمكن الخصم من استعادة البيانات الأصلية تمامًا من خلال وجود نصين مشفرين فقط. لكن حتى هذه ليست النهاية. كشفت الدراسة الأمنية [53] عن AES-GCM أن إعادة الاستخدام nonce تسبب مشاكل أسوأ. ويسمح للمهاجم بحل مشكلة تهشير المفتاح الأساسي، مما يجعل التزوير بعدها تافهاً.

## 3.2. الحل المقترح

لمنع الهجمات الموضحة في القسم السابق، يجب حل مشكلة إعادة استخدام نفس الأزواج {key، nonce} من قِبل مستمعي المجموعة في نفس الوقت، مع الأخذ بعين الاعتبار جميع مزايا إعادة استخدام الموارد الرئيسية للمجموعة، نود اتباع هذه الاستراتيجية وتجنب إنشاء دورات DTLS منفصلة إضافية. كما نود أن نقترح نهجًا من شأنه إعادة استخدام الوظائف التي يوفرها بروتوكول DTLS

التقليدي الحالي وبروتوكول البث المتعدد القائم على DTLS، بحيث يتم تجنب العمليات غير الضرورية وتحقيق التكامل السلس.

لذلك، فإننا نقترح **اشتقاق مفتاح كتابة مستمع فردي لكل مستمع مجموعة بطريقة يستطيع مرسل المجموعة أيضًا اشتقاق هذه المفاتيح لكل مستمع.** والفكرة هي تطبيق HMAC للاشتقاق باستخدام جزء من موارد مفاتيح المجموعة كمفتاح سري، واستخدام عنوان الـ IP ورقم المنفذ للمستمع كنص صريح بحيث يتم تطبيق HMAC عليه أيضاً. إن عنوان IP ورقم المنفذ معروفان بشكل عام، وبالتالي يمكن الوصول إليهما بسهولة من قبل كل عضو في المجموعة. وسنشرح عملية اشتقاق المفتاح بشكل مفصل فيما يلي.

### 3.2.1. خوارزمية اشتقاق المفتاح

نلخص الخطوط العريضة لخوارزمية اشتقاق المفتاح فيما يلي:

1. قبل إرسال استجابة إلى طلب الإرسال المتعدد الأول، يقوم كل مستمع ببناء سلسلة بايتات تتكون من سلسلة تحوي على عنوان IPv6 أحادي الإرسال ورقم المنفذ المستخدم للاتصال الآمن داخل المجموعة. تحتوي السلسلة على حجم 16 + 2 = 18 بايت (144 بت).

2. ويتم إعداد سلسلة 32 بايت (256 بت)، يتم تشكيلها من موارد قفل المجموعة لكي يتم استخدامها كمفتاح سري جديد. والسلسلة هي عبارة عن (Client_write_key + Server_write_key)

3. ثم يتم استخدام HMAC بالاشتراك مع SHA-256 كدالة عشوائية زائفة (PRF) لإنشاء سلسلة سرية فردية جديدة. نظرًا لأن SHA-256 دالة هاش أساسية، يكون طول سلسلة الإخراج 256 بت (32 بايت).

4. يتم حفظ أول 16 بايت من سلسلة الخرج كمفتاح كتابة للمستمع MAC Server write_key

و16 بايت الأخير كمفتاح كتابة المستمع Server_write_key. إذا تم استخدام التشفير المصادق عليه فقط في المجموعة ولم تخزن العقد مفاتيح كتابة MAC، فسيتم حفظ آخر 16 بايت كـ Server_write_key ويتم تجاهل الـ 16 بايت الأولى.

5. عندما يتلقى مرسل المجموعة أول رسالة استجابة من مستمع، يعرّف المرسل المجموعة التي ينتمي إليها المستمع خلال حقل معرف المجموعة الموجود في الرسالة، فيستمد عنوان IP للمستضيف ورقم المنفذ من رؤوس IP وUDP ويقوم بحساب مفاتيح المستمعين الفردية التي تتبع نفس الخوارزمية. بعد اشتقاق المفتاح الفردي، يعالج المرسل الاستجابة المتلقاة.

### 3.3. تبرير الحل:

تم اختيار HMAC مع SHA-256 لاشتقاق المفتاح في خوارزميتنا لأنه خيار تستخدمه الإصدارات الحالية من TLSو DTLS لكل من توسيع المفتاح وحساب MAC لجميع أنواع التشفير في IoT. الشيء الجيد في HMAC هو أن قوته الأمنية تعتمد تماما على خصائص التشفير لتابع التهشير الأساسي والمفتاح المستخدم. ويعتبر SHA-256 قويًا جدًا خلال العقد المقبل[32] ، علاوة على ذلك، مفهوم HMAC مع تابع تهشير قوي للتشفير يعني أن معرفة النص الشائع دون معرفة المفتاح لن يعطي للخصم أي معلومات عن النص المهشر. بشرط الاختيار الصحيح للمفتاح المستخدم، فإنه سيضمن قوة الأمن لجزء التهشير من الخوارزمية المقترحة. مواصفات HMAC [31] لا تشجع بشدة على استخدام مفاتيح أقصر من طول خرج تابع التهشير الأساسي لأنه يقلل بشكل كبير من القوة الأمنية لعملية التشفير ولكن أيضا يذكر أن المفاتيح أطول من خرج تابع التهشير لا يزيد بشكل كبير من القوة الأمنية للتشفير. وبالتالي، فإن السلسلة (Client write key and the Server write key) تستخدم كمفتاح في HMAC. هذه المفاتيح يتم الحصول عليها من السر الرئيسي باستخدام PRF المذكور في [17]. هذا يعني أنها تلبي جميع متطلبات العشوائية. علاوة على ذلك، يتم استخدام سلسلة ( Client

(write key +Server write key) بدلاً من(Server MAC write key + Server write key) لأن بعض التطبيقات تدعم التشفير المصادق فقط، وخاصة في بيئة مقيدة، لحفظ الموارد وبالتالي لا يتم تخزين حتى (Server MAC write key) من أجل تجنب إجبار التطبيقات على تخزين مفاتيح MAC عند عدم الحاجة إليها، تم اختيار خيار المفتاح الأول .

يتم استخدام سلسلة عنوان IPv6 أحادي الإرسال ورقم المنفذ كسلسلة إدخال نظرًا لأنها فريدة لكل عقدة ويتم تضمينه أيضًا في رؤوس حزم المستوى الأدنى. الحقيقة الأخيرة تسمح لمدير المجموعة لتجنب توزيع أي بيانات إضافية ولا يتعين على المرسل تخزين المعلومات مسبقًا. المرسل يستقبل الرسالة الأولى، فيحسب المفتاح المقابل مرة واحدة ثم يقوم ببساطة بإضافة المستمع إلى قائمته لتتبع أرقام التسلسل.

### 3.3.1.اعتبارات أمنية

نناقش أدناه العديد من مشكلات الأمان المتعلقة بالنهج الجديد التي يجب مراعاتها.

#### 3.3.1.1. الوقاية من هجوم حجب الخدمة DoS

هناك احتمالان لأن يعرف مرسل البث المتعدد كل المستمعين الموجودين في المجموعة. أولاً، يحصل مرسل الإرسال المتعدد على معلومات حول جميع أعضاء المجموعة الموجودين من وحدة تحكم المجموعة مسبقًا حتى يكون لديه قائمة بعناوين IP وأرقام المنافذ المستخدمة للاتصال بجميع أعضاء المجموعة. ثانياً، يعتبر المرسل كل رد من عنوان غير معروف له رأس DTLS مناسب للاتصال الجماعي كأول رسالة من مستمع. في الحالة الأولى إذا حاول خصم إرسال رسالة استجابة بعنوان IP الخاص به، سيتم تجاهل الرسالة بسبب عدم تطابق العنوان ولكن حتى إذا كانت الرسالة بعنوان IP مخادع، فسيتم تشفيرها و / أو حساب MAC الخاص بالرسالة باستخدام مفتاح غير صالح وبالتالي يتم تجاهله أيضًا من قبل المرسل. في الحالة الثانية إذا كان الخصم يستخدم عنوانه الخاص، فسيتعين

على المرسل إجراء حساب HMAC قبل التحقق واكتشاف أن الرسالة غير صالحة. في الختام، لا يمكن للخصم تزوير الرسائل أو خداع أعضاء المجموعة في أي من الحالات ولكن لمنع التعرض المحتمل لهجوم DoS، فمن الأفضل توزيع المعلومات حول أعضاء المجموعة الحاليين مقدمًا.

### 3.3.2. السرية على مستوى المجموعة

على الرغم من أن النهج المقترح يفترض مفاتيح الكتابة المختلفة لكل مستمع في المجموعة، لكن لا يزال يوفر سرية المجموعة فقط. هذا يعني أن جميع أعضاء المجموعة موثوق بهم داخل المجموعة ولكن أيضًا أن أي عضو في المجموعة يمكنه استرداد أو حساب مفتاح لأي عضو آخر في المجموعة وحتى صياغة رسالة في حالة تعرض العضو للاختراق ويعرف عناوين الأعضاء الآخرين. ومع ذلك، فإن هذا الشكل من الثقة يعتبر ملائمًا لمعظم سيناريوهات إنترنت الأشياء حيث يُستخدم الاتصال لإرسال الأوامر وخاصة في الشبكات منخفضة الطاقة حيث تعتبر الكفاءة أحد اهم الاهتمامات السابقة.

### 3.3.3. تفرد معرفات المجموعة

وحدة تحكم المجموعة مسؤولة عن تعيين معرفات المجموعة. على الرغم من أنه من المهم ضمان تفرد معرفات المجموعة، إلا أنه ليس بذات الأهمية كما تفرد معرفات المرسل. من المحتمل أن يضر تكرر معرف المجموعة بالاتصال بإحدى المجموعات المجاورة لأن المرسل سيتوقف دائمًا عن البحث عندما يعثر على المجموعة الأولى في القائمة بمعرف معين ولكن المرسل ما زال قادر على التمييز بين المجموعتين عن طريق رقم المنفذ. من المفترض أن يتم إجراء الاتصالات مع مجموعات مختلفة بواسطة تطبيقات مختلفة للمرسل وبالتالي باستخدام منافذ مختلفة. الملاحظة الأكثر أهمية هي أن المعرّف المتكرر لن يفسد الأمان لأن جميع الأعضاء في المجموعتين سيكونون فرديين ومتميزين بالمفاتيح داخل الشبكة.

# التنفيذ العملي والاختبارات

في الفصل الرابع نصف الوظائف التي قمنا بتنفيذها، والسيناريوهات التجريبية وتحقيقها في تطبيقات المرسل والمستمع، وتقييم تجريبي ومناقشة نتائج التقييم. للقيام بالتنفيذ، نشير إلى للأدوات التي كانت متاحة قبل البدء، وما هي الإجراءات التي قمنا بها لإثراء البرتوكولات والخوارزميات وكيف أصبح في النهاية. كما نقوم بوصف المشاكل التي تم حلها لدعم الاتصالات اللازمة وكيفية الإعدادات التجريبية وترتيبها. أخيرًا، ونقوم أيضا بتحليل ومناقشة النتائج بالنسبة لكل من متطلبات الذاكرة، وأداء الاتصالات والطاقة تقييم الاستهلاك التي تم الحصول عليها في التجارب.

## 4. التنفيذ

قمنا بتنفيذ مكتبتنا للاتصال الجماعي الآمن والنماذج اللاحقة باستخدام نظام التشغيل Contiki OS v2.7 [22] وهو نظام تشغيل مفتوح المصدر لإنترنت الأشياء يوفر اتصالاً بالإنترنت لأجهزة لاسلكية منخفضة الطاقة. يتم تزويد نظام التشغيل Contiki ببيئة تطوير كاملة تسمى Instant Contiki. ويشمل أدوات تطوير مختلفة مثل محاكي شبكة Cooja [54] ، وتطبيق Erbium REST Engine [55]، و CoAPالذي نستخدمه كمكتبة اتصال CoAP. نقطة الانطلاق للتنفيذ هي مكتبة TinyDTLS 0.5 المتاحة للعموم [24] بواسطة Olaf Bergmann. إنه تطبيق DTLS خفيف الوزن للبيئات المقيدة التي توفر اتصالات أحادية الإرسال آمنة. يمكّن TinyDTLS العقدتين اللتين تستخدمان تطبيق DTLS من إجراء المصافحة وبعد ذلك التواصل بشكل آمن مع بعضهما البعض باستخدام معلمات الأمان المتفاوض عليها.

خوارزميات التشفير المدعومة في التنفيذ العملي هي TLS_PSK_WITH_AES_128_CCM_8 و TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 حيث يوفر خوارزميات التشفير الأولى تشفيرًا يستند إلى AES في وضع CBC ويوفر خوارزمية التشفير الثانية تشفيرًا معتمدًا على المنحنيات الإهليجية. تتطلب خوارزمية التشفير الأولى مفاتيح مشتركة مسبقًا لإجراء مصافحة واستخلاص مورد مفتاح الجلسة. يجب الاعتراف بأن تنفيذ خوارزمية التشفير الأخيرة لا تزال قيد التطوير واختيارية، وبالتالي، فإننا نركز بشكل أساسي على الأولى ولا نقوم بدعم الثانية في الملفات التنفيذية لحفظ الذاكرة. أخيرًا، يتم تضمين المكتبة في Contiki OS كتطبيق والإصدار المدروس موجود في بيئة Instant Contiki.

من أجل تنفيذ دعم اتصالات البث المتعدد الموضح في [20]، قمنا أولاً وقبل كل شيء بإنشاء بنية GSA التي تستخدمها مجموعة العقد لتخزين عنوان البث المتعدد المعين للاتصال بالمجموعة، ومعلمات الأمان المستخدمة، وعدادات الحقبة ورقم التسلسل. علاوة على ذلك، يقوم مستمعو المجموعات بتخزين قائمة بجميع مرسلي المجموعات وتتبع أرقام الحقبة والأرقام التسلسلية بشكل منفصل لكل منهم. ويقوم مرسلو المجموعات أيضًا بتخزين معرفات المرسل الخاصة بهم التي تم تعيينها لهم بواسطة وحدة تحكم المجموعة، ويقوم المستمعون بتحديد هوية المرسلين في القائمة بواسطة المعرف. ثم قمنا بتنفيذ جميع الوظائف لإنشاء بنية GSA وإضافة معلمات إليها بما في ذلك إضافة المرسلين في القائمة المقابلة. أخيرًا، قمنا بتنفيذ وظائف إضافية لإجراءات التشفير وفك التشفير التي تستخدم بيانات الأمان من GSA وتكون قادرة على معالجة النوع الجديد من رأس DTLS لحزم البث المتعدد. بعد ذلك، قمنا بتنفيذ دعم حماية الاستجابة الذي اقترحه [21]. ويتم تضمين معرف المجموعة في بنية GSA. كما يقوم مرسلو المجموعات بتخزين قائمة مستمعي المجموعات لنفس الأغراض مثل قائمة المرسلين المخزنين بواسطة المستمعين. يتم إعادة استخدام رأس حزمة المجموعة DTLS لرسائل الاستجابة مع استبدال معرف المجموعة بمعرف المرسل. إذا كان المستمع بحاجة إلى إرسال استجابة ولكن إن لم يكن هناك جلسة

آمنة أحادية الإرسال لطلب أرسال متعدد البث من المرسل، يستخدم المستمع معلمات الأمان من GSA المتعلقة بعنوان الإرسال المتعدد الذي تلقى المستمع الطلب عليه. وإذا تلقى المرسل رسالة أحادية الإرسال آمنة ولكن لا توجد جلسة DTLS أحادية الإرسال مع الرسالة الأصل، يحاول المرسل معالجة الحزمة على أساس أنه تم تشفيرها باستخدام مجموعة موارد القفل.

أخيرًا، نفذنا التحسينات على حماية الاستجابة الآلية المقترحة في هذه الأطروحة. بملف اشتقاق المفتاح الذي يتضمن وظائف لأداء وظائف التهشير المرتبط بالأجزاء المتعلقة بالمستمع من كتلة المفتاح. ويتم استدعاء وظائف من التطبيق مباشرة ومعالجة البيانات من موارد GSA. ثم يقوم مرسل DTLS بتخزين كتلة مفاتيح فردية لكل مستمع في القائمة بينما يقوم مستمعي DTLS بالكتابة فوق كتلة المفاتيح الخاصة بهم بقيم جديدة.

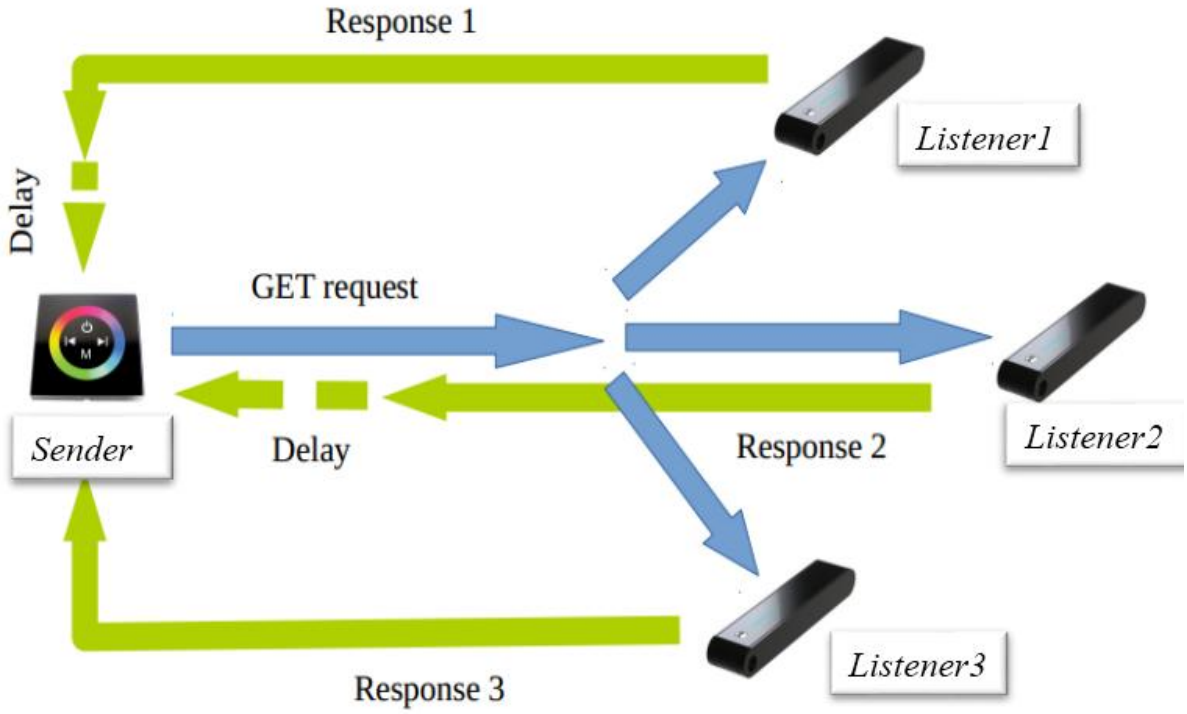
### 4.1. تكيف برتوكول COAP

نقدم تأخير استجابة مستمع عشوائي كما هو مقترح من قبل [12] لتجنب الاصطدامات بين استجابات المستمعين المختلفة. عندما يتلقى المستمع طلبًا، فإنه يختار نقطة عشوائية خلال فترة ما يسمى بوقت الراحة وينتظر انتهاء الفترة قبل إرسال الاستجابة. ويمكن حساب الحد الأدنى التقريبي لقيمة وقت الراحة باستخدام تقدير حجم المجموعة ومعدل نقل البيانات المستهدف وحجم الاستجابة المقدّر. نقوم بإثراء مكتبة CoAP على وجه التحديد في Contiki من خلال تنفيذ وقت الفراغ للاتصال الجماعي.

### 4.2. سيناريو تجريبي

في السيناريو التجريبي، تؤلف طوبولوجيا العقد المقيدة طوبولوجيا نجمية. يتم إجراء الاتصال عبر قفزة واحدة فقط لأنه لا يوجد تطبيق مستقر لبروتوكولات توجيه البث المتعدد في Contiki في الوقت الحالي، وبالتالي لا تتضمن بيئة Instant Contiki ذلك.

يكون هناك عقدة مركزية واحدة نسميها المرسل، كمرسل CoAP ومرسل DTLS ومرسل مجموعة. وتكون هناك ثلاث عقد أخرى نطلق عليها المستمعين كمستمع CoAP ومستمعين DTLS ومستمعي المجموعات. يظهر السيناريو في الشكل 4.1. يتضمن السيناريو خمسة إعدادات تجريبية للحصول على بيانات للمقارنة بين الإرسال الأحادي والإرسال الجماعي والنُهج الأمنة وغير الآمنة. يتم وصف الإعدادات بالتفصيل في القسم الفرعي التالي.




الشكل 4- 1:سيناريو تجريبي

يرسل المرسل طلب CoAP "/ hello" من نوع GET إلى المستمعين الثلاثة من خلال اتصال أحادي الإرسال أو الإرسال المتعدد. يتم تعيين قيم رمزية فريدة ذات 4 بايتات بواسطة المرسل لكل طلب مجموعة ويتم تضمينها في رأس CoAP. من المفترض أن يقوم المستمعين بمعالجة الطلب والرد برسالة

استجابة تتضمن سلسلة "Hello World!" ورمز مميز متطابق لرمز الإرسال في رسالة الطلب. يختلف إنجاز السيناريو اعتمادًا على ما إذا كان طلب GET يتم إرساله كرسالة أحادية أو كرسالة متعددة الإرسال.

في حالة البث الأحادي، يرسل المرسل طلب تأكيد Confirmable CoAP إلى مستمع، وينتظر رده وعند الحصول عليه ينتقل إلى طلب تالي. طلب تأكيد يعني أن المرسل لا يتابع طلب المستمع التالي حتى يتلقى الرد.

في حالة الإرسال المتعدد، يرسل المرسل طلب CoAP واحد متعدد البث وهو غير مؤكد للأسباب التي تمت مناقشتها سابقاً. علاوة على ذلك، بعد معالجة طلب البث المتعدد، يخضع المستمعين لتأخير عشوائي قبل إرسال الاستجابة كما هو موضح في القسم الفرعي 2.1.1، تم إعداد التأخير العشوائي ليكون في نطاق من 0 إلى 0.25 ثانية. لقد فحصنا القيم المختلفة للتأخير العشوائي من ثانية واحدة إلى ربع ثانية وتوقفنا عند القيمة الحالية، بعد أن وصلنا إلى الحالة المستقرة حيث يكون الاتصال فعالًا ولا تزال التصادمات لا تحدث.

ينتظر المرسل باستمرار الردود ويقوم بمعالجتها فور تلقيها. في كلتا الحالتين، يستخدم المرسل قيمة الرمز المميز في الحزمة المستلمة لمطابقة طلب سابق. تبديل الوقت بين الطلبات التي يقوم بها المرسل هو 10 ثوانٍ. تهدف الإعدادات إلى أن تكون الظروف أقرب ما تكون إلى تطبيقات الحياة الحقيقية، وبالتالي فإن طلبات confirmable تُستخدم في إعدادات البث الأحادي وتكون الإعدادات Non- confirmableفي إعدادات البث المتعدد. من المسلم به أن طلبات GET تُستخدم على وجه التحديد لتوفير إمكانية لتقييم آلية حماية الاستجابة.

## 4.3. إعدادات السيناريو

نقوم بتشكيل خمسة إعدادات ضمن السيناريو الموضح لتقييم أداء الاتصالات وخصائصها جميعًا. نحن نقوم بتزويد التطبيقات مسبقاً بعنوان الـ IP ورقم المنفذ في جميع الإعدادات. علاوة على ذلك، **نظرًا لعدم وجود جهاز توجيه في الإعداد الخاص بنا**، فإننا نقوم يدويًا بإضافة إدخالات في جداول التوجيه لجميع عناوين IP العقد المجاورة. إذا لم يتم ذلك الإدخال، فسيتم إسقاط الرسائل الأولى نظرًا لأن العقدة لا تعرف عنوان MAC لعقدة الوجهة.

الإعدادات الخمسة التي تم تكوينها هي كما يلي:

1) CoAP Unicast أحادي الإرسال

يقوم المرسل بالتتالي بإرسال رسالة طلب أحادي الإرسال غير آمن إلى كل مستمع وينتظر الرد. هذا يعني أن المرسل يرسل الطلب إلى المستمع الأول ولا يفعل أي شيء حتى يحصل على الاستجابة. ثم يتابع المرسل إلى المستمع التالي. ويكون المرسل والمستمعين على دراية بعناوين IP وأرقام المنافذ وعناوين MAC لبعضها البعض.

2) CoAP Multicast الأرسال المتعدد

يشترك المستمعين للاستماع على عنوان البث المتعدد معين منذ البداية. وينقل المرسل رسالة طلب واحدة مع عنوان الإرسال المتعدد كوجهة وينتظر الردود. يتم توفير عنوان الإرسال المتعدد إلى المرسل. الإدخال في جدول التوجيه للجيران غير ضروري للمرسل لأنه يتم إرسال حزم الإرسال المتعدد حاليًا في Contiki باستخدام عنوان MAC للبث. يجيب المستمعين على عنوان أحادي الإرسال للمرسل يتم توفيره في التطبيق.

3) DTLS Unicast-Unicast طلب أحادي واستجابة أحادية

في هذا الإعداد، نضيف طبقة DTLS إلى اتصالات CoAP أحادي البث لجعلها آمنة. يتم تزويد المرسل والمستمعين بمفاتيح مشتركة مسبقًا لاستخدامها في عملية المصافحة. بعد التشغيل والتشكيل، يقوم المرسل لاحقًا بمصافحة منفصلة مع كل مستمع. من أجل تجنب خلط رسائل المصافحة من جميع المستمعين، فقد حددنا فترة زمنية قدرها 3 ثوانٍ للمصافحة باستخدام عقدة واحدة. وبالتالي، يقوم المرسل بالمصافحة مع المستمع الأول؛ إذا استغرق الأمر أقل من 3 ثوانٍ، فينتظر انتهاء الفترة الزمنية ثم ينتقل إلى مصافحة المستمع التالي.

عند إنشاء جلسات آمنة لـ DTLS مع جميع المستمعين، يبدأ المرسل في إرسال طلبات الإرسال الأحادي الآمنة بشكل فردي باستخدام معلمات مشتركة مع كل مستمع كما هو في إعداد CoAP أحادي البث.

يحمي المستمعين أيضًا الاستجابات باستخدام معلمات الأمان التي تتم مشاركتها بشكل فردي مع المرسل وترسلها إلى عنوان البث الأحادي للمرسل.

4) DTLS Multicast-Unicast طلب متعدد البث واستجابة أحادية مع إجراء عملية مصافحة

في هذا الإعداد، لا يزال المرسل يبدأ بمصافحة كل مستمع، لكنه بعد ذلك يرسل رسالة طلب الإرسال المتعدد الآمن DTLS بدلاً من ثلاث رسائل أحادية الإرسال. لحماية طلب البث المتعدد، يستخدم المرسل المعلمات من مجموعة أمان المجموعة (GSA) والتي تم تحميلها مسبقًا على المرسل والمستمعين. يتم تسليم GSA لأعضاء المجموعة بواسطة وحدة تحكم مجموعة وهناك بروتوكولات منفصلة لذلك. ومع ذلك، فإن إدارة مفتاح المجموعة خارج نطاق هذه الأطروحة، وبالتالي فإننا نقوم مسبقًا بتكوين GSA في العقد. يستخدم المستمعين جلسات DTLS آمنة أحادية الإرسال، والتي تم إنشاؤها من خلال المصافحة، لحماية استجاباتها.

5) طلب متعدد البث واستجابة أحادية بدون إجراء عملية مصافحة DTLS Multicast-Unicast

لا يزال المستمعين يرسلون ردودًا على عنوان البث الأحادي للمرسل، ومع استخدم معلمات الأمان من GSA لحماية الاستجابات. هذا يعني أن العقد لا يلزمها إجراء عمليات المصافحة على الإطلاق في هذا الإعداد. لأن GSA كافية لتأمين الرسائل في كلا الاتجاهين. وكما في الإعداد 4، يتم تحميل GSA مسبقًا على المرسل والمستمعين. ولكن لم تعد هناك حاجة إلى المفاتيح المشتركة مسبقًا للمصافحة، لذا يتم استبعادها في عملية التشفير للتطبيقات.

وبدلاً من ذلك، يتم تنفيذ خوارزمية اشتقاق المفتاح الموضحة في القسم الفرعي 3.2.1 على جانبي المرسل والمستمع. ويقوم المستمعين بحساب المفاتيح الفردية مباشرة بعد بدء التطبيقات. يحسب المرسل مفتاح فردي للمستمع عندما يتلقى المرسل الاستجابة الأولى من هذا المستمع. وبالتالي، يمكن اعتبار اشتقاق المفتاح بالنسبة للمرسل كجزء من معالجة الاستجابة.

## 4.4. تقيم النتائج:

لتقييم النهج الذي نتبعه في التواصل الجماعي الآمن ومقارنته مع الآخرين، نقيس الخصائص المهمة لقابلية الاستخدام وظروف البيئة المقيدة. بداية، يتم قياس أشغال الذاكرة لتقييم الحمل في الذاكرة. ثانياً، يتم قياس الوقت اللازم لأداء التبادل والحساب والمصافحة. أخيرًا، يعتبر قياس استهلاك الطاقة لعمليات مختلفة أمرًا مهمًا للأجهزة المقيدة. تتم جميع التجارب على نفس النوع من العقد في المحاكي، وبالتالي يمكن اعتبار وقت انتشار الإشارة ثابت.

### 4.4.1. آثار البرتوكول على الذاكرة

نقوم بقياس آثار البرتوكول على ذاكرة القراءة فقط (ROM) وذاكرة الوصول العشوائي (RAM) لجميع إعداداتنا. يتم تعريف استهلاك ذاكرة الوصول العشوائي (RAM) من خلال المتغيرات التي تم تهيئتها

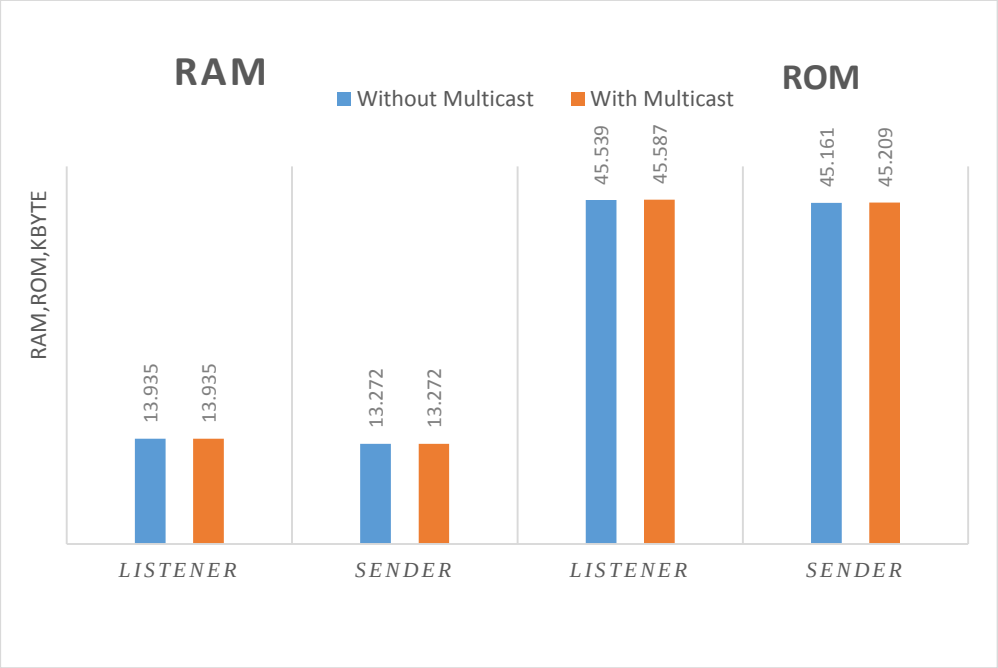
مسبقًا، بينما استهلاك ROM هو في الأساس حجم البرنامج الذي تم تحميله في العقدة. وتتضمن النتائج التي تم الحصول عليها باستخدام بيئة Contiki إلى استهلاك الذاكرة للبرامج التي تم تحميلها على عقد المرسل والمستمع والتي تتضمن مكدس الاتصال بالكامل. ومع الأخذ بعين الاعتبار، أنه تم إيقاف تشغيل بروتوكول التوجيه وبرتوكول TCP ولا يتم تضمينهما في صورة برنامج الذي تم تنفيذه للعقد نظرًا لعدم استخدامها في السيناريو الخاص بنا.

قبل المتابعة إلى الإعدادات، يجب مناقشة تفاصيل إضافية. كما ذكرنا في القسم الفرعي 4.1، قمنا بتنفيذ تكييف CoAP للاتصال المتعدد البث. من الواضح، أنه يساهم في استهلاك الذاكرة ويمكن أن يؤثر على قياساتنا لذلك نقوم باستخدام اتصال مع وجود البث المتعدد ومن ثم يمكن إيقاف تشغيل البث المتعدد للـ CoAP.

لتقدير المساهمة التي يضيفها البث المتعدد على استهلاك الذاكرة، نقيس استهلاك الذاكرة من خلال التغيرات التي أجريناها لتكيف الاتصال المتعدد للبرامج مع CoAP وبدون التغيرات ويتم ذلك في بيئة Contiki. وترد النتائج في الجدول 4.1 والشكل 4.2.

| | ROM, bytes | | RAM, bytes | |
|---|---|---|---|---|
| | Sender | Listener | Sender | Listener |
| Without multicast | 45.161 | 45.539 | 13.272 | 13.935 |
| With multicast | 45.209 | 45.587 | 13.272 | 13.935 |

*الجدول 4-1: شغّل الذاكرة دون ومع تكييف COAP للاتصال المتعدد.*



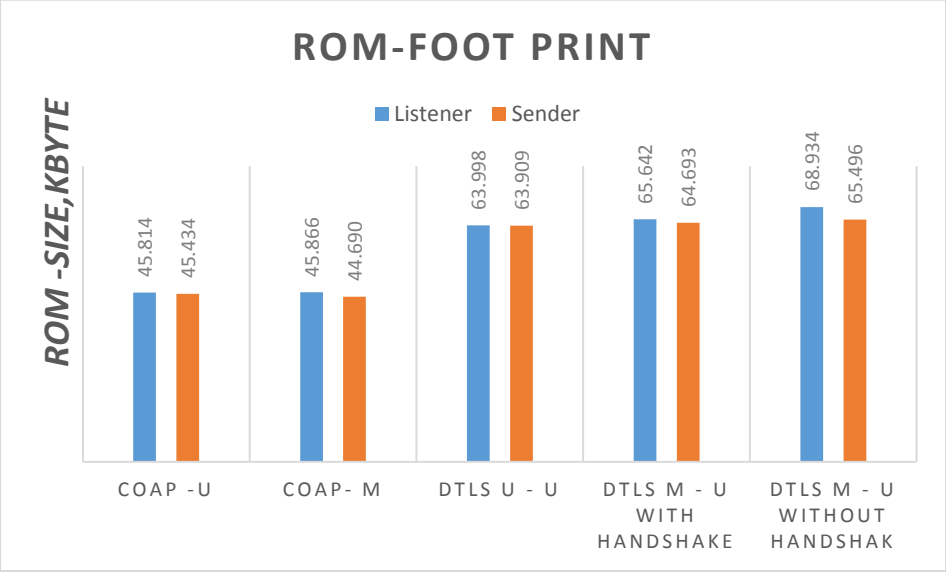

*الشكل 4- 2: رسم توضيحي يبين تغيرات استهلاك الذاكرة مع وجود البث المتعدد وبدونه.*

تشير النتائج التي تم الحصول عليها إلى استهلاك الذاكرة كما يمكن أن يرى، يضيف التكيف فقط 48 بايت في ROM إلى كل من تطبيقات المرسل والمستمع. يمكن اعتبار هذه الإضافات العامة غير ذات أهمية كبيرة، لذا فإننا نحافظ على التهيئة المضمنة لجميع الإعدادات لتحقيق التجانس.

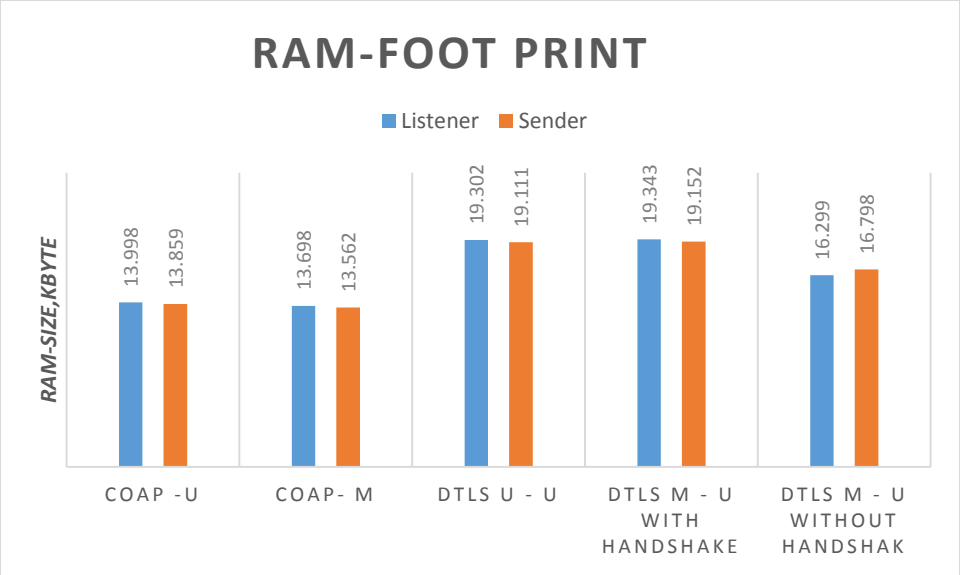
يتم عرض نتائج قياس استهلاك الذاكرة المتبعة لإعداداتنا التجريبية في الجدول 4.2. تظهر أيضًا البصمات المدمجة ROM وRAM كمخططات بيانية في الشكل 4.3 والشكل 4.4 على التوالي لأغراض توضيحية.

| Setting | ROM, bytes | | RAM, bytes | |
|---|---|---|---|---|
| | Sender | Listener | Sender | Listener |
| CoAP Unicast | 45.434 | 45.814 | 13.859 | 13.998 |
| CoAP Multicast | 44.690 | 45.866 | 13.562 | 13.698 |
| DTLS Unicast - Unicast | 63.909 | 63.998 | 19.111 | 19.302 |
| DTLS Multicast – Unicast with handshake | 64.693 | 65.642 | 19.152 | 19.343 |
| DTLS Multicast – Unicast without handshake | 65.496 | 68.934 | 16.798 | 16.299 |

الجدول 4- 2: إشغال الذاكرة في الإعدادات التجريبية




الشكل 4-3: بصمة ROM في الإعدادات التجريبية




*الشكل 4-4: بصمة ذاكرة الوصول العشوائي في الإعدادات التجريبية*

الاتجاه التصاعدي لحجم ROM قابل للتفسير لأننا نضيف تدريجياً وظيفة أمنية إلى السيناريو من إعداد إلى إعداد، وهذا يؤدي بدوره إلى زيادة حجم التعليمات اللازمة لتحقيق هذه الوظائف الأمنية. يرجع الانخفاض في استهلاك ذاكرة ROM و RAMلإعداد CoAP Multicast مقارنة بإعداد CoAP Unicast بشكل أساسي إلى حقيقة أن المرسل في الإرسال المتعدد ليس لديه إدخالات تحتوي على عناوين أحادية الإرسال وعناوين MAC للمستمعين المجاورين في جدول التوجيه ليوصل الرسالة إلى عقد المستمع بينما يكون هناك حاجة إلى هذه الإدخالات في حالة الاتصال أحادي البث.

علاوة على ذلك، يتطلب تطبيق المستمع في الغالب ROM وذاكرة RAM أكبر من تطبيق المرسل لأنه يشتمل على وظائف لمعالجة طلب REST في التطبيق نفسه بينما لا يحتاج المرسل إلى ذلك.

ربما كانت النتيجة الأكثر أهمية هي أن النهج المقترح من قبلنا أنه يتطلب 2.35 كيلو بايت و3 كيلوبايت من ذاكرة الوصول العشوائي RAM لتطبيقات المرسل والمستمع على التوالي للاتصال بالمجموعة الكاملة أقل من النهج المقترح في مسودة IETF من قبل فريق عمل DICE [20]. في نفس

الوقت، يكون متطلبات ROM أعلى بنحو 1 كيلوبايت من نتائج فريق عملDICE لكل من المرسل والمستمع.

بشكل عام، تعد متطلبات ذاكرة الوصول العشوائي (RAM) الأكثر أهمية بالنسبة للأجهزة المقيدة لأنها تحتوي على سعة حجم أقل بكثير من ROM. هناك عدة أسباب لذلك، مثل ذاكرة الوصول العشوائي (RAM) تتطلب إمدادًا ثابتًا بالطاقة وهو أغلى من ROM عادةً من حيث مفهوم الكلفة لكل كيلوبايت.

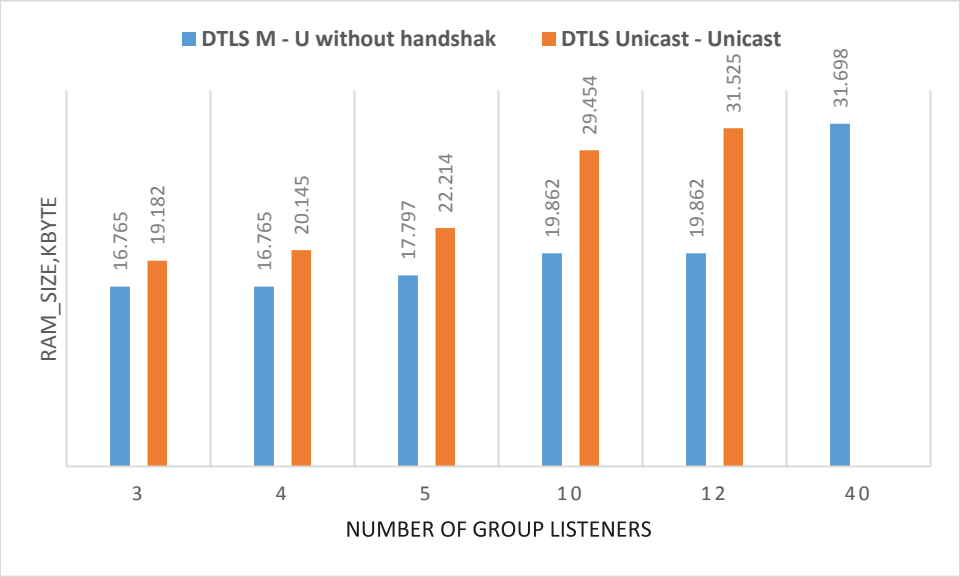
السبب الرئيسي للاختلاف في متطلبات الذاكرة بين الإعدادات الآمنة هو عدد سياقات التشفير وسياقات التهشير التي يمكن استخدامها بالتوازي، وبالتالي، يجب تخصيص ذاكرة لها. حيث يحتاج المرسل إلى سياق كتابة واحد لإرسال رسائل البث المتعدد وثلاثة سياقات قراءة لمعالجة الردود.

أخيرًا، نود أن ندرس قابلية التوسع في مقاربتنا ومقارنته مع DTLS التقليدية، أي لنرى كيف تتغير متطلبات الذاكرة الوصول العشوائي RAM وفقًا لعدد المستمعين في المجموعة. نحن نفكر في كيفية تأثير ذلك على شغل ذاكرة الوصول العشوائي (RAM) فقط على جانب المرسل لأن حجم المجموعة لا يحتوي على تأثير على المستمعين وليس من المفترض أن تعرف عقدة الاستماع بالمستمعين الأخرين الموجودين في المجموعة. يوضح الجدول 4.3 والشكل 4.4 متطلبات الذاكرة بناءً على عدد من المستمعين.

وما يحد اختباراتنا بالنسبة لعدد المستخدمين هو أن نظام Contiki OS يقوم بتخصيص 32 كيلو بايت من ذاكرة الوصول العشوائي إلى نفس العقد المتوفرة فيه، وزيادة عدد المستمعين يمكن أن تفوق الذاكرة المتوفرة وبالتالي يعطي أخطاء برمجية.

| Group_size | Sender | |
|---|---|---|
| | RAM, bytes | |
| | DTLS Unicast - Unicast | DTLS Multicast – Unicast without handshake |
| 3 | 19.182 | 16.765 |
| 4 | 20.145 | 16.765 |
| 5 | 22.214 | 17.797 |
| 10 | 29.454 | 19.862 |
| 12 | 31.525 | 19.862 |
| 40 | | 31.698 |

الجدول 4-3: إشغال RAM بالنسبة للمرسل مع زيادة عدد المستمعين.




الشكل 4-5: إشغال RAM بالنسبة للمرسل في الإعدادين DTLS M-U without handshake و DTLS Unicast-Unicast

من الواضح أن نهج DTLS التقليدي يصل إلى الحد الأقصى لمقدار الذاكرة بالفعل مع 12 مستمعًا، بينما يتطلب منهجنا قدرًا مماثلًا من الذاكرة لـ 40 مستمعًا. لذلك، يمكننا أن ندعي بأمان أن نهجنا أكثر قابلية للتوسع من النهج التقليدي، وعمومًا، يمكن استخدامه في الممارسة العملية. تجدر الإشارة إلى أن النهج الذي تتبعه مجموعة عمل DICE يتطلب ذاكرة أكثر من DTLS التقليدية لأن طلبات البث المتعدد تحتاج إلى تخصيص ذاكرة للسياقات الأمنية بالإضافة إلى بنية معطيات الرسائل.

أيضًا، لا يقوم نظام Contiki OS بتخصيص الذاكرة تدريجيًا بل بالكتل الكبيرة. وبالتالي، قد لا يصنع مستمعان إضافيان أي اختلاف في بعض الحالات.

### 4.4.2. أداء الاتصالات

لتقييم قابلية الاستخدام والكفاءة، نقيس الوقت اللازم لإجراء معاملة مجموعة واحدة. نعتبر معاملة المجموعة مصطلحًا نعرفه على أنه حصول المرسل على المعلومات بطلب واحد من جميع المستمعين. وهذا يلزم تنفيذ الإجراءات المختلفة بواسطة المرسل والمستمعين، بناءً على الإعداد الذي يتم تنفيذه من أجل تحقيق تبادل المعلومات المطلوب. فيما يلي نوضح مما تتكون المعاملة لكل إعداد:

1. في إعداد CoAP Unicast، تبدأ المعاملة عندما يبدأ المرسل في إعداد رسالة طلب CoAP لإرسالها إلى المستمع الأول وينتهي عندما يختتم المرسل بمعالجة استجابة من المستمع الأخير. في هذا الإعداد، يرسل المرسل الطلب إلى المستمع التالي فقط عندما يتلقى ويعالج استجابة عن طلب سابق.
2. في CoAP Multicast، تبدأ المعاملة عندما يبدأ المرسل في إعداد طلب البث المتعدد CoAP وينتهي عندما يختتم المرسل بمعالجة الاستجابة الثالثة على مستوى CoAP. لاحظ أنه نظرًا لأن المستمعين يخضعون لتأخير -عشوائي قبل الرد، فسيكون ترتيب الردود الواردة أيضًا تعسفيًا

بحيث يمكن أن تصل الاستجابة الثالثة من أي من المستمعين. أيضًا، تصبح مدة المعاملة قيمة عشوائية في بعض النطاقات. ومع ذلك، ينبغي أن تصل إلى بعض القيم المتوسطة المستقرة.

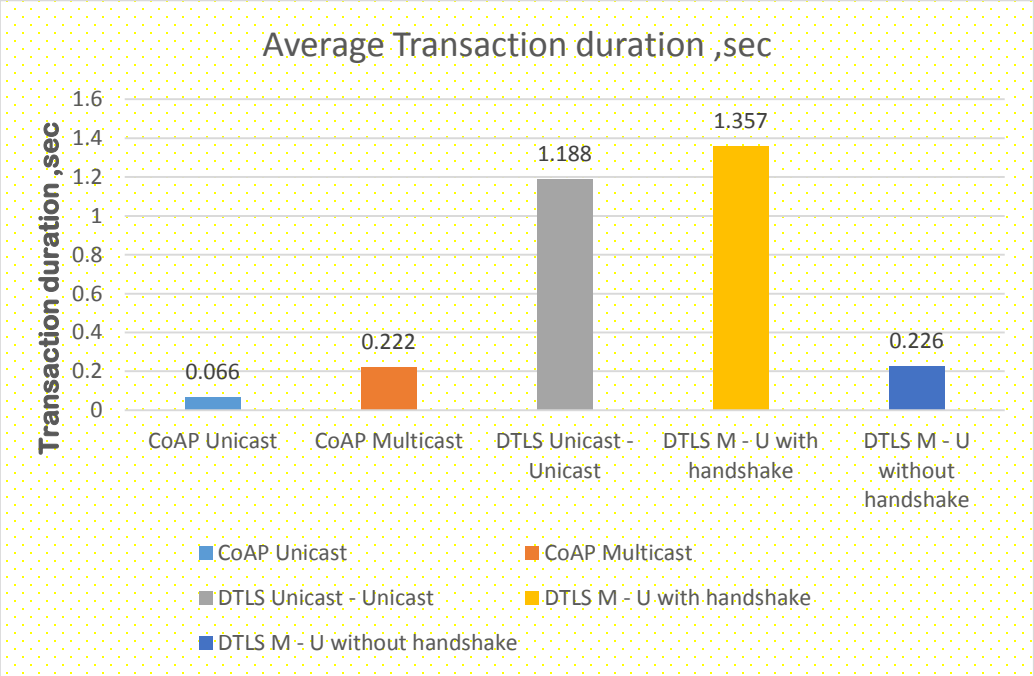
3. في DTLS Unicast-Unicast، تبدأ المعاملة عندما يبدأ المرسل عملية مصافحة مع المستمع الأول ثم يتابع مع المستمعين الآخرين. بعد الانتهاء من المصافحة باستخدام المستمع الأخير، يتابع المرسل بشكل مشابه لإعداد CoAP Unicast، مع اختلاف فقط في أن يتم تمرير طلب CoAP إلى طبقة DTLS ليتم حمايته قبل طبقة النقل ويتم فك تشفير الاستجابات قبل الانتقال إلى مستوى CoAP.
4. فيDTLS Multicast- Unicast with handshake، تبدأ المعاملات أيضًا بإجراء المصافحة ولكن تتبع نهج الإرسال المتعدد من إعداد CoAP Multicast مع إضافة طبقة DTLS لمعالجة الرسائل. تتم المعاملة عند تلقي آخر رد.
5. في DTLS Multicast- Unicast without handshake، لا يتم إجراء أي عملية مصافحة. وتتضمن المعاملة إرسال طلب البث المتعدد، وتنفيذ حساب هاش المفتاح عند تلقي كل استجابة والمعالجة الفعلية للردود المستلمة.

كما هو الحال في إعداد CoAP Multicast تبدأ المعاملة عندما يبدأ المرسل في إعداد طلب البث المتعدد CoAP والذي ينتقل بعد ذلك إلى طبقة DTLS. وتستمر العملية مع تلقي المرسل للرد الأول، وإنشاء إدخال لمعلومات المستمع الأول وحساب Write key-Server Write (Server MAC key) بناءً على عنوان الـ IP أحادي الإرسال ورقم منفذ مستمع المجموعة الجديد هذا، ومعالجة الاستجابة الفعلية. يتم تكرار نفس العمليات للاستجابات الثانية والثالثة. وتنتهي المعاملة عند معالجة الاستجابة الثالثة الأخيرة.

لاحظ أنه في التشغيل الفعلي، لن يتم إجراء المصافحات واشتقاق المفتاح الفردي إلا مرة واحدة لفترة محددة من فترة إنشاء المفتاح، بحيث يتضمن الجزء الأول من المعاملة عليهما فقط بينما تشمل

المعاملات اللاحقة فقط جزء الرد على الطلب. ومع ذلك، لا تزال هناك حاجة إلى إجراء عمليات المصافحة واشتقاق المفتاح بعد كل عملية إعادة صياغة للمفتاح يمكن أن تكون متكررة جدًا مع وجود عدد كبير من أعضاء المجموعة، خاصةً إذا كانوا ينضمون ويغادرون بشكل دوري. وبالتالي، نحن نعتبر إصدارًا كاملاً من المعاملات لإعطاء صورة لأسوأ سيناريو من حيث أداء الاتصالات واستهلاك الطاقة.

لإنجاز التقييم، نقوم بإجراء 20 تجربة لكل إعداد، ثم نقوم بحساب متوسط القيم لكل أعداد. يعرض الشكل 4.5 والجدول 4.3 النتائج الإجمالية لقياس زمن المعاملة.




*الشكل 4-6: مدة المعاملة في السيناريو الذي تم فحصه لإعدادات البث الأحادي غير الآمن والبث المتعدد غير الآمن والبث الأحادي الآمن و الإرسال المتعدد الآمن جزئيًا وإعدادات اتصال البث المتعدد الآمن*

| | Average Transaction duration ,sec |
|---|---|
| **CoAP Unicast** | **0.066** |
| **CoAP Multicast** | **0.222** |
| **DTLS Unicast - Unicast** | **1.188** |
| **DTLS Multicast – Unicast with handshake** | **1.357** |
| **DTLS Multicast – Unicast without handshake** | **0.226** |

*الجدول 4-4: قيم مدة المعاملة في السيناريو الذي تم فحصه لخمسة إعدادات*

يمكن أن نرى من الجدول أن هناك أكثر من ثلاثة أضعاف الفرق في المدة بين إعدادات CoAP Unicast و CoAP Multicast. ويرجع ذلك إلى التأخير العشوائي الذي يتراوح من 0 إلى 0.25 ثانية للردود على طلبات البث المتعدد التي قررنا الحفاظ عليها متجانسة لجميع إعدادات البث المتعدد.

من حيث المبدأ، يمكن تنفيذ المعاملة من الناحية النظرية بشكل أسرع بالنسبة لإعداد CoAP Multicast مقارنتاً بسيناريو البث الأحادي الذي يحتاج المرسل فيه إلى إرسال رسالة واحدة فقط من خلال ضبط ودراسة التأخير العشوائي لعقد المستمع بشكل أوسع. **ويمكن اعتبار تطوير خوارزمية تسمح للمستمعين باختيار لحظة لإرسال رد وتجنب الاصطدامات بمثابة عمل مستقبلي.** وهو عمل ليس بالسهل لأن المسافة بين المرسل وكل مستمع مختلفة، وبالتالي، زمن انتشار الإشارة أيضا مختلف، في حين أن الخوارزمية بشكل عام يجب أن تحل المشكلة من أجل توزع عشوائي للعقد وربما تغيير ديناميكي للتوزع.

تتضمن الإعدادات 3 و4 عمليات المصافحة وهذا هو السبب الرئيسي لإظهار الوقت الأطول بشكل ملحوظ. سينمو هذا البطء أكثر فأكثر مع زيادة عدد أعضاء المجموعة. الأهم من ذلك، هو أن نهجنا أسرع بست مرات من نهج الإرسال المتعدد مع وجود عملية مصافحة، وأسرع بأكثر من خمسة أضعاف

منهج البث الأحادي التقليدي، وحتى أنه قابل للمقارنة مع CoAP Multicast غير الآمنة، مع الأخذ في الاعتبار التأخير العشوائي لعقد المستمع. نظرًا لأن نهج الإرسال المتعدد من قبل مجموعة عمل DICE الممثلة في الإعداد رقم 4 يتضمن المصافحة والتأخير العشوائي، فهو أقل من حيث الأداء الحسابي حتى من النهج التقليدي للـ DTLS مع جلسات أحادية البث منفصلة.

نحن أيضًا نقيس بشكل فردي الوقت الذي تستغرقه كل عملية مصافحة بالإضافة إلى جزء الطلب الفعلي الذين يكونان معًا إجمالي مدة المعاملة. ويتم عرض نتائج الإعدادات 3 و4 و5 في الجدول 4.4.

| | **Handshake1, sec** | **Handshake2, sec** | **Handshake3, sec** | **Request, sec** |
|---|---|---|---|---|
| **DTLS Unicast - Unicast** | **0.558** | **0.267** | **0.283** | **0.081** |
| **DTLS Multicast - Unicast with** | **0.566** | **0.283** | **0.299** | **0.210** |
| **DTLS Multicast - Unicast without** | | | | **0.226** |

*الجدول 4- 5 :التوقيت للمراحل الفردية من المعاملات للإعدادات مع اتصال آمن*

هناك العديد من الملاحظات القيمة التي يمكن استنتاجها من هذه النتائج. أولاً، تستغرق عملية المصافحة مع المستمع الأول وقت أكثر بمرتين من الوقت اللازم لمصافحة عقد المستمع الأخرى. ربما يرجع ذلك إلى حقيقة أن على المرسل تهيئة وظائف التهشير والتشفير عند الاستدعاء الأول بطبقة DTLS. ملاحظة مهمة أخرى هي أنه يتطلب التبادل الفعلي لرسائل الطلب والاستجابة في إعداد DTLS Unicast - Unicastإلى 0.015 ثانية فقط أكثر من مدة معاملة المجموعة بأكملها في إعداد CoAP Unicast، أي 0.066 ثانية. يشير هذا إلى أن الاتصال الآمن ينتج عنه تكاليف اتصال معقولة تقريباً

بنسبة 25٪، وستنخفض النسبة فقط مع نمو المسافة بين العقد. أخيرًا، يتطلب تبادل رسائل الطلب والاستجابة قدرًا مماثلًا من الوقت لكل من إعدادات

DTLS Multicast-Unicast without handshake و DTLS Multicast-Unicast with handshake هذا يعني أن عملية التهشير الرئيسية واشتقاق المفاتيح الفردية في منهجنا لا تؤدي إلى تأثير كبير على المعاملة من ناحية المدة الزمنية.

بالإشارة إلى الإعدادات الآمنة DTLS Unicast -Unicastو DTLSMulticast-Unicast with handshake، ستكون هناك حاجة بالنسبة لعقدة مستمع جديدة لإجراء مصافحة DTLS بواسطة جميع عقد المرسل في المجموعة. والأسوأ من ذلك، أن عقدة المرسل الجديدة ستقوم بمصافحة DTLS مع جميع عقد الاستماع الموجودة في المجموعة، والتي من المفترض أن تكون كبيرة. وفي كلتا الحالتين، قد يقلل هذا بشكل كبير من الاستجابة العامة للشبكة وتوافرها، ويفرض اعتماد آليات إضافية لإبقاء عقد المرسل على علم بعقد المستمع الجديدة. بينما لا يتطلب منهجنا المتبع في إعداد DTLS Multicast-Unicast without handshake إجراء أي مصافحة DTLS وبالتالي، ليس له أي تأثير خاص على الأداء العام للشبكة وتوافرها، ويسمح لعقد المرسل بالبقاء غير معروف بالنسبة لعقد المستمع الموجودة في المجموعة.

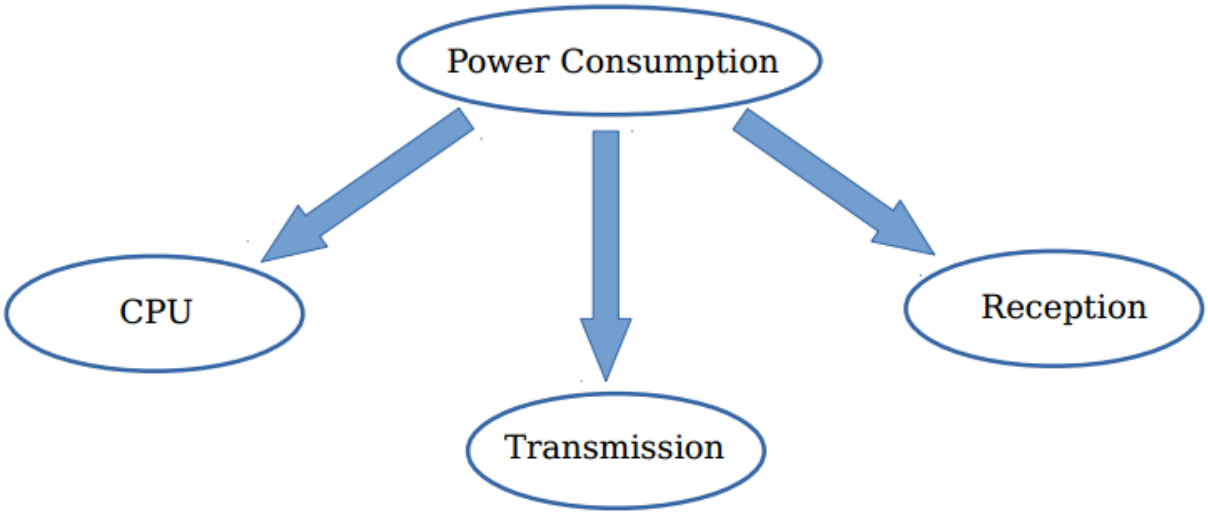
### 4.4.3. استهلاك الطاقة

أن الطاقة التي تستهلكها العقدة المقيدة خلال فترة زمنية محددة تكون طاقة مستهلكة بإحدى الأشكال التالية، الطاقة المستهلكة من قبل وحدة المعالجة المركزية $E_{CPU}$، والطاقة المستهلكة في نقل البيانات $E_{TX}$، والطاقة المستهلكة في استقبال البيانات $E_{RX}$، والطاقة المستهلكة في الوسط الراديوي أثناء الاستماع الفعلي، وأخيراً، الطاقة المستهلكة أثناء كون العقدة في وضع الطاقة المنخفضة. في بحثنا، لا تستخدم أجهزة الاتصال أي دورة **عمل راديوية** وهي ألية من الآليات لتقليل استهلاك طاقة العقدة عن

طريق التبديل بين استخدام الراديو وعدم استخدامه[56]. على الرغم من المزايا الواضحة لتوفير الطاقة، فإن استخدام هذه الآليات يؤدي إلى زيادة ملحوظة في فقد الحزمة التي يمكن أن تخلق مشاكل، على سبيل المثال، أثناء إجراء مصافحة DTLS.

**نحن نعتبر أن موضوع بروتوكولات أمان متكاملة وفعالة مع دورة العمل الراديوية موضوع مرتبط بتحسين أداء الشبكة وبالتالي فهو خارج نطاق هذا العمل ونتركه بمثابة عمل مستقبلي.** عدم وجود دورة العمل الراديوية يعني أن الأجهزة لا تتحول أبدًا إلى وضع الطاقة المنخفضة، وتستمع دائمًا إلى الوسط الفيزيائي وتعمل وحدة المعالجة المركزية بشكل مستمر.

لقياس استهلاك الطاقة للإعدادات في السيناريوهات التي تم إعدادها، قمنا بحساب استهلاك الطاقة من خلال ضرب استهلاك الطاقة للعقدة مضروب بالمدة الزمنية التي كانت العقدة فيها نشطة، والذي تم جمعه عن طريق استخدام Energest framework المقدم من Contiki [57]، حيث توفر وحدة Energest تقديرًا خفيفًا للطاقة قائم على برامج أجهزة إنترنت الأشياء مقيدة الموارد. نستخدم نفس بنية المعاملة كما في القسم الفرعي 4.4.2. ويشمل ذلك إعداد طلب CoAP أو معالجة استجابات CoAP أو بالإضافة إلى ذلك تشفيرها وفك تشفيرها في حالة الاتصال الآمن. نحن أيضًا نقيس الطاقة المستهلكة لإرسال واستقبال الرسائل، لكننا نتجاهل الطاقة التي تستهلكها العقد للاستماع المستمر للوسط الفيزيائي. يوضح الشكل 4.7 مخطط مكونات الاستهلاك التي يتم قياسها.




*الشكل 4- 7: يعبر عن مكونات استهلاك الطاقة في التجارب التي أجريت على جهاز مقيد.*

يوضح الجدولين 4.5 و4.6 مساهمات الطاقة المختلفة الإجمالية التفصيلية. نشير إلى عمليات اشتقاق الموارد الأساسية باعتبارها مجموعة من الإجراءات بما في ذلك استرداد المفاتيح واشتقاق المفتاح ومصافحة DTLS. نشير أيضًا إلى تبادل رسائل CoAP كمجموعة من الإجراءات المتعلقة برسائل CoAP المرسلة / المستقبلة، بما في ذلك معالجة الأمان الخاصة بها.

| setting | Energy Consumption, mJ | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sender | | | | | | Listener | | | | | |
| | Key material operations | | | CoAP Message exchange | | | Key material operations | | | CoAP Message exchange | | |
| | $E_{CPU}$ | $E_{TX}$ | $E_{RX}$ | $E_{CPU}$ | $E_{TX}$ | $E_{RX}$ | $E_{CPU}$ | $E_{TX}$ | $E_{RX}$ | $E_{CPU}$ | $E_{TX}$ | $E_{RX}$ |
| **CoAP Unicast** | | | | **0.65** | **0.67** | **0.40** | | | | **0.10** | **0.15** | **0.19** |
| **CoAP Multicast** | | | | **0.46** | **0.21** | **0.40** | | | | **0.10** | **0.15** | **0.18** |
| **DTLS Unicast - Unicast** | **2.09** | **3.91** | **3.14** | **0.77** | **0.61** | **0.56** | **0.78** | **1.26** | **1.08** | **0.13** | **0.22** | **0.17** |
| **DTLS Multicast –Unicast with handshake** | **2.09** | **3.91** | **3.14** | **0.54** | **0.20** | **0.56** | **0.78** | **1.26** | **1.08** | **0.14** | **0.22** | **0.16** |
| **DTLS Multicast – Unicast without handshake** | **0.13** | | | **0.54** | **0.20** | **0.56** | **0.04** | | | **0.14** | **0.22** | **0.16** |

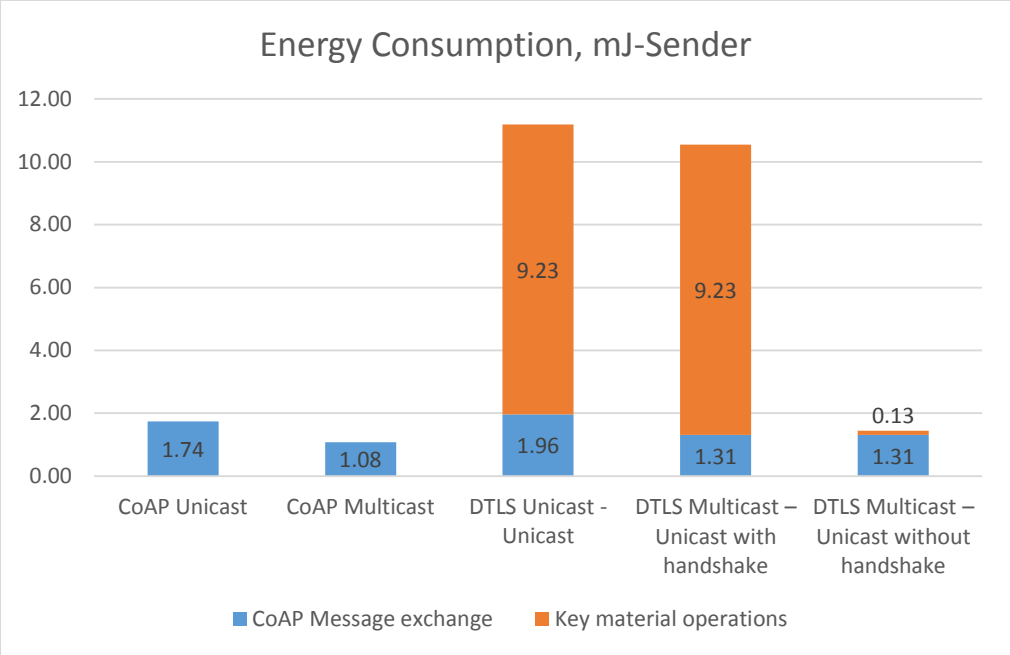
*الجدول 4- 6: استهلاك الطاقة المفصل على عقدة المرسل وعقد المستمع.*

يقدم الجدول 4.6 الاستهلاك الإجمالي للطاقة لكل إعداد من الإعدادات التي قمنا بها.

| Total Energy Consumption, mJ | | | | |
|---|---|---|---|---|
| | Sender | | Listener | |
| | CoAP Message exchange | Key material operations | CoAP Message exchange | Key material operations |
| CoAP Unicast | 1.74 | | 0.44 | |
| CoAP Multicast | 1.08 | | 0.43 | |
| DTLS Unicast - Unicast | 1.96 | 9.23 | 0.53 | 3.15 |
| DTLS Multicast – Unicast with | 1.31 | 9.23 | 0.53 | 3.15 |
| DTLS Multicast – Unicast | 1.31 | 0.13 | 0.53 | 0.04 |

*الجدول 4- 7: استهلاك الطاقة الإجمالي على عقدة المرسل وعقد المستمع.*

يوضح الشكل 4.8 والشكل 4.9 إجمالي استهلاك الطاقة في كل معاملة جماعية على عقدة المرسل وعلى إحدى عقد المستمع، على التوالي. لم نلاحظ أي اختلاف ذي صلة بين عقد المستمع المختلفة.

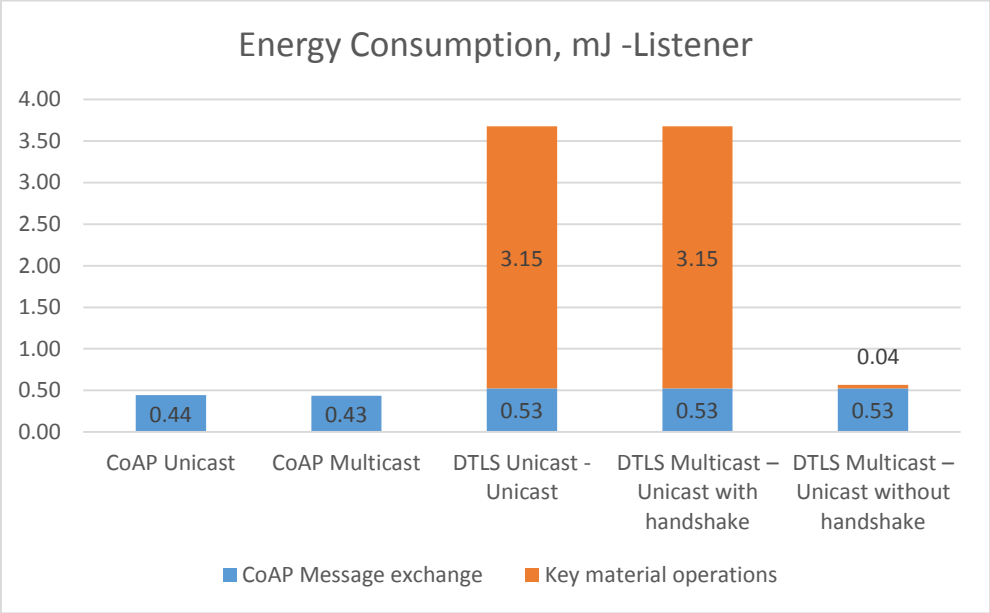



*الشكل 4- 8: الطاقة المستهلكة بواسطة عقدة المرسل أثناء معاملة مجموعة أولية.*

بالنظر إلى الشكل 4.8 والطريقتين غير الآمنتين، نلاحظ أن إعداد CoAP Unicast يعرض استهلاك طاقة أعلى بنسبة 40٪ تقريبًا من إعداد CoAP Multicast هذا بسبب إرسال ثلاث رسائل مختلفة أحادية الإرسال، بدلاً من رسالة متعددة الإرسال واحدة. وبالطبع، سيزداد هذا الاختلاف مع زيادة عدد عقد الاستماع. كما يؤدي الإعدادان الآمنان DTLS Unicast - Unicast و DTLS Multicast – Unicast with handshake إلى استهلاك طاقة أعلى بكثير. ويرجع ذلك بشكل رئيسي إلى القيام بعملية مصافحة DTLS مع عقد المستمع الثلاثة، بالإضافة إلى المعالجة الإضافية وإرسال أو استقبال رسائل CoAP كسجلات DTLS.

في هذه الحالة أيضًا، يعرض إعداد DTLS Multicast – Unicast with handshakeاستهلاكًا للطاقة أقل قليلاً من إعدادDTLS Unicast - Unicast، بسبب إرسال رسالة طلب الإرسال المتعدد المفردة، بدلاً من ثلاث رسائل طلب أحادي الإرسال. كما نلاحظ أيضا، لا يتطلب منهجنا الذي اتبعناه من المرسل في إعدادDTLS Multicast – Unicast with handshake إجراء أي مصافحة مع عقد

الاستماع. وبالتالي، يقتصر تأثير الأمن على إرسال / استقبال رسائل CoAP الموسعة كسجلات DTLS، وتنفيذ عمليات الموارد الأساسية خفيفة الوزن. ويترتب على ذلك استهلاكًا للطاقة يقل بمقدار 7 أضعاف عن أحد النهجين الآمنين الآخرين، ويمكن مقارنته مع الطريقة التي يتم عرضها بواسطة نهج البث المتعدد غير الآمنCoAP Multicast الآن، نأخذ عملية استهلاك الطاقة على جانب عقدة المستمع .




*الشكل 4- 9: الطاقة المستهلكة بواسطة عقدة المستمع أثناء معاملة مجموعة أولية.*

كما هو مبين في الشكل 4.9، فإن الطريقتين غير الآمنتين CoAP Multicast و CoAP Unicastتعرضان نفس استهلاك الطاقة. يتوافق هذا مع حقيقة أن كل عقدة مستمع ترد ببساطة برسالة استجابة، بغض النظر عن الاستقبال السابق لرسالة أحادية الإرسال أو رسالة إرسال متعدد. للسبب نفسه، تعرض الطرق الآمنة الثلاثة نفس استهلاك الطاقة في تبادل رسائل CoAP الفعلي.

أيضًا، يعرض الإعدادان الآمنان DTLS Unicast - Unicast و DTLS Multicast – Unicast with handshake نفس استهلاك الطاقة لعمليات الموارد الرئيسية، بسبب أداء مصافحة DTLS مع عقدة المرسل.

ومع ذلك، لا يتطلب منهجنا المعتمد في إعداد DTLS Mul-Mul إجراء أي مصافحة باستخدام عقدة المرسل. وبالتالي، يقتصر تأثير الأمن على نقل / استقبال رسالة استجابة CoAP الموسعة كسجلات DTLS، وتنفيذ عمليات خفيفة الوزن لتوليد الموارد الأساسية.

ويترتب على ذلك أن النهج الذي نتبعه يعرض استهلاكًا للطاقة يقل بمقدار 6 أضعاف عن كل من الطريقتين الآمنتين الأخريين، ويمكن مقارنته مع النهجين غير الآمنين CoAP Unicast و CoAP Multicast. كما ناقشنا أيضًا في القسم 4.4.2، تشير النتائج في الشكلين 4.8 و4.9 إلى المعاملة الجماعية الأولى(التهيئة)، التي تنفذها عقدة المرسل مع عقد المستمع الثلاث. بمعنى أنه، ابتداءً من معاملة المجموعة الثانية، لا تحتاج عقدة المرسل نفسها إلى إجراء أي مصافحة أخرى بنفس عقد المستمع. ونتيجة لذلك، فإن إعدادات البث المتعدد الآمنة و DTLS Multicast–Unicast without handshake وDTLS Multicast–Unicast with handshake ستعرضان أداءً متماثلاً في ذلك الحين. ومع ذلك، يمكن ملاحظة فوائد منهجنا على الأداء في كل مرة تنضم فيها عقد جديدة إلى مجموعة البث المتعدد وتبدأ في تبادل الرسائل الآمنة. هذا يجعل منهجنا ملائم بشكل خاص لتبنيه في وجود مجموعات البث المتعدد على نطاق واسع وعدد العقد الكبير، لا سيما إذا كانت ذات عضوية ديناميكية بالنسبة لانضمام العقد وخروجها من المجموعة. علاوة على ذلك، يمكن ضغط حزمة البث المتعدد بشكل أكثر كفاءة على طبقة 6LoWPAN كعمل مستقبلي.

من الملاحظ يؤدي الأمان إلى زيادة استهلاك الطاقة بشكل كبير في الإعدادات 3 و4 على كل من المرسل والمستمعين ويرجع ذلك أساسًا إلى عملية مصافحة DTLS شديدة الاستهلاك للطاقة. حيث

تستهلك عملية المصافحة بمفردها حوالي3.05 mJ من جانب المرسل و3.12 mJ من جانب المستمع. ونظرًا لأنه يتعين على المرسل إجراء ثلاثة مصافحة في السيناريو الخاص بنا، فإن الاستهلاك يتضاعف ثلاث مرات ليصل إلى حوالي9.14 mJ ويمكن أن يصل إلى قيم حرجة في حالة تزايد عدد عقد المجموعة، على سبيل المثال، 100 عضو في المجموعة وهو رقم معقول وفقًا لوثائق اتصال مجموعة CoAP [16]. في نفس الوقت، يستهلك الاشتقاق الرئيسي للموارد في الإعداد 5 بناءً على مقاربتنا 0.04 ميغا جول فقط لكل مستمع! نتيجة لذلك، نستنتج انه يستهلك المرسل في إعداد DTLS Multicast–Unicast without handshake الآمن طاقة أقل من CoAP Unicast غير الآمن! وهذا يعني أنه من خلال مقاربتنا للأمان، قد لا يتطلب إدخال أي موارد إضافية من جانب المرسل في حالة ما إذا كان النظام يتحول من اتصال أحادي الإرسال إلى اتصال متعدد البث.

أخيرًا، توضح النتائج أن حسابات الأمان على حدث طلب واحد (فك تشفير طلب وتشفير استجابة عند عقدة المستمع وتشفير طلب وفك تشفير استجابة عند المرسل) تسبب استهلاك طاقة إضافي طفيف جداً وبالتالي لا تضيف أي أعباء على الفعالية بالاتصال.

# الخاتمة والأعمال المستقبلية

## 5.1. الخلاصة

تهدف هذه الأطروحة إلى إنشاء إطار للاتصال الجماعي الآمن الذي يمكن استخدامه بشكل واسع في تطبيقات الإنترنيت للشبكات ذات الموارد المقيدة، لا سيما في عالم إنترنت الأشياء المتنامي بسرعة. وتم اختيار بروتوكولDTLS للاتصال الآمن من طرف إلى طرف، وتطبيقه الخفيف المضمن في نظام التشغيلContiki OS ومقترحات تكييف البروتوكول لاتصالات المجموعة كأساس لتطويرنا.

أولاً، قمنا بتطبيق دعم اتصالات البث المتعدد الآمنة التي سمحت بوجود مستمعين متعددين بالإضافة إلى مرسلين متعددين في مجموعة واحدة كما وفرنا حماية للرد. ثم بررنا بعد ذلك لماذا وجدنا طريقة القيام بجلسات أحادية منفصلة آمنة لرسائل استجابة المستمع على طلبات الإرسال المتعدد غير فعالة. وتم تحليل النهج الحالي لحماية الاستجابة وتبيان نقاط الضعف والمشاكل الأمنية فيها. حيث كان هذا النهج عرضة لهجوم إعادة الإرسال بسبب إمكانية تشكل رؤوس متطابقة من السجلات DTLS لمستمعي المجموعة المختلفين، كما لم يحمي المستمعين في المجموعات من إعادة استخدام أزواج {key، nonce} متطابقة مما قد يؤدي إلى الكشف الكامل عن النص المشفر في حالة التشفير المصادق عليه. ولقد اقترحنا آلية لتحسين حماية الاستجابة التي تم تحليلها، وبالتالي التخفيف من الهجمات التي تم اكتشافها. علاوة على ذلك، تم اقتراح تدابير حول كيفية جعل النهج محمي ضد هجمات حجب الخدمة في الشبكات المقيدة في إنترنت الأشياء، وتمت مناقشة الاعتبارات الأمنية الأخرى المحتملة. كما تم تنفيذ النهج المحسن وتحليله.

تم إنشاء إعداد تجريبي، مكون من عقدة واحدة كعقدة مرسل متعدد البث، وثلاث عقد كمستمعين جماعيين.

تضمن سيناريو الاتصال الذي قمنا بتنفيذه على إرسال طلب الإرسال المتعدد من عقدة المرسل وإرسال استجابات البث الأحادي من عقد المستمع. لتقييم النهج الذي اقترحناه، قمنا بفحص خمسة إعدادات مختلفة للسيناريو:

اتصال أحادي الإرسال غير آمن، اتصال متعدد الإرسال غير آمن، اتصال آمن أحادي الإرسال، اتصال آمن متعدد الإرسال مع مصافحة، واتصال آمن متعدد الإرسال بدون مصافحة. تم فحص الإعدادات من حيث شغل الذاكرة، وأداء الاتصالات واستهلاك الطاقة. أظهرت دراستنا متطلبات ROM أعلى قليلاً ولكن متطلبات RAM أقل بشكل ملحوظ مقارنة بنهج DTLS التقليدي ونهج DTLS البث المتعدد المقدم في النهج الحالية. كما وجدنا أن نهجنا أيضا قابل للتوسع والتعميم. وكان أداء الاتصالات في مقاربتنا أفضل من الطريقتين الآمنتين الحاليتين إلى حد كبير وكان قابلًا للمقارنة مع إعداد البث المتعدد غير الآمن.

## 5.2. الخاتمة

لقد قدمنا، نهجنا القائم على DTLS لتأمين اتصالات البث المتعدد في مجموعات من أجهزة إنترنت الأشياء مقيدة الموارد. توفر مقاربتنا تكييفًا لطبقة سجل DTLS، وتستغل الموارد الأساسية الشائعة المشتركة بين أعضاء المجموعة لحماية رسائل البث المتعدد ورسائل الرد أحادية الإرسال ذات الصلة بكفاءة، ولا تتطلب إجراء أي مصافحة DTLS. لقد قمنا بتطبيق منهجنا على نظام التشغيل Contiki، وأظهرت نتائجنا أن مقاربتنا خفيفة للغاية على أنظمة إنترنت الأشياء مقيدة الموارد وتتفوق على الطرق المقترحة سابقًا، من حيث شغل الذاكرة أداء الاتصالات واستهلاك الطاقة.

## 5.3. الأعمال المستقبلية

هناك العديد من المواضيع التي يمكن أن تتكامل مع عملنا لتعطي إطار عمل متكامل وسنذكر بعض المواضيع التي لم نخض فيها في عملنا وتركناها كعمل مستقبلي.

**في البداية**، يمكن العمل على بروتوكولات إدارة مفتاح المجموعة وتحليلها، وتنفيذ البروتوكول الأكثر ملاءمة ودمجه في إطار عملنا كعمل مستقبلي. وهذا من شأنه أن يلغي الحاجة في التوزيع المباشر للموارد الرئيسية بين أعضاء المجموعة من قبل وحدة تحكم المجموعة.

**ثانياً**، يمكن تصميم خوارزمية لاكتشاف وتطبيق وقت التأخير الأمثل قبل إرسال استجابات البث الأحادي بواسطة مستمعي المجموعات. قد يؤدي هذا إلى زيادة أداء الاتصالات الجماعية عن طريق تقليل وقت انتظار الاستجابة.

**أخيرا**، معظم أجهزة إنترنت الأشياء تستخدم حاليًا في مجموعة كبيرة من الخدمات والتطبيقات (مثل الصحة الإلكترونية والزراعة الذكية والشبكات الذكية والأتمتة المنزلية). ومع ذلك، لا يزال الافتقار إلى قابلية التوسع وقلة موثوقية الاتصال إمكانية تعرضه للتهديدات الأمنية المختلفة ويمثل هذان العائقان تحديات كبيرة في الاعتماد الأوسع نطاقًا على بروتوكول التوجيه لشبكات الطاقة المنخفضة والخسار (LLNs)، ويعتبر العمل على تصميم بروتوكول توجيه موحد للشبكات ذات القدرة المنخفضة والخسارة (LLNs)بمثابة عمل مستقبلي لتوسيع عملنا والتكامل معه للحصول على برتوكول توجيه أمن لإنترنيت الأشياء.

# الملاحق

الملحق أ

# شرح مفصل لعملية المصافحة في برتوكول DTLS

يستخدم بروتوكول المصافحة(Handshake) في مرحلة المصافحة ويتكون من عدة رسائل يتم تبادلها كما هو موضح في الشكل 1. يعرض الشكل جميع رسائل المصافحة والترتيب الذي يتم إرسالها به. تعتمد الرسائل المميزة بعلامة النجمة على الحالة، ويتم إرسال بعض خوارزميات التشفير، ولكن ليس كلها.

سيشبه تدفق رسالة مصافحة نموذجية تدفق الرسالة في الشكل 1(a) ابتداءً من رسالة ClientHello الثانية. يتم إرسال رسالة HelloRequest بعد تفاوض جلسة العمل الأولية إذا كان جانب الملقم يريد إعادة التفاوض على معلمات الاتصال. يتم إرسال HelloVerifyRequest بواسطة العميل للتأكد من أن العميل شرعي، أي أن رسالة ClientHello تم إرسالها من عنوان IP أصلي. تحتوي الرسالة على ملف تعريف ارتباط عديم الحالة (الذي يمكن للخادم التحقق منه دون تذكره)، والذي يجب على العميل تضمينه في رسالة ClientHello الثانية.

تحتوي رسالة ClientHello على صفيف من وحدات البايت العشوائية، تُستخدم لاحقًا في مرحلة إنشاء المفتاح، وقائمة بخوارزميات التشفير المناسبة وأساليب الضغط المدعومة. قد تحتوي أيضًا على معرف جلسة، يشير إلى أن العميل يريد تحديث جلسة العمل الحالية أو إعادة فتح جلسة سابقة. إذا تم إرسالها

بعد تلقي رسالةHelloVerifyRequest، كما تحتوي الرسالة ClientHello على ملف تعريف الارتباط من تلك الرسالة. وأخيراً وليس آخراً، قد تحتوي رسالة ClientHello على قائمة بالامتدادات، المستخدمة للتفاوض على تفاصيل إضافية للجلسة.

يتم إرسال رسالة ServerHello عند تلقي رسالة ClientHello تحتوي على مجموعة من البايتات العشوائية، تُستخدم لاحقًا في مرحلة إنشاء المفتاح، ونوع التشفير وطريقة الضغط، تكون محددة في خانات رسالة ClientHello المستلمة. إذا كانت رسالة ClientHelloتحتوي على امتدادات إضافية، والتي يدعمها الخادم ويرغب في استخدامها، فإن رسالة ServerHello سوف تحتوي على قائمة بتلك الرسائل أيضًا.

يتم تبادل رسائل الشهادات إذا كان نظام المصادقة المحدد يتطلبها. ويرسل العميل رسالة شهادة إذا تلقيت رسالة CertificateRequest من الملقم. كما يحتاج أيضًا إلى إرسال رسالةCertificateVerify، تحتوي على بيانات مشفرة خاصة بالمفتاح، في أخر الرسالة لإثبات امتلاك المفتاح الخاص المطابق للمفتاح العام الذي تم توفيره مع شهادة العميل.

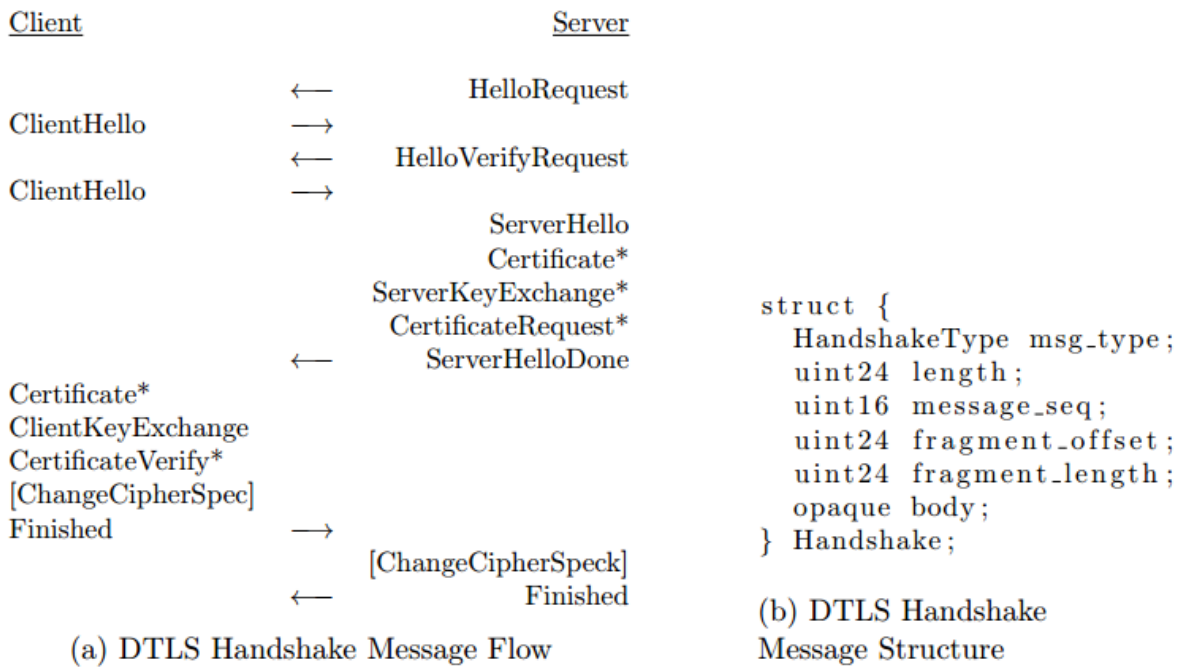
يتم إرسال ServerKeyExchange عندما يتطلب النظام عملية تبادل المفاتيح ومعلمات إضافية. على سبيل المثال، يتم إرسال هذه الرسالة إذا تم استخدام طريقة تبادل مفتاح Diffie Hellman ولا يتم إصلاح المعلمات المستخدمة من قِبل النظراء. ويرسل الملقم CertificateRequest إذا كان يريد العميل بمصادقة نفسه. يتم إرسال رسالة ServerHelloDone للإشارة إلى أن الملقم لن يرسل المزيد من

الرسائل عند هذه النقطة. وبناءً على مخطط تبديل المفاتيح المحدد، تحتوي رسالة ClientKeyExchange على سر سابق أو بيانات تم استخدامها لاستخلاص سر رئيسي سابق (على سبيل المثال، معلمات خوارزمية Diffie Hellman أو معرفات PSK. يستخدم سر ما قبل الرئيسي في مرحلة اشتقاق المفتاح.

رسائل ChangeCipherSpec ليست رسائل مصافحة. ومع ذلك، يتم استخدامها داخل المصافحة لإشراك الخوارزميات والمفاتيح في عملية التفاوض لاستخدامها في الرسائل النهائية. تحتوي الرسالة النهائية على بيانات مشتقة من تهشير كل الرسائل (المرسلة والمستلمة) حتى الآن (باستثناء التكرارات، HelloVerifyRequest وجميع الرسائل المرسلة والمستلمة قبلها) بالترتيب الذي تم إرسالها / استلامها. وهي مشفرة وموثقة بالمفاتيح والخوارزميات المختارة، وهي بمثابة اختبار ناجح لعملية المصافحة. كما ذكرنا سابقًا، تم تصميم DTLS للاستخدام مع قنوات نقل غير موثوقة. هذا هو السبب في أنDTLS، في معظم الأحيان، لا يتطلب أن يتم استلام جميع السجلات أو استلامها في التسلسل الصحيح. ومع ذلك، فإن البروتوكول يتضمن آليات، والتي تمكن من إعادة الإرسال حيثما دعت الحاجة، عملية المصافحة هي واحدة مثل هذه الحالة، لأنه يتطلب تبادل موثوق للرسائل لتحقيق النجاح.

يعالج DTLSإعادة إرسال الرسائل من خلال المصافحة بالتصميم، ويتضمن بشكل فعال الاتصالات الموثوقة عند إنشاء اتصال وعند إعادة التفاوض على معلمات الجلسة.

يظهر الشكل الدقيق لرسالة المصافحة في الشكل 1 (b). تمكّن هذه البنية الكشف عن فقدان الرسائل بالإضافة إلى إعادة تجميع رسائل المصافحة المجزأة. ويتم استخدام الحقل seq لحساب عداد رسائل العقد من خلال عملية المصافحة وتمكين الطرف الآخر في الاتصال من الكشف عن فقد رسالة مصافحة سابقة. تُستخدم حقول الطول، ونهاية التجزئة، وطول الأجزاء لتلائم الأجزاء الواردًة داخل مخزن مؤقت حتى يتم إعادة بناء رسالة مصافحة كاملة وجاهزة للمعالجة.




*الشكل 1:عملية المصافحة في برتوكول الـDTLS*

نتيجة لعملية المصافحة الناجحة، يتم إنشاء حالة اتصال للجلسة التي تم التفاوض عليها بين نظيري الاتصال. وتحتوي حالة الاتصال المذكورة على معلومات، مثل دور النظير في الجلسة المعينة، والخوارزميات المستخدمة للتشفير، وتوليدMAC ، واشتقاق المفتاح، بالإضافة إلى الموارد المستخدمة في توليد المفاتيح. ويوضح الشكل 2 الوصف المفاهيمي لهذه المعلمات.

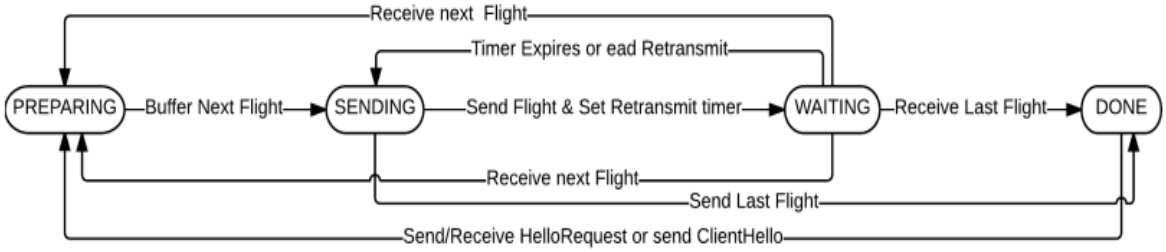
```
struct {
  ConnectionEnd              entity;
  PRFAlgorithm               prf_algorithm;
  BulkCipherAlgorithm        bulk_cipher_algorithm;
  CipherType                 cipher_type;
  uint8                      enc_key_length;
  uint8                      block_length;
  uint8                      fixed_iv_length;
  uint8                      record_iv_length;
  MACAlgorithm               mac_algorithm;
  uint8                      mac_length;
  uint8                      mac_key_length;
  CompressionMethod          compression_algorithm;
  opaque                     master_secret[48];
  opaque                     client_random[32];
  opaque                     server_random[32];
} SecurityParameters;
```

*الشكل 2: الوصف المفاهيمي لمعلمات الجلسة*

**مخطط الحالة لـ DTLS**

يستخدم DTLS مخطط الحالة المعروضة في الشكل 3 لتوجيه إرسال الرسائل وإعادة الإرسال والاستقبال. يملي مخطط الحالة إرسال الرسائل في دفقات، حيث تكون الدفقة عبارة عن سلسلة من الرسائل المرسلة قبل إرسالها يجب أن يتم توقع استجابة ما. بمجرد أن يتم إنشاء دفقة يتم نقلها وتنتظر استجابة. إذا لم تصل الاستجابة خلال إطار زمني معين، فسيتم إعادة إرسال آخر دفقة. كما يعيد النظير إرسال آخر دفقة له إذا لم يتلقى جميع الرسائل المتوقعة من الدفقة الحالية للطرف المتصل الآخر. وبالتالي، إذا استلم أحد الزملاء إعادة إرسال، فإنه يعيد إرسال آخر دفقة له للسماح للزميل المتعاقب باستقبال أي رسائل فائتة. تنتهي آخر دفقة من الزملاء بنقل رسالة منتهية. بمجرد إرسال الدفقة الأخيرة،

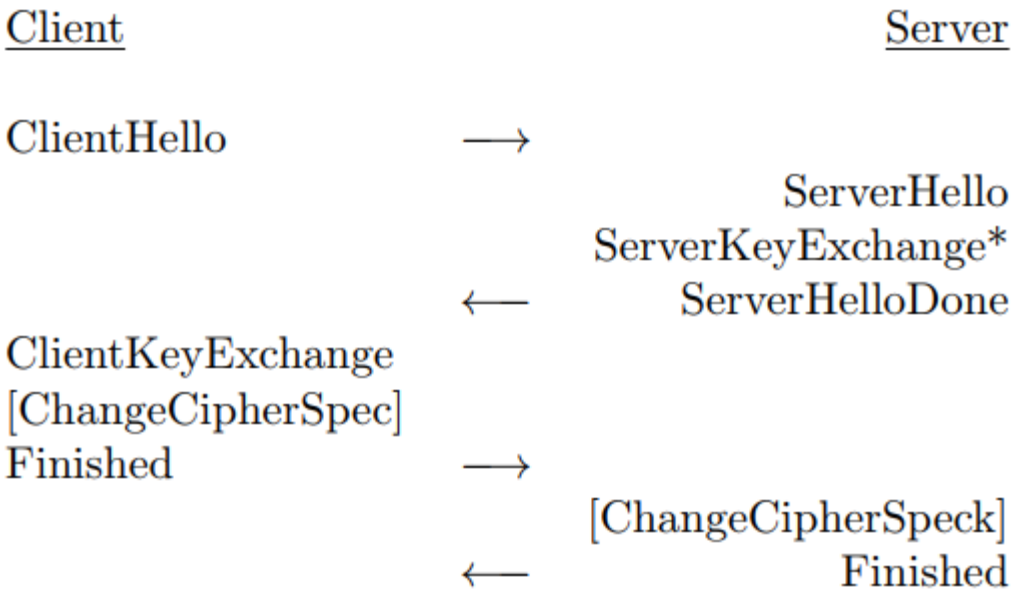
بالنسبة إلى جانب الملقم، أو استلام آخر دفقة نظير، يتم إدخال الحالة التي تم تنفيذها من جانب العميل. وإرسال وتلقي رسائل HelloRequest أو ClientHello إعادة تشغيل عملية تبادل الرسائل.




*الشكل 3: مخطط الحالة لبرتوكول الـDTLS*

**تشارك المفاتيح المسبق**

عندما يتم استخدام المفاتيح المشتركة مسبقًا، يستمر تدفق رسائل مصافحة DTLS كما هو موضح في الشكل 4 وتعتمد الرسائل المميزة بعلامة النجمة على الحالة. يتم استخدام العميل والخادم رسالة hello للموافقة على استخدام مجموعة تشفير TLS PSK WITH AES 128 CCM 8. إذا كان الخادم يريد استخدام سر معين مسبقًا، فإنه يرسل تلميحًا للهوية في رسالة ServerKeyExchange الخاصة به. إذا تلقى الخادم رسالةServerKeyExchange، فإنه يتحقق مما إذا كان يحتوي على السر المشترك المسبق. إذا كان كذلك، فإنه يتضمن هوية psk لهذا السر في رسالة ClientKeyExchange الخاصة به. إذا تم تلقي أي رسالةServerKeyExchange، يحدد العميل سر مسبق وترسل هويته إلى الخادم في رسالةClientKeyExchange وتخلص المصافحة إلى أن كلا الجانبين يرسلان بصمة مشفرة للتغيير وانتهاء الرسالة.




الشكل 4:تشارك المفاتيح المسبق

هناك نوعان من الآثار الهامة لاستخدام مفاتيح مشتركة مسبقا لعملية المصافحة. أولاً، تكون عملية تبادل الإشارات ذات عدد رسائل أقل. لا يتم تبادل أي رسائل شهادات، مما يعني أيضًا نقل البيانات بشكل كبير لأن الشهادات يمكن أن تكون طويلة جدًا. ثانياً، لا يتطلب الأمر أي عمليات للتوقيع والتحقق وتشفير المفتاح العام، مما يسهل أيضاً المصافحة من وجهة نظر حسابية. يتم استخدام السر المشترك مسبقًا لتشكيل سر ما قبل الرئيسي، والذي يستخدم بدوره في توليد السر الرئيسي وفي النهاية مفاتيح تشفير الجلسة. يظهر البناء لسر ما قبل الرئيسي في الشكل 5.

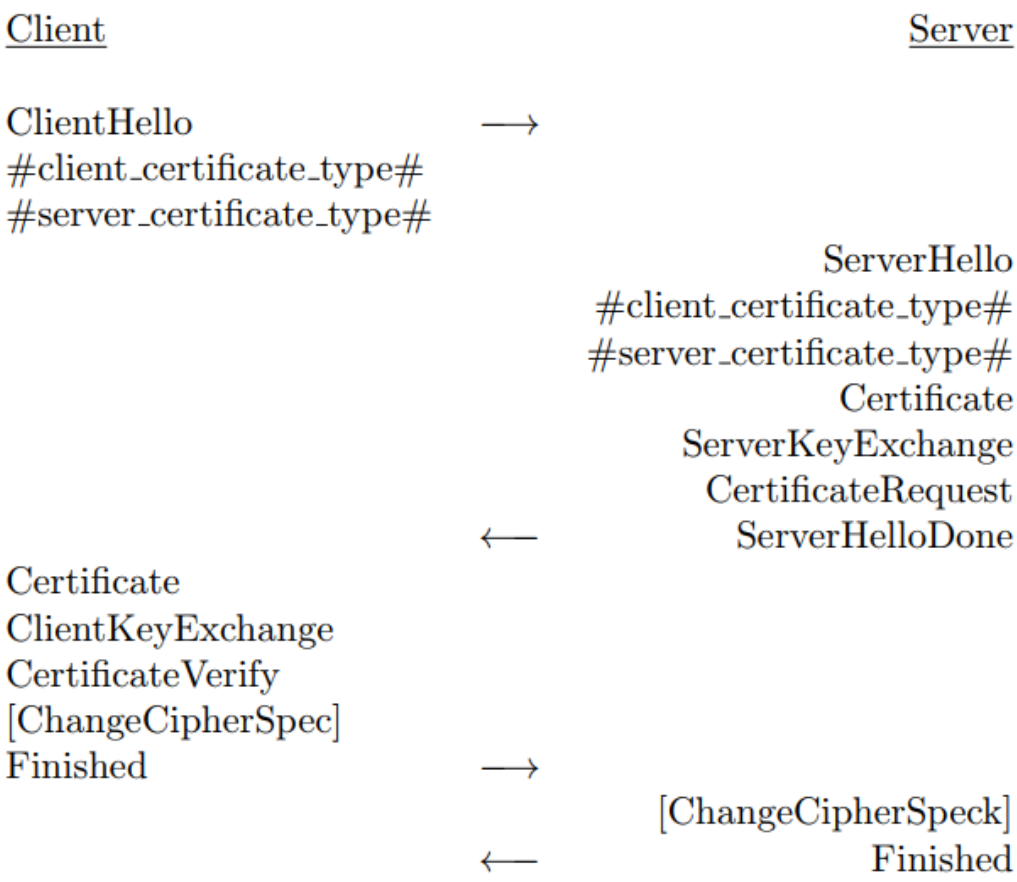
| 2 bytes | N bytes | 2 bytes | N bytes |
|---|---|---|---|
| N | 0....0 | N | PSK |

,where N = PSK byte size

الشكل 5: البناء المسبق السري للمفاتيح

يتطلب استخدام مفاتيح مشتركة مسبقًا أن يتم أنشاء تواصل العقد مسبقًا مع أسرار لكل اتصال مرغوب وكذلك مع هويات مطابقة لهذه الأسرار. يجب استخدام مخططات لتهيئة الأجهزة مع مفاتيح وذلك لتحديث هذه المفاتيح بمرور الوقت. وهناك عيب محتمل من استخدام تبادل المفاتيح هو أنه لا توجد سرية تامة. السرية التامة للأمام هي خاصية خاصة بمخططات التبادل الرئيسية، والتي تضمن أنه حتى لو تم اختراق تبادل المفاتيح لجلسة معينة، تظل البيانات المتبادلة في الجلسات السابقة محمية. على سبيل المثال، توفر طريقة تبادل المفاتيح المؤقتة Diffie Hellman سرية تامة، لأنها لا تستخدم معلمات ثابتة لإجراء تبادل المفاتيح.

المفاتيح العمومية الأولية عند استخدامها، يستمر تدفق رسائل مصافحة DTLS كما هو موضح في الشكل 6.




*الشكل 6:تدفق رسائل المصافحة مع استخدام المفاتيح العامة*

يتم استخدام رسالة الشهادة لنقل المفتاح العام للملقم. ويجب أن تتضمن رسائل hello الخاصة بالملقم والخادم نوع امتدادات شهادة الخادم والملقم للإشارة إلى دعمهم للمفاتيح العامة الأولية. إذا وافق الخادم والملقم على استخدام المفاتيح العامة الأولية للمصادقة، فإنهما يستخدمان نهجًا للتحقق من مفتاح النطاق. تعتبر المفاتيح العامة الأولية هي الأرضية المشتركة بين المفاتيح والشهادات المشتركة مسبقًا. يتم تجنب التحميلا الزائد للشهادة على العقد، ولكن قد لا تزال الاتصالات مستمرة في تنفيذ عمليات المفتاح العام التي تحتاج لعمليات معالجة كبيرة.

قام IETFباختيار TLS ECDHE ECDSA WITH AES 128 CCM 8 باعتباره إلزاميًا لدعم مجموعة التشفير في البيئات المقيدة عند استخدام المفاتيح العامة الأولية. توفر مجموعة التشفير سرية

تامة. ومع ذلك، فإن الحفاظ على المفاتيح العامة الأولية لكل جهاز لا يزال يتطلب بعض الحذر. علاوة على ذلك، فإن المصافحة ذات الصلة تتضمن رسائل أكثر بكثير بالإضافة إلى عمليات مكلفة حسابياً لإجراء تبادل المفاتيح.

**الشهادات**

عند استخدام الشهادات، تكون عملية المصافحة مماثلة لتلك الخاصة بالمفاتيح العامة الأولية. ومع ذلك، يتم تبادل الشهادات والتحقق منها .يتطلب استخدام الشهادات أن يكون لدى الأجهزة زوج مفاتيح عمومي خاص بالإضافة إلى شهادة خاصة بها .بالإضافة إلى ذلك، يجب تهيئة الأجهزة باستخدام المفاتيح العامة لكل وكالة توثيق موثوق بها بالإضافة إلى قوائم إلغاء الشهادات .سيؤدي دعم مجموعات التشفير ذات الصلة إلى مصافحات أطول وأكثر تكلفة من الناحية الحسابية. وعلاوة على ذلك، ستكون الرسائل المتبادلة كبيرة مقارنة باستخدام مفاتيح مشتركة مسبقًا أو مفاتيح عامة، لأن الشهادات تميل إلى أن تكون بطول مئات البايتات، وفي الغالب الأعم لا يتم إرسال شهادات متعددة في سلسلة. بالنسبة لشهادات العقد المقيدة الواردة للمعالجة، سوف تتطلب الشهادات مساحة تخزين كبيرة ووظيفة التحقق من الشهادة.

TLS ECDHE ECDSA WITH AES 128 CCM 8تم تعريفه بأنه الأكثر تطبيقاً لمجموعة التشفير للتطبيقات التي ترغب في استخدام الشهادات. الفائدة هنا هي أن مجموعة التشفير هذه توفر سرية تامة. آخر شيء يجب مراعاته هو أنه حتى لو كان من الممكن التحكم في هيكل وحجم الشهادة، فإن الحفاظ على الشهادات على جميع الأجهزة ذات الصلة يمثل مشكلة كبيرة كما هو الحال مع الطرق السابقة.

## التشفير المصادق عليه AEAD Authenticated Encryption with Associated Data

تحتوي عملية التشفير المصادق عليه على أربعة مدخلات، كل منها عبارة عن سلسلة الثمانية :

- مفتاح السرية K، والذي يجب أن يتم إنشاؤه بطريقة عشوائية أو شبه عشوائية.
- nonce N يجب أن تكون كل الـ nonce المستخدمة في استدعاءات متمايزة لعملية التشفير المصادق تكون مختلفة ومتمايزة عن الاستدعاءات الأخرى، لأي قيمة معينة للمفتاح، ما لم يكن كل nonce هو بطول صفري. يجب أن تستخدم التطبيقات التي يمكنها توليد إحصاءات منفصلة في طريقة تشكيلnonce ، وقد تستخدم أي طريقة أخرى تلبي متطلبات التفرد. ويتم إلزام بعض التطبيقات على استخدام nonces ذات طول صفري.
- نص عادي P، يحتوي على البيانات المراد تشفيرها والمصادقة عليها.
- البيانات المرتبطةA ، التي تحتوي على البيانات المراد توثيقها، ولكن غير مشفرة وهناك خرج واحد.
- النص المشفر C، وهو على الأقل بطول النص العادي، وإلا لا يمكن تنفيذ عملية التشفير المطلوبة. جميع المدخلات والمخرجات عبارة عن سلاسل ثمانية متغيرة الطول، أطوالها تخضع للقيود التالية :عدد الثمانيات في المفتاح K هو بين 1 و255. بالنسبة لكل خوارزميةAEAD ، يجب أن يكون طول K ثابتًا.

### إرشادات حول استخدام خوارزمياتAEAD

تقدم هذه الفقرة النصيحة التي يجب اتباعها من أجل استخدام خوارزمية AEAD بأمان. إذا كان التطبيق غير قادر على تلبية متطلبات التفرد على إنشاءnonce، فيجب عليه استخدام nonce ذات طول صفري. والخوارزميات العشوائية، مناسبة للاستخدام مع هذه التطبيقات. خلاف ذلك، يجب أن يستخدم التطبيق

nonces بطول اثني عشر من الثمانيات. نظرًا لأن الخوارزميات يتم تشجيعها لدعم هذا الطول، يجب أن تستخدم التطبيقات هذا الطول للمساعدة في التشغيل البيني.

**المتطلبات على توليد Nonce**

من الضروري للأمن أن يتم إنشاء nonce بطريقة تحترم شرط أن تكون كل قيمة nonceمتميزة لكل استدعاء لعملية التشفير المصادق عليه، لأي قيمة ثابتة للمفتاح. في هذا القسم، نوجه الانتباه إلى بعض النتائج لهذا المطلب في سيناريوهات مختلفة عندما تكون هناك أجهزة متعددة تقوم بالتشفير باستخدام مفتاح واحد، يجب أن تقوم هذه الأجهزة بالتنسيق لضمان أن تكون قيمة فريدة.

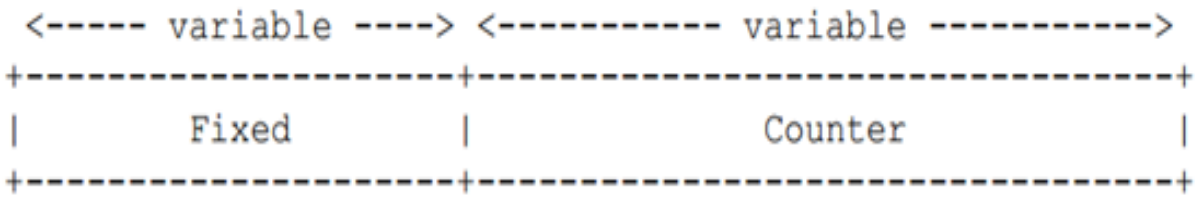
تتمثل إحدى الطرق البسيطة للقيام بذلك في استخدام تنسيق غير متماثل يحتوي على حقل مميز لكل جهاز من الأجهزة، كما هو موضح في الفقرة التالية. لاحظ أنه لا توجد حاجة لتنسيق تفاصيل الشكل غير المصنف بين فك التشفير والتشفر، طالما يتم إرسال nonce بأكمله أو تخزينها مع النص المشفر، وبالتالي تتوفر لدى جهة فك التشفير. إذا لم يكن nonce الكامل متاحًا لفاك التشفير، فسوف يحتاج فاك التشفير إلى معرفة كيفية هيكلة nonce حتى يتمكن من إعادة بنائه. يجب أن توفر التطبيقات محركات التشفير مع بعض الحرية في اختيار nonces الخاصة بهم؛ على سبيل المثال، يمكن أن يحتوي nonce على كلاً من العداد والحقل الذي تم تعيينه بواسطة المشفر ولكن لا يتم معالجته بواسطة جهاز الاستقبال. تتيح هذه الحرية لمجموعة من أجهزة التشفير التنسيق بسهولة أكبر لضمان تمايزها. إذا تم استخدام مفتاح سري لفترة زمنية طويلة، على سبيل المثال، عبر عمليات إعادة تمهيد متعددة، فسيحتاج الأمر إلى تخزين عنصر nonce في الذاكرة. في مثل هذه الحالات، من الضروري استخدام إشارة التحقق من nonce؛ بمعنى، يجب تخزين قيمة nonce الحالية لتوفير معلومات الحالة اللازمة لاستئناف التشفير في حالة حدوث فشل غير متوقع. إن إحدى الطرق البسيطة لتوفير تأكيد عالٍ على عدم استخدام قيم غير متماثلة بشكل متكرر هي الانتظار حتى تتلقى عملية التشفير تأكيدًا من عملية التخزين تشير إلى

أنه قد تم تخزين قيمة nonce التالية بالفعل. نظرًا لأن هذه الطريقة قد تضيف وقتًا طويلاً، فقد يكون من المرغوب فيه تخزين قيم غير متماثلة ويتم أعداد عدة قيم في التسلسل. على سبيل المثال، يمكن تخزين 100 nonce، وبعد ذلك يمكن استخدام القيم من 1 إلى 99 للتشفير. يمكن تخزين 200قيمة nonce في نفس الوقت الذي يتم فيه استخدام القيم من 1 إلى 99، وهكذا يمكن تجنب العديد من المشاكل المتعلقة بإعادة استخدام nonce عن طريق تغيير مفتاح في وضع يكون فيه تنسيق nonce صعبًا. يجب أن تصف كل خوارزمية AEAD ما الذي سينتج عن التدهور الأمني من إعادة الاستخدام غير المقصود لقيمةnonce .

**توصيات لتشكيلNonce**

يوصى بالطريقة التالية لبناء nonces يتم تنسيق nonce كما هو موضح في الشكل 7، مع الثمانيات الأولية التي تتكون من حقل ثابت، والثماني الأخيرة التي تتكون من حقل عداد.

لكل مفتاح ثابت، يتم إصلاح طول كل من هذه الحقول وبالتالي طول nonce، يجب أن تدعم التطبيقات nonces طول 12 ثمانية حيث يكون حقل العداد بطول أربعة ثمانيات




*الشكل 7:نسق بناء الـ nonce*

تشكل حقول العداد الخاصة بالأرقام المتعاقبة تسلسلًا متزايدًا رتيبًا، عند اعتبار تلك الحقول أعدادًا صحيحة بدون إشارة في ترتيب بايت الشبكة. يجب أن يبقى طول حقل العداد ثابت لكافة القيم التي تم إنشاؤها لجهاز تشفير محدد. يجب أن يكون جزء العداد مساويًا للصفر بالنسبة للعدد الأول، وزيادة

بمقدار واحد لكل تعاقب nonceيتم إنشاؤه. ومع ذلك، قد يتم تخطي أي قيمة عداد معينة، وتركها خارج تسلسل القيم المستخدمة، إذا كانت ملائمة. على سبيل المثال، يمكن لأحد التطبيقات اختيار تخطي القيمة الأولية الصفرية للعداد، وتعيين حقل العداد الأولي الخاص بـ nonce بقيمة 1. وبالتالي، يمكن أن تنشأ على الأقل $2 \wedge (8*C)$ nonces عندما يكون طول حقل العداد C هو ثماني بتات. يجب أن يظل الحقل الثابت ثابتًا لجميع القيم التي تم إنشاؤها لجهاز تشفير محدد.

# 6. المراجع


[1] J. Zheng, D. Simplot-Ryl, C. Bisdikian, and H. T. Mouftah,"The internet of things [Guest Editorial]," IEEE Commun. Mag., vol. 49, no. 11, pp. 30–31, Nov. 2011.

[2] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges," in 2012 10th International Conference on Frontiers of Information Technology (FIT), 2012, pp. 257–260.

[3] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," Ad Hoc Netw., vol. 10, no. 7, pp. 1497–1516, Sep. 2012.

[4] J. Höller, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, and D. Boyle, "Chapter 2 - M2M to IoT– The Vision," in From Machine-To-Machine to the Internet of Things, J. Höller, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, and D. Boyle, Eds. Oxford: Academic Press, 2014, pp. 9–37.

[5] R. Wittmann and M. Zitterbart, Multicast Communication: Protocols, Programming, & Applications. Morgan Kaufmann, 2000.

[6] R. Roman, P. Najera, and J. Lopez, "Securing the Internet of Things," Computer, vol. 44, no. 9, pp. 51–58, Sep. 2011.

[7] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, and H.-Y. Du, "Research on the architecture of Internet of Things," in 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), 2010, vol. 5, pp. V5–484–V5–487.

[8] N. Salman, I. Rasool, and A. H. Kemp, "Overview of the IEEE 802.15.4 standards family for Low Rate Wireless Personal Area Networks," in 2010 7th International Symposium on Wireless Communication Systems (ISWCS), 2010, pp. 701–705.

[9] Deering, S. and Hinden, R., "Internet Protocol, Version 6 (IPv6), Specifcation." Network Working Group, Dec-1998.

[10] Postel, J., "RFC 768. User Datagram Protocol." 28-Aug-1980.

[11] G. Mulligan, "The 6LoWPAN Architecture," in Proceedings of the 4th Workshop on Embedded Networked Sensors, New York, NY, USA, 2007, pp. 78–82.

[12] Z. Shelby, K. Hartke, and C. Bormann, "RFC 7252. The Constrained Application Protocol (CoAP)." Internet Engineering Task Force, Jun-2014.

[13] P. J. Leach, T. Berners-Lee, J. C. Mogul, L. Masinter, R. T. Fielding, and J. Gettys, "RFC 2616. Hypertext Transfer Protocol -- HTTP/1.1." Network Working Group, Jun-1999.

[14] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Comput. Netw., vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[15] L. Tan and N. Wang, "Future internet: The Internet of Things," in 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), 2010, vol. 5, pp. V5–376– V5–380.

[16] A. Rahman and E. Dijk, "RFC 7390. Group Communication for the Constrained Application Protocol (CoAP)." Internet Engineering Task Force, Oct-2014.

[17] Tim Dierks, "RFC 5246. The Transport Layer Security (TLS) Protocol Version 1.2." Network Working Group, Aug-2008.

[18] "RFC 793. TRANSMISSION CONTROL PROTOCOL." Sep-1981.

[19] Eric Rescorla and Nagendra Modadugu, "RFC 6347. Datagram Transport Layer Security Version 1.2." Internet Engineering Task Force, Jan-2012.

[20] O. Garcia-Morchon, S. Kumar, A. Rahman, E. Dijk, and S. Keoh, "DTLS-based Multicast Security in Constrained Environments, draft-keoh-dice-multicast-security-08." DICE Working Group, 03-Jul-2014.

*[21] M. Tiloca, "Efficient Protection of Response Messages in DTLSBased Secure Multicast Communication," in Proceedings of the 7th International Conference on Security of Information and Networks, New York, NY, USA, 2014, pp. 466:466–466:472.*

*[22] SICS Network Embedded Systems Group, "Contiki OS." [Online]. Available: http://www.contiki-os.org/. [Accessed: 20- May-2015].*

*[23] Tim Winter et al., "RFC 6550. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks." Internet Engineering Task Force, Mar-2012.*

*[24] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," IEEE Commun. Surv. Tutor. vol. 15, no. 3, pp. 1389–1406, Third 2013.*

*[25] C. Sammarco and A. Iera, "Improving Service Management in the Internet of Things," Sensors, vol. 12, no. 9, pp. 11888– 11909, Aug. 2012.*

*[26] S. Pollin, M. Ergen, S. Ergen, B. Bougard, L. Der Perre, I. Moerman, A. Bahai, P. Varaiya, and F. Catthoor, "Performance Analysis of Slotted Carrier Sense IEEE 802.15.4 Medium Access Layer," IEEE Trans. Wirel. Commun., vol. 7, no. 9, pp. 3359–3371, Sep. 2008.*

*[27] N. Kushalnagar, G. Montenegro, D. E. Culler, and J. W. Hui, "RFC 4944. Transmission of IPv6 Packets over IEEE 802.15.4 Networks." Network Working Group, Sep-2007.*

*[28] J. Ko, A. Terzis, S. Dawson-Haggerty, D. E. Culler, J. W. Hui, and P. Levis, "Connecting low-power and lossy networks to the internet," IEEE Commun. Mag., vol. 49, no. 4, pp. 96–101, Apr. 2011.*

*[29] K. Egevang and P. Francis, "The IP Network Address Translator (NAT)," 1994.*

*[30] "How TLS/SSL Works: Logon and Authentication." [Online]. Available: https://technet.microsoft.com/enus/library/cc783349(v=ws.10).aspx. [Accessed: 04-Apr-2018].*

*[31] H. Krawczyk, R. Canetti, and M. Bellare, "RFC 2104. HMAC: Keyed-Hashing for Message Authentication." Network Working Group, Feb-1997.*

*[32] National Institute of Standards and Technology, "FIPS PUB 180-4. Secure Hash Standard (SHS)." Federal Information Processing Standards, Mar-2012.*

*[33] TLS/DTLS Profiles for the Internet of Things. Tech. rep. url: https://tools.ietf.org/html/draft-ietf-dice-profile-17 (visited on 05/12/2016).*

*[34] GSAKMP: Group Secure Association Key Management Protocol. Tech. rep. url: https://tools.ietf.org/html/rfc4535 (visited on 06/01/2016).*

*[35] MIKEY: Multimedia Internet KEYing. Tech. rep. url: https://tools.ietf.org/html/rfc3830 (visited on 06/01/2016).*

*[36] The Multicast Group Security Architecture. Tech. rep. url: https://tools.ietf.org/html/rfc3740 (visited on 06/01/2016).*

*[37] Multicast Security (MSEC) Group Key Management Architecture. Tech. rep. url:ttps://tools.ietf.org/html/rfc4046 (visited on 06/01/2016).*

*[38] DTLS-based Multicast Security in Constrained Environments draft-keoh-dice-multicastsecurity-08. Tech. rep. url: https://tools.ietf.org/html/draft-keoh-dicemulticast-security-08 (visited on 05/22/2016).*

*[39] T. Hardjono and B. Weis, "RFC 3740. The Multicast Group Security Architecture." Network Working Group, Mar-2004.*

*[40] U. Meth, A. Colegrove, G. Gross, and H. Harney, "RFC 4535. GSAKMP: Group Secure Association Key Management Protocol." Network Working Group, Jun-2006.*

*[41] D. Bailey and D. McGrew, "RFC 6655. AES-CCM Cipher Suites for Transport Layer Security (TLS)." Internet Engineering Task Force, Jul-2012.*

[42] A. Choudhury, D. McGrew, and J. Salowey, "RFC 5288, AES Galois Counter Mode (GCM) Cipher Suites for TLS." Network Working Group, Aug-2008.

[43] National Institute of Standards and Technology, "FIPS PUB 197. Advanced Encryption Standard (AES)." Federal Information Processing Standards, 26-Nov-2011.

[44] Sandeep Kumar, "Group Communication Security for LowPower and Lossy Networks (LLNs), draft-kumar-dicegroupcomm-security-00." DICE Working Group, 02-Jul-2014.

[45] K. Seo and S. Kent, "RFC 4301. Security Architecture for the Internet Protocol." Network Working Group.

[46] Stephen Kent, "RFC 4303. IP Encapsulating Security Payload (ESP)." Network Working Group, Dec-2005.

[47] Stephen Kent, "RFC 4302. IP Authentication Header." Network Working Group.

[48] Charlie Kaufman, "RFC 4306. Internet Key Exchange (IKEv2) Protocol." Network Working Group, Dec-2005.

[49] N. Doraswamy and D. Harkins, IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks, Second Edition, Second. Prentice Hall, 2003.

[50] D. Ignjatic, G. Gross, and B. Weis, "RFC 5374. Multicast Extensions to the Security Architecture for the Internet Protocol." Network Working Group, Nov-2008.

[51] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, and U. Roedig, "Securing communication in 6LoWPAN with compressed IPsec," in 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011, pp. 1–8.

*[52] N. Modadugu and E. Rescorla, "The Design and Implementation of Datagram TLS.," in NDSS, 2004.*

*[53] D. A. McGrew and J. Viega, "The Security and Performance of the Galois/Counter Mode (GCM) of Operation," in Progress in Cryptology - INDOCRYPT 2004, A. Canteaut and K. Viswanathan, Eds. Springer Berlin Heidelberg, 2004, pp. 343–355.*

[54] "An Introduction to Cooja," GitHub. [Online]. Available: https://github.com/contiki-os/contiki. [Accessed: 20-May-2015].

[55] *Kovatsch, Matthias, Duquennoy, Simon, and Dunkels, Adam, "Erbium (Er) REST Engine and CoAP Implementation for Contiki." [Online]. Available: http://people.inf.ethz.ch/mkovatsc/erbium.php. [Accessed: 01- Jun-2015].*

[56] *"Contiki Wiki. Radio Duty Cycling,"* GitHub, *14-Dec-2014. [Online]. Available:* https://github.com/contiki-os/contiki. *[Accessed: 11-Jun-2015].*

[57]contiki-ng/contiki-ng. (n.d.). Retrieved October 18, 2019, from https://github.com/contiki-ng/contiki-ng/wiki/Documentation:-Energest.