تأمين مجموعات البث المتعدد في مركز توزيع المفاتيح

***أ.د. أحمد الكردي، ** محمد الرسلان**

***كلية الهندسة المعلوماتية–جامعة ادلب – إدلب – سوريا**

***كلية الهندسة المعلوماتية–جامعة ادلب – إدلب – سوريا**

ملخص

يعدّ مركز توزيع المفاتيح طرفاً ثالثاً موثوقاً، ومسؤولاً عن إدارة توزيع وتحديث مفاتيح التشفير في نظام آمن. يساعد مركز توزيع المفاتيح على بناء اتصال ثنائي (طرف– إلى– طرف) بين طرفين يرغبان بإجراء جلسة منطقية محمية بينهما. مع تطور نظم الاتصالات، أصبحت الخدمات تعتمد في عملها على اتصالات البث المتعدد، وذلك لإرسال بيانات بشكل آمن، من طرف إلى عدة أطراف أخرى، إضافة إلى قابلية التوسع والمرونة التي توفرها هذه الأنظمة، من خلال إضافة أو إزالة المستخدمين، ودمج أو تقسيم مجموعات الاتصال، مما يساعد على تسريع نشر هذه الخدمات على الشبكات.

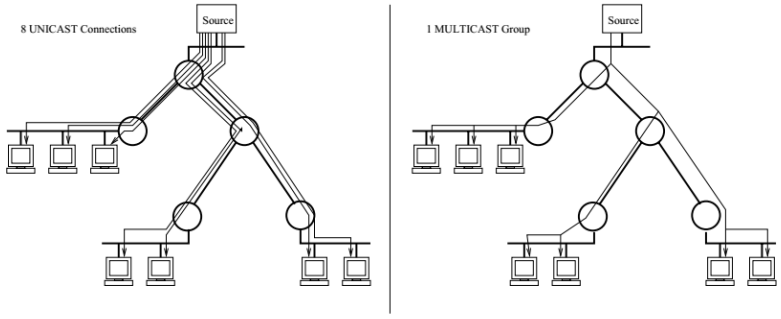
بالرغم من هذا التطور، تعتبر عملية إدارة المفاتيح في نظم مجموعات البث المتعدد مهمة صعبة، من حيث تأمين سرية الاتصال، قابلية التوسع، والمرونة.

في هذا البحث، نقدم نهجاً لملائمة عمل مركز توزيع المفاتيح لإدارة مفاتيح التشفير في نظام مجموعات البث المتعدد. يتلخص هذا النهج في تقديم إجراءات توفر المرونة في إضافة أو إزالة المستخدمين، ودمج أو تقسيم مجموعات الاتصال، إضافة إلى التحديث الدوري الملائم لهذا النهج. نستعرض عدد الرسائل المتبادلة، وعدد عمليات التشفير في كل إجراء.

الكلمات المفتاحية: البث المتعدد، البث الأحادي، التحديث الدوري، سرية الاتصال، التشفير.

1- مقدمة

يعتبر البث المتعدد اتصالاً أساسياً في الشبكات العالمية (انظر الشكل **1**)، يتيح لعقدة في شبكة ما إرسال وحدة بيانات إلى مجموعة من العقد المستلمة برسالة واحدة، لتجنيب هذه العقدة عملية إرسال نسخ منفصلة من البيانات، لكل عقدة هدف على حدة، باستخدام البث الأحادي [**18,26**].



الشكل 1. البث الأحادي والبث المتعدد.

مع النمو والتطور السريع في البنية التحتية للشبكات العالمية، أدى ذلك لتطور سريع في التطبيقات والخدمات المقدمة عبر الانترنت، بحيث تجمع بين الصوت والصورة والنص [**5,25**]. مع أن غالبية الاتصالات للخدمات المتوفرة كانت أحادية البث، إلا أن هناك تزايد في الحاجة لاتصالات البث المتعدد، خصوصا أن التطبيقات والخدمات المطورة أصبحت تعتمد على مبدأ التشاركية والمجموعات في عملها مثل مؤتمرات الفيديو، وألعاب المجموعات التفاعلية، والتلفاز عبر الانترنت، والتعليم الالكتروني المنظم [**3,7,18,22**].

ومع هذ التطور، فإن إدارة المفاتيح، وتوفير الأمان في المجموعات يعد إحدى العوائق الكبيرة التي تقف في وجه هذا الانتشار[**4**]. تكمن الصعوبة في أن المجموعات قابلة للتوسع، وديناميكية [**15**]، فالمستخدمون يغادرون و/أو ينضمون إلى المجموعات في لحظات مختلفة [**3,24,25**]. إضافة لذلك، تتميز بإمكانية دمج وتقسيم مجموعات الاتصال، تبعا لمتطلبات التطبيق [**11,20,21**]. تتلخص المشكلة في تصميم بروتوكول يوزع مفتاح المجموعة بشكل آمن، على جميع الأعضاء ويقوم بتحديث هذا

المفتاح، بحيث يمكن للأعضاء الحاليين فقط في المجموعة الحصول عليه مهما كانت التغيرات التي تطرأ على المجموعة من خلال الاضافة أو الإزالة للمستخدمين والدمج أو التقسيم للمجموعات [**5,15,16**] .

ناقشت العديد من الأبحاث أساليب وطرائق لتحقيق اتصال آمن بين عناصر مجموعة. طرح [**9**] نهجاً لتحقيق الاتصال الجماعي باستخدام خوارزمية (Diffie-Hellman) **DH** المستخدمة لتأسيس جلسة ثنائية آمنة (Two-party DH Key Exchange)، بحيث يتم تطوير هذه الخوارزمية، لتأسيس اتصال آمن (N-party DH Key Exchange) بين مجموعة من n طرف في ببروتوكول (Group Diffie-Hellman) **GDH** وتم تطوير هذا البروتوكول على عدة مراحل، نظراً للكلفة الحسابية للنسخة الأولى. وقدم [**5,6**] نهجاً لإدارة المفاتيح اللازمة لتأمين اتصالات مجموعة باستخدام بروتوكول **GKMP** (Group Key Management Protocol). وهناك أبحاث ركزت في محتواها على نظام غير مركزي لإدارة المفاتيح كما في [**13,15**]، فقد تم استخدام نظام إدارة متعدد المستويات، تُقسم فيه المجموعة إلى عدة مجموعات فرعية، لكل مجموعة فرعية متحكم يدعى **GSI** (Group Security Initiator)، حيث يكون متحكم المجموعة الفرعية مسؤولاً عن تأمين سرية الاتصالات التي تتم بين المستخدمين ضمن المجموعة الفرعية. وفي [**16**] تم استعراض نهج لإدارة المفاتيح للمجموعات يعتمد على تقسيم النطاق (**domain**) إلى مناطق صغيرة (**area**)، حيث يتوضع المستخدم في منطقة واحدة من هذه المناطق. في هذا النهج لدينا نوعان من مراكز التحكم، متحكم خاص لكل النطاق، ومتحكم خاص بكل منطقة.

وفقا لكل ما سبق، يمكن تلخيص المتطلبات اللازمة لتأمين المجموعات وفق النقاط الآتية [**17,11,15,19**]:

- **التحقق من هوية المستخدمين:** وتتم هذه العملية عند اشتراك المستخدم في التطبيق الذي يديره الخادم المسؤول عن إدارة النظام كاملا، لمنع أي مستخدم غير مخول من الحصول على الخدمات التي يوفرها النظام

- **قابلية التوسع:** وتعني امكانية إضافة أعضاء إلى المجموعات والخدمات، وإزالتهم أو مغادرتهم بشكل ديناميكي ومرن.
- **السرية اللاحقة:** بحيث لا يكون للمستخدمين الذين غادرو المجموعة، إمكانية معرفة أو اختراق سرية المعلومات المتبادلة بين المستخدمين ضمن المجموعة في المستقبل بعد مغادرتهم.
- **السرية السابقة:** بحيث لا يكون للمستخدمين الذين انضموا إلى المجموعة، إمكانية معرفة أو اختراق سرية المعلومات المتبادلة بين المستخدمين ضمن المجموعة في الماضي قبل انضمامهم.
- **دمج مجموعات الاتصال:** وتعني إمكانية دمج مجموعتي مستخدمين مشتركين في اتصال جماعي، بمجموعة واحدة، بحيث يتم المحافظة على السرية السابقة واللاحقة.
- **تقسيم مجموعة اتصال:** وتعني إمكانية تقسيم مجموعة مستخدمين مشتركين في اتصال جماعي، إلى مجموعتي اتصال، ويعتمد هذا المبدأ على التطبيق ويحدث هذا عندما يقرر التطبيق تقسيم المجموعة إلى عدة مكونات أو استبعاد عدة أعضاء في وقت واحد.
- **عدم الترابط بين المفاتيح:** الهدف من ذلك منع المستخدمين من استنباط المفاتيح المخصصة للمجموعات الأخرى التي لا ينتمون لها من خلال مفتاح مجموعتهم.

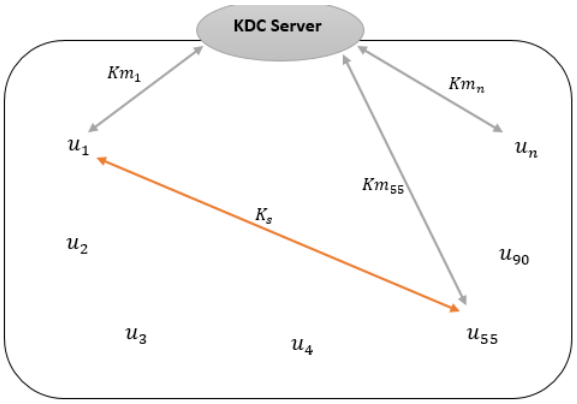
2- المجموعات الآمنة في مركز توزيع المفاتيح

يُستخدم مركز توزيع المفاتيح، لتأمين السرية في الاتصالات الثنائية (**end-to-end**)، بحيث يكون مسؤولاً عن تأسيس جلسة موثوقة بين طرفين، من خلال توزيع مفتاح مشترك (**shared secret key**) للطرفين، لتشفير البيانات المرسلة من الطرف الأول، وإرسالها ضمن شبكة عامة ليتم فك تشفيرها عند الطرف الآخر (المستقبل)، ويعتمد على مستويين من المفاتيح [**1,2**] هما:

1. مفتاح الجلسة K_S (session key): هو مفتاح متناظر، يستخدم لتشفير البيانات المرسلة بين طرفين حيث يحتاج المستخدمان لتشكيل هذا المفتاح لكل جلسة.

2. المفتاح الرئيسي K_m (master key): لكل مستخدم مفتاح رئيسي وحيد مشترك بينه وبين مركز توزيع المفاتيح. يتم توزيع هذا المفتاح على المستخدمين بطريقة ممكنة، كتسليمه فيزيائيا أو عبر البريد الالكتروني أو تبادل هذا المفتاح عبر خوارزمية **DH** (Diffie-Hellman) [**10**].

في حال طلب أحد المستخدمين تأمين الاتصال الثنائي (طرف–إلى–طرف) بينه وبين مستخدم آخر، يتم تبادل مفتاح جلسة (session key) مشترك بينهما لتشفير المعلومات المتبادلة. حيث يقوم مركز توزيع المفاتيح بتوليد مفتاح لهذه الجلسة وتبادل هذا المفتاح بين **KDC** وكل مستخدم على حدة مشفراً بالمفتاح الرئيسي لهذا المستخدم، وبذلك يحصل كلا المستخدمين على مفتاح جلسة مشترك بينهما [**1**]، كما هو مبين في الشكل **2**.

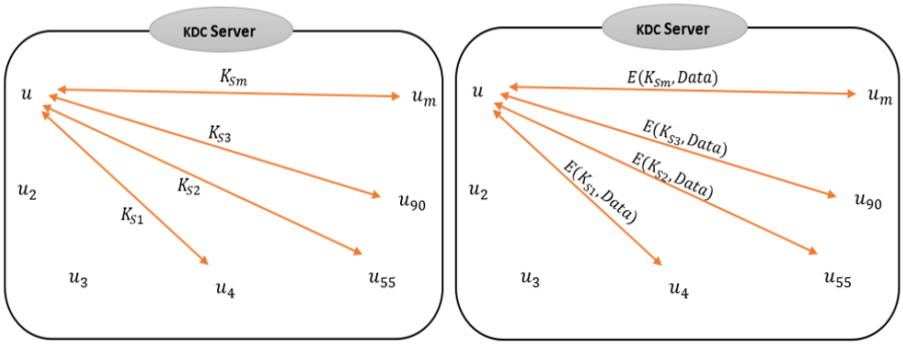


الشكل 2. مركز توزيع المفاتيح – تكوين جلسة.

في هذه الحالة، يقوم كل مستخدم بتخزين مفتاحه الرئيسي المشترك مع الخادم **KDC**، ومفاتيح جلسة، بعدد الجلسات التي طلب تكوينها مع بقية المستخدمين في النظام.

إذا أراد مستخدم ما **u**، الاتصال مع **m** مستخدماً، فإنه يتوجب على نظام **KDC** إنشاء **m** مفتاحٍ مشتركٍ بين المستخدم u وجميع المستخدمين m، الذي يرغب المستخدم **u** بالتواصل معهم، ويتوجب عليه الاحتفاظ بـ **m+1** مفتاحاً مشتركاً لكل مستخدم في المجموعة، إضافة لمفتاحه الرئيسي المشترك مع الخادم **KDC**، كما هو موضح بالشكل **3**.

لإرسال رسالة إلى **m** مستخدماً، يجب على المستخدم **u** إرسال الرسالة نفسها لكل مستخدم على حدة، مشفرةً بمفتاح الجلسة المشترك مع هذا المستخدم، وهذا يعني تشفير الرسالة نفسها **m** مرة، و **m** عملية إرسال (أنظر الشكل **3**). هذا يتطلب عملاً مكلفاً في حال كان عدد المستخدمين كبيراً، (مثلاً **m=10000**).



الشكل 3. تأمين جلسة مع m مستخدم عبر الجلسة الثنائية.

أما إذا احتاجت مجموعة مكونة من **m** مستخدماً للتخاطب فيما بينهم، فإنها تحتاج إلى استخدام $\frac{m(m-1)}{2}$ مفتاح جلسة مشترك، واحد بين كل طرفين وبالاتجاهين، وإلى **(m-1)** عملية تشفير وإرسال لكل رسالة. هذا يعني وجود عددٍ كبيرٍ من المفاتيح التي تحتاج إلى إدارة عملية توزيعها بين الأطراف المختلفين بشكل آمن **[1]**، إضافة إلى عدد كبير من عمليات التشفير والإرسال. علاوة على ذلك، ستزداد المسألة تعقيداً في حال كانت المجموعة قابلة للتوسع من حيث ضم/حذف مستخدمين جدد، وفي حال أردنا دمج/تقسيم مجموعات الاتصال.

يمكننا تلخيص عمل مركز توزيع المفاتيح وفق المهام الآتية:

- ❖ المهمة الأولى: التحقق من المستخدمين بالطريقة الممكنة وتكوين مفتاح رئيسي (master key) مشترك مع كل مستخدم.
- ❖ المهمة الثانية: تكوين جلسات الاتصال الثنائي **(one-to-one)** الآمنة عن طريق توزيع مفتاح الجلسة **(session key)** لكل طرفين يرغبان بالتواصل الآمن.

❖ المهمة الثالثة: إدارة جلســات الاتصـــال الثنائي وإجراء التحديث الدوري لمفاتيح التشفير الخاصة بهذه الجلسات وانهاء الجلسة.

كل ما أتينا على ذكره في هذه المقدمة يعد جوهر البحث في هذه المقالة. يتركز عملنا في هذه المقالة على تطوير مهام مركز توزيع المفاتيح بحيث يكون ملائماً لإدارة المفاتيح اللازمة لتشـفير البيانات المتبادلة في مجموعات البث المتعدد. نعمل على تطوير المهمة الثانية من مهام مركز توزيع المفاتيح (تكوين الجلســـة الثنائية المُؤمنة)، لتكوين مجموعات البث المتعدد الآمنة، بحيث يشترك بها عدة مستخدمين. نقدم أسـلوباً لتنظيم المجموعات في نظام مركز توزيع المفاتيح، وإجراءات عمل توفر المتطلبات اللازمة في المجموعات المذكورة آنفاً، وإجراء التحديث الدوري لمفاتيح التشـفير. نسـتعرض عدد عمليات التشـفير، والإرسـال اللازمة لإدارة توزيع وتحديث المفاتيح في كل إجراء.

3- تنظيم المجموعات في مركز توزيع المفاتيح

يتألف النظام الذي يديره مركز توزيع المفاتيح كما ذكرنا، من خادم هو مركز توزيع المفاتيح (**KDC server**)، ومن مجموعة من المستخدمين المتباعدين، يشترك كل مستخدم مع الخادم بمفتاح رئيسي (**master key**).

بفرض أن مركز توزيع المفاتيح يتضمن لائحة للمستخدمين المُسجلين في النظام (UserList)، تتضمن هذه اللائحة معرف المستخدم، ومفتاحه الرئيسي، كما هو مبين بالجدول الآتي:

User Id	Master Key
u_9	K_{m9}
u_{17}	K_{m17}
u_{20}	K_{m20}
u_1	K_{m1}
u_2	K_{m2}
……	……..

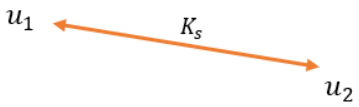
جدول1 . قائمة المستخدمين

يحفظ مركز توزيع المفاتيح أيضا، لائحة الاتصالات الثنائية الآمنة (**SessionList**)، تتضمن هذه اللائحة معرف المستخدمين المشتركين بالجلسة، ومفتاح الجلسة المشترك بينهما، كما هو مبين بالجدول التالي:

User1 Id	User2 Id	Session Key
u_9	u_3	$K_{S\,9-3}$
u_{17}	u_1	$K_{S\,17-1}$
u_2	u_5	$K_{S\,2-5}$
…….		……

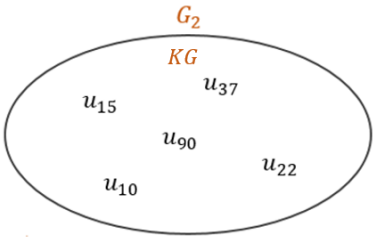
جدول2 . لائحة الاتصالات الثنائية الآمنة

بما أن تأمين الاتصال الثنائي، بين أي مستخدمين في نظام **KDC**، يتشكل باستخدام التشفير بالمفتاح المشترك (**session key**) بينهما، يمكن أن نعتبر أن تأمين مجموعة يتم بتشكيل مجموعة ضمن النظام، عناصرها المستخدمون الراغبون بتأمين التواصل، ومفتاح هذه المجموعة يكون مشتركاً بين المستخدمين. يتم تشفير البيانات المتبادلة باستخدام مفتاح المجموعة (انظر الشكل **4**).



الشكل 4. الجلسة الآمنة.

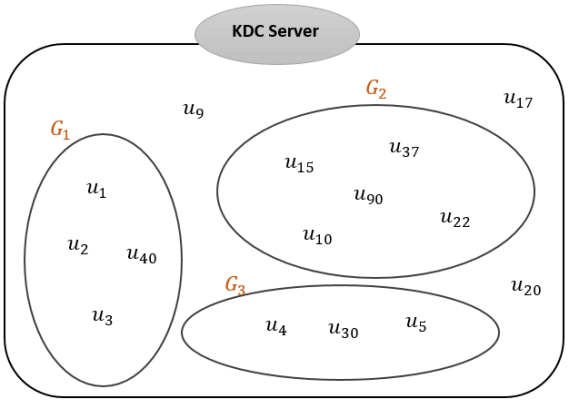
يمكن توسيع المجموعة لتشمل عدداً كبيراً من المستخدمين، بإضافة مستخدمين إلى المجموعة، وتزويدهم مفتاح المجموعة $\boldsymbol{KG}$، وبالتالي ويمكن للمستخدمين ضمن هذه المجموعة تبادل البيانات بينهم باتصال بث متعدد آمن، حيث يتم تشفير البيانات المتبادلة باستخدام مفتاح المجموعة. لا يمكن لاي طرف خارج المجموعة الآمنة ولو استطاع، بطريقة ما، الحصول على البيانات المتبادلة بين عناصر المجموعة، كشف سرية هذه البيانات، كونه لا يمتلك مفتاح المجموعة، وهذا يحفظ أمن المجموعة بشكل كامل. انظر الشكل 5.



الشكل .5 مجموعة آمنة

من أجل إنشاء أكثر من مجموعة آمنة ضمن النظام الذي يديره مركز توزيع المفاتيح، يقوم المركز بعملية التحقق من المستخدم. تضمن هذه العملية اشتراك المركز مع هذا المستخدم بمفتاح رئيسي، وبذلك يكون المستخدم قد أُعطي صلاحية الانضمام إلى النظام بشكل عام، دون صلاحية إنشاء اتصالات آمنة، لأن هذا المستخدم لم يستلم بعد من الخادم **KDC** أي مفتاح لمجموعة. يقوم الخادم **KDC** بتسجيل المستخدم في لائحة المستخدمين (أنظر الشكل **6**).

يبين الشكل 6 نظام مركز توزيع المفاتيح مكوناً من ثلاث مجموعات **G1,G2,G3**، كل مستخدم ضمن مجموعة يحتفظ بحوزته بمفتاحه الرئيسي، ومفتاح المجموعة التي ينتمي اليها، الذي يستخدمه في تشفير البيانات المتبادلة مع المستخدمين ضمن المجموعة الواحدة.



الشكل 6. تنظيم مجموعات البث المتعدد في نظام مركز توزيع المفاتيح

نعرف لائحة المجموعات (**GroupList**)، تتضمن معلومات عن المجموعات في نظام مركز توزيع المفاتيح، كمعرف المجموعة، ومفتاح المجموعة المستخدم لتشفير الاتصال بين المستخدمين ضمن المجموعة، كما في الجدول الآتي:

Group id	Group key
G_1	KG_1
G_2	KG_2
G_3	KG_3

جدول 3. لائحة المجموعات

بالإضافة إلى ذلك، نعرف لائحة الاتصالات المتعددة (**UserGroupList**) في نظام **KDC**، والتي تعبر عن ارتباط المستخدمين مع المجموعات. تتضمن هذه اللائحة معرف المستخدم، ومعرف المجموعة التي ينتمي إليها المستخدم.

بالنسبة للمستخدمين الغير منضمين لأية مجموعة، لن يكون لهم أية صفوف في هذه اللائحة، كما في الجدول التالي:

User Id	Group id
u_1	G_1
u_2	G_1
u_3	G_1
u_{40}	G_1
u_{15}	G_2
u_{10}	G_2
u_{90}	G_2
u_{37}	G_2
u_{22}	G_2
u_4	G_3
u_5	G_3
u_{30}	G_3

جدول 4. لائحة الاتصالات المتعددة

يستعمل المستخدمون ضمن المجموعة **G1** المفتاح $\boldsymbol{KG_1}$ لتشفير البيانات وارسالها ضمن المجموعة **G1**، وكذلك الأمر نفسه بالنسبة للمستخدمين ضمن المجموعتين **.G2,G3**

3-1- إجراء إنشاء مجموعة آمنة (Group Initialize)

يتم في هذا الإجراء تكوين مجموعة آمنة وفقا للنهج المقترح آنفا. بفرض أن اللائحة **listusers** تتضمن المستخدمين الراغبين بإنشاء المجموعة:

```
createMulticastGroup(List<User> listuser)
{
   Group G = newGroup()      // generate new group G with key KG
   GroupList.add(G)
   foreach (User u in listusers){
      messag =E(Km , KG)
      unicast: KDC  --message-->  u
     UserGroupList.add(u, G)
   }
}
```

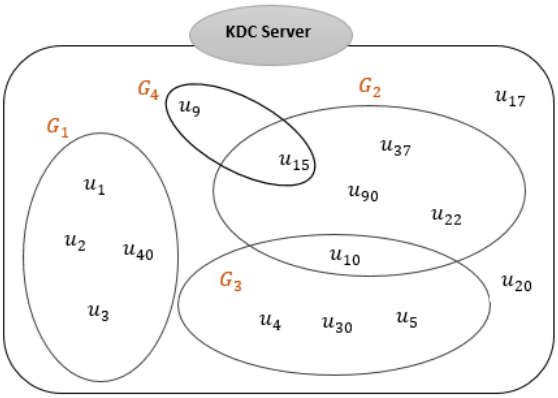
إن تشكيل مجموعة مؤلفة من **n** مستخدماً، يتطلب **n** عملية تشفير، و**n** عملية إرسال أحادي البث، لتسليم مفتاح المجموعة إلى جميع المستخدمين الراغبين بتشكيل المجموعة وهذا موضح بالجدول الآتي:

message id	message	num encryptions	type	num sends
1	$E(K_m , KG)$ → u	n	unicast	n

جدول 5 . احصائيات الارسال والتشفير

يمكن للمستخدم في النظام تأسيس أكثر من اتصال واحد. نعبر عن هذه العملية بتقاطع المجموعتين، كما في الشكل **7**.

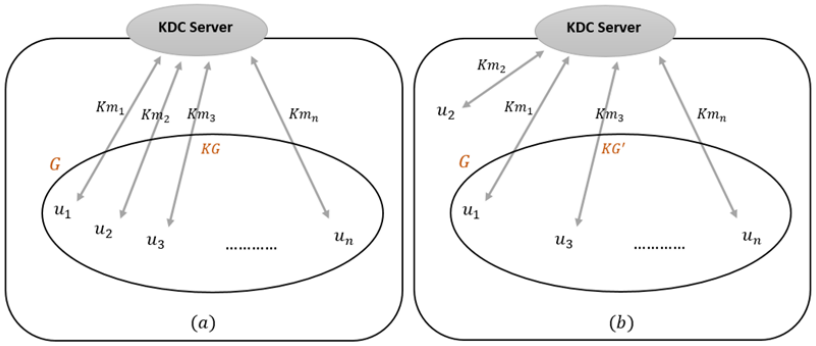
إذ يشترك المستخدم **u10** في المجموعتين **G3,G2**، وبالتالي يحفظ لديه المفتاحين $\boldsymbol{KG_2}, KG_3$ اللذين يستخدمهما في تشفير بياناته المتبادلة مع عناصر كل مجموعة.



الشكل 7. مجموعات متداخلة.

4- إدارة المفاتيح

كما ذكرنا في مقدمة البحث، فإن المجموعات تمتاز بالمرونة من حيث إمكانية إضافة وإزالة المستخدمين، بشكل متكرر، إضافة لإمكانية دمج أو تقسيم مجموعات الاتصال. تتطلب التغيرات في أعضاء المجموعة من الخادم إجراء تحديث لمفتاح المجموعة عند كل تغير، لضمان السرية السابقة والسرية اللاحقة **[13,17,18,15]**. في هذا السياق نقترح إجراءات لتحديث المفاتيح عند إضافة أو إزالة أي مستخدم من المجموعة، وعند دمج أو تقسيم المجموعات.



الشكل 8. تحديث مفتاح مجموعة

نوضح في الشكل **8** عملية إضافة (الشكل **a8**) وإزالة (الشكل **b8**) للمستخدم **u2** من المجموعة **G** التي تتضمن **n** مستخدماً، إذ يحتفظ كل مستخدم بمفتاحه الرئيسي المشترك مع الخادم **KDC** ومفتاح المجموعة **G** التي ينتمي لها.

4-1- إجراء حذف مستخدم من مجموعة (Member Leave/Delete)

نبين من خلال هذا الإجراء كيفية حذف مستخدم من مجموعة في النهج المقترح:

```
deletUser (Group G , User u)
{
    KG′=generatNewKey(KG)
    foreach (element e in UserGroupList where e.groupId= G ){
      u_x=e.userId
       if(u_x≠ u){
          message=E(K_m , KG′)    // where km is master key of user u_x
          unicast: KDC ---message---> u_x
       }
      else
         UserGroupList.remove(e)
    }
}
```

يتم تحديث مفتاح المجموعة، نتيجة لحذف مستخدم **u2** من المجموعة (أنظر الشكل **8b**)، بتوليد مفتاح جديد **KG'**، فإذا أراد **KDC** إرساله إلى بقية المستخدمين في المجموعة، مشفراً باستخدام المفتاح القديم للمجموعة:

message = E(KG , KG')

عندئذ، يستطيع المستخدم u2 الحصول عليه، لأنه يمتلك المفتاح القديم للمجموعة **KG**، لهذا السبب لا بد من إرسال المفتاح إلى كل مستخدم في المجموعة، عدا المستخدم u2، مشفراً عن طريق المفتاح الرئيسي للمستخدم. وفق هذه العملية، نضمن وصول مفتاح المجموعة الجديد إلى كل مستخدم بسرية، مشفراً بمفتاحه الرئيسي، وبالتالي، لا يمكن للمستخدم **u2** الحصول عليه، وكشف سرية البيانات المتبادلة لاحقا ضمن المجموعة. بهذا نحقق عامل السرية اللاحقة ضمن المجموعة. بالنسبة إلى لائحة الاتصالات المتعددة، فيتم حذف الصف الذي يعبر عن ارتباط المستخدم **u** مع المجموعة **G** التي تم حذفه منها كما في الجدول التالي:

User Id	Group id
u_1	G

u_3	G
……..	……
u_n	G

جدول 6. لائحة الاتصالات المتعددة

يحتاج إجراء الحذف إلى **(n-1)** عملية تشفير و **(n-1)** عملية إرسال أحادي البث لكل عنصر من عناصر المجموعة، عدا العنصر المحذوف، حيث **n** عدد عناصر المجموعة، ويوضح الجدول الآتي رسالة التحديث في إجراء الحذف:

message	Message	encryptions	type	sends
1	$E(K_m , KG')$ → every user in group except u	n-1	unicast	n-1

جدول 7. احصائيات الارسال والتشفير

4-2- إجراء إضافة مستخدم إلى مجموعة (Member Join/Add)

نبين من خلال هذا الإجراء كيفية إضافة مستخدم إلى مجموعة وفقاً للنهج المقترح:

```
addUser(Group G , User u)
{
   KG′=generatNewKey(KG)
   message1=E(KG, KG′)    // where KG is old group key
   multicast: KDC ──message1──→ G
   messag2=E(Km , KG′)
   unicast: KDC ──message2──→ u
   UserGroupList.add(u,G)
}
```

لإضافة مستخدم إلى المجموعة **G** (انظر الشكل **8a**)، نجد أن عملية الإضافة تتطلب إرسال مفتاح المجموعة الجديد برسالتين:

- الرسالة الأولى: بث متعدد إلى عناصر المجموعة، تتضمن المفتاح الجديد للمجموعة مشفراً بمفتاحها القديم.
- الرسالة الثانية: أحادية البث إلى المستخدم **u2** تتضمن مفتاح المجموعة الجديد مشفراً بالمفتاح الرئيسي للمستخدم **u2**.

بهذا الشكل يحصل جميع عناصر المجموعة، بما فيهم المستخدم u2 على المفتاح الجديد للمجموعة **'KG**، الذي يُستخدم لتشفير البيانات المتبادلة ضمن المجموعة، وبما أن **u2** لم يحصل على المفتاح القديم للمجموعة **KG**، لذا لا يمكنه كشف سرية البيانات السابقة المتبادلة ضمن المجموعة قبل دخوله إليها، وهذا يوفر عامل السرية السابقة.

يتم إضافة صف إلى لائحة الاتصالات المتعددة، يتضمن معرف المستخدم ومعرف المجموعة التي أُضيف إليها، كما في الجدول الآتي:

User Id	Group id
$\mathbf{u_1}$	G
$\mathbf{u_2}$	G
$\mathbf{u_3}$	G
……..	……
$\mathbf{u_n}$	G

جدول 8. لائحة الاتصالات المتعددة

يحتاج إجراء الضم إلى عملية تشفير واحدة، وعملية إرسال متعدد واحدة، إلى جميع عناصر المجموعة، إضافة إلى عملية تشفير واحدة، وعملية إرسال واحدة، أحادية البث، موجهة للمستخدم المضاف، كما هو مبين في الجدول التالي:

message	Message	encryptions	type	sends
1	$E(KG, KG') \rightarrow G$	1	multicast	1
2	$E(K_m, KG') \rightarrow u$	1	unicast	1

جدول 9. احصائيات الارسال والتشفير

4-3- إجراء دمج مجموعتي اتصال (Merge Groups)

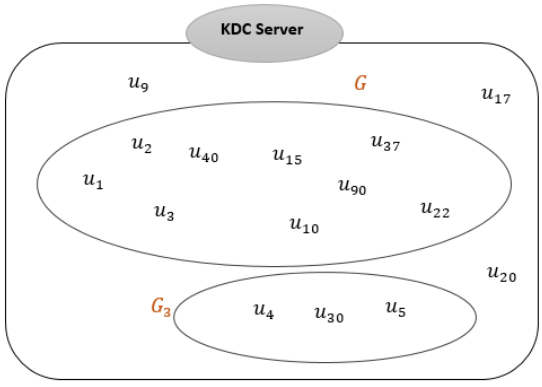
نقدم في هذه الفقرة إجراء لدمج مجموعتين (**margeGroups**)، تتضمن كل مجموعة عدداً من المستخدمين المشتركين باتصال بث متعدد آمن عن طريق مفتاح المجموعة. إن إجراء دمج مجموعة **G1** مكونة من **m** مستخدماً إلى مجموعة **G2**

تتضمن **n** مستخدماً، يتم بإنشاء مجموعة جديدة **G**، وإضافة جميع المستخدمين من المجموعتين **G1,G2** لها:

```
mergeGroups(Group G1 , Group G2)
{
   Group G = newGroup()       // generate new group G with key
KG
   GroupList.add(G)
   message1=E(K$G_1$, $KG$)    // where KG is key for  new group G
   multicast: KDC ──message1──▶ $G_1$

   message2=E(K$G_2$, $KG$)    // where KG is key for  new group G
   multicast: KDC ──message2──▶ $G_2$
  foreach (element e in UserGroupList){
     if(e.groupId= G1 OR e.groupId= G2)
           e.groupId=G
  }
 GroupList.remove(G1)
 GroupList.remove(G2)
}
```

بالاعتماد على النظام الموضح في الشكل **6**، وبتنفيذ إجراء دمج المجموعتين **G1,G2**، يمكن إنشاء مجموعة جديدة **G** بمفتاح جديد **KG**، ومن ثم إضافة هذه المجموعة إلى لائحة المجموعات (أنظر الشكل **9**).



الشكل 9. دمج مجموعتي اتصال.

يتم بعد ذلك إرسال مفتاح المجموعة الجديدة **KG** إلى جميع عناصر كل مجموعة على حدة برسالة بث متعدد، مشفراً باستخدام مفتاح المجموعة القديم، ومن ثم يتم تحديث لائحة المستخدمين، بتحويل خاصية معرف المجموعة التي ينتمي إليها المستخدم (**group id**) لكل المستخدمين في المجموعتين **G1,G2** إلى المجموعة **G**، كما هو مبين في الجدول أدناه:

User Id	Group id
u_1	G
u_2	G
u_3	G
u_{40}	G
u_{15}	G
u_{10}	G
u_{90}	G
u_{37}	G
u_{22}	G
u_4	G_3
u_5	G_3
u_{30}	G_3

جدول 10. لائحة المستخدمين بعد التحديث

وبعد ذلك نحذف المجموعتين **G1,G2** من لائحة المجموعات، لتكون اللائحة كما في الجدول الآتي:

Group id	Group key
G	KG
G_3	KG_3

جدول 11. لائحة المجموعات بعد التحديث

يوضح الجدول التالي الرسائل اللازمة لعملية التحديث. تتطلب عمليةُ التحديثِ عند الدمج عمليتي تشفيرٍ وعمليتي إرسالٍ متعددِ البثِ، كما هو مبين أدناه:

message	message	encryptions	Type	sends
1	E(KG_1, KG) → G1	1	Multicast	1

2	E(KG_2, KG) → G2	1	Multicast	1

جدول 12. احصائيات الارسال والتشفير

4-4- إجراء تقسيم مجموعة اتصال (Group Split)

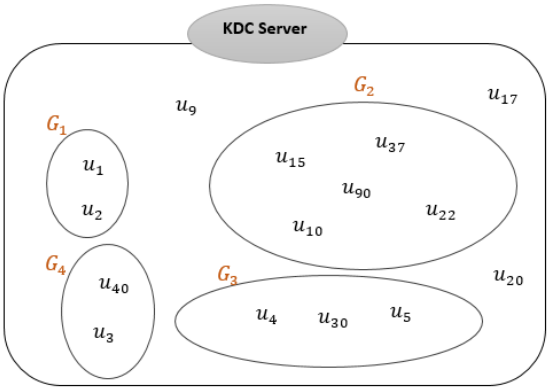
نقدم في هذه الفقرة إجراء لتقسيم مجموعة (**splitGroup**). يتم تقسيم مجموعة إلى مجموعتين، بإنشاء مجموعة جديدة ونقل عدد من المستخدمين إلى المجموعة الجديدة. عند تقسيم مجموعة G_n مكونة من **n** مستخدماً مشتركاً بالاتصال المتعدد، ستتشكل مجموعتان بعد التقسيم، الأولى هي المجموعة الأصلية G_n، والمجموعة الأخرى هي المجموعة الجديدة ولتكن G_m. بفرض **m** هو عدد المستخدمين المنقولين إلى المجموعة G_m، بالتالي يبقى في المجموعة G_n الأصلية **(n-m)** مستخدماً.

نعرف اللائحة (**listusers**) التي تتضمن المستخدمين الذين سيتم حذفهم عن المجموعة الأولى، بغية تقسيمها، ونقلهم إلى المجموعة الجديدة:

```
splitGroup(Group G1 , list<User> listusers) // listusers include list
of user to
{                                                    // transfer them to
new group G2
   Group G2 = newGroup()       // generate new group G2 with key
KG2
   GroupList.add(G2)
   KG1'=generatNewKey(KG1)
  foreach (element e in UserGroupList){
      if(e.groupId= G1){
          u=e.userId
          if(! listusers.contain(u)) {
              message=E(Km , KG1')    // where km is master key of
user u
            unicast: KDC ---message---> u
          }
         else{
            message=E(Km , KG2)    // where km is master key of
user u
            unicast: KDC ---message---> u
```

```
            e.groupId=G2
        }
    }
  }
}
```

بالعودة إلى الشكل **6**، بفرض أننا بصدد تقسيم المجموعة **G1** إلى مجموعتين، وبفرض أن المجموعة الجديدة تتضمن لائحة من المستخدمين **{listusers=u3,u40}**، بالنتيجة سيبقى المستخدمان **u1,u2** ضمن المجموعة **G1** (انظر الشكل **10**).



الشكل 10. تقسيم مجموعة اتصال

وفق إجراء التقسيم الجديد المقترح، يتم إنشاء مجموعة جديدة **G4** وإضافتها إلى لائحة المجموعات مع مفتاحها، كما في الجدول الآتي:

Group id	Group key
G_1	KG_1
G_2	KG_2
G_3	KG_3
G_4	KG_4

جدول 13. لائحة المجموعات

من أجل كل مستخدم من مستخدمي المجموعة **G1:**

- إذا كان هذا المستخدم لا ينتمي إلى لائحة المستخدمين (**listusers**)، المُراد نقلهم إلى المجموعة الجديدة G4، عندئذ سيبقى المستخدم ضمن المجموعة G1، وفي

هذه الحالة يتم تزويده بمفتاح محدث للمجموعة **G1** مشفر باستخدام مفتاحه الرئيسي.

- أما إذا كان المستخدم من ضمن لائحة المستخدمين (**listusers**) المُراد نقلهم، فإنه يتم تزويده بمفتاح المجموعة الجديدة **G4** مشفراً باستخدام مفتاحه الرئيسي ومن ثم نعدل معرف المجموعة (**groupId**) التي ينتمي إليها المستخدم إلى المجموعة **G4** فتصبح لائحة المستخدمين كما في الجدول الآتي:

User Id	Group id
u_1	G_1
u_2	G_1
u_3	G_4
u_{40}	G_4
u_{15}	G_2
u_{10}	G_2
u_{90}	G_2
u_{37}	G_2
u_{22}	G_2
u_4	G_3
u_5	G_3
u_{30}	G_3

جدول 14. لائحة المستخدمين بعد التحديث

يتطلب التحديث في حالة التقسيم إلى **n** عملية تشفير، و **n** عملية إرسال أحادية البث، حيث **n** هو عدد العناصر في المجموعة المراد تقسيمها. الرسائل اللازمة لعملية التحديث الناتج عن تقسيم مجموعة، موضحة بالجدول الآتي:

message	message	encryptions	type	sends
1	$E(K_m , KG_1') \rightarrow$ G1	n-m	unicast	n-m
2	$E(K_m , KG_4) \rightarrow$ G2	m	unicast	m

جدول 15. احصائيات الارسال والتشفير

وفق هذه العملية نضمن تحديث المفاتيح بحيث يستلم كل مستخدم باق في المجموعة **G1** فقط مفتاحاً جديداً للمجموعة **G1**، ويستلم كل مستخدم منقول إلى المجموعة **G4**

مفتاح المجموعة **G4** الجديد، وبهذا الشكل نضمن السرية اللاحقة للبيانات المتبادلة لاحقاً ضمن المجموعتين **G1,G4** بعد الانقسام. مع العلم أن المستخدمين في المجموعتين **G1,G4** يمكنهم الاطلاع على البيانات التي تم تبادلها في المجموعة الأصلية **G1** قبل انقسامها وهذا منطقي.

4-5- إجراء التحديث الدوري للمفاتيح

في نظم الاتصالات طرف-إلى-طرف، يؤدي التغيير الدوري لمفتاح الجلسة (**session key**) إلى زيادة مستوى الأمان، على افتراض وجود متنصت يقوم باقتناص الرسائل المشفرة وتحليلها لاستنتاج مفتاح التشفير، فإن التحديث الدوري لمفتاح التشفير يقلل من النصوص المشفرة بهذا المفتاح، التي يمكن للمتنصت تحليلها **[1,2,27,28]** ، وبالتالي يلزم تحديث مفاتيح المجموعات المستخدمة لتشفير البيانات عند انتهاء مدة صلاحيتها. نفرض أن مركز توزيع المفاتيح يحفظ الفترة الزمنية لآخر تحديث من أجل كل مفتاح بمجموعة وفق اللائحة الآتية:

Group id	Parent id	Group key	Update time	expired
G1	Null	KG1	t1	true
G4	G1	KG5	t2	false

جدول 16. فترات التحديث

لتحديث مفتاح أي مجموعة نقترح إجراء يتلخص بما يلي:

عند انتهاء فترة صلاحية مفتاح المجموعة سواء كانت هذه المجموعات بنفس مستوى السرية (نفس مدة الصلاحية للمفتاح)، أو بمستوى سرية متفاوت، يقوم مركز توزيع المفاتيح بالعمل التالي:

من أجل كل مجموعة في لائحة المجموعات، يتم اختبار مدة صلاحية المفتاح. في حال انتهاء مدة الصلاحية يتم توليد مفتاح جديد للمجموعة ومن ثم تشفيره بمفتاحها القديم وإرساله برسالة بث متعدد إلى جميع عناصر المجموعة. في حالة تقاطع مجموعات أي اشتراك المستخدم بأكثر من مجموعة بث متعدد فان هذا المستخدم

سيستلم أكثر من رسالة تحديث، وذلك نتيجة للحلقة التكرارية التي تمر على كل المجموعات في النظام. يتم التحديث بعملية تشفير واحدة وعملية إرسال متعدد واحدة لكل مجموعة.

```
periodicSystemUpdate()
{
    foreach (Group G in GroupList) {
        if (G.expired){
            KG'=generateNewKey()
             message= E(KG , KG')
             multicast : KDC ---message---> G
        }
    }
}
```

5- الخاتمة

قدمنا في هذه المقالة، نهجا مطورا لعمل مركز توزيع المفاتيح، من خلال تطوير مهام مركز توزيع المفاتيح، المتمثلة في إنشاء الاتصال الثنائي الآمن بواسطة مفتاح الجلسة (**session key**)، وكذلك تمثيل المستخدمين ضمن مجموعات بث متعدد، في نظام يديره مركز توزيع المفاتيح **KDC**، بغية تأمين السرية في اتصالات البث المتعدد، من خلال التشفير بمفتاح المجموعات، بحيث يراعي النهج المقترح مبدأ تكوين الاتصالات الثنائية بالمفتاح المشترك ويطوره. علاوة على ذلك، اقترحنا إجراءات لإضافة وإزالة المستخدمين من مجموعات اتصال البث المتعدد. وإجراءات لدمج وتقسيم مجموعات الاتصال. أخيرا، فيما يتعلق بالمهمة الثالثة من مهام مركز توزيع المفاتيح المتمثلة بتحديث مفاتيح الجلسات، اقترحنا إجراء للتحديث الدوري لجميع المفاتيح اللازمة ضمن النهج المقترح. وضحنا في كل إجراء قمنا به في هذه المقالة عدد عمليات التشفير والإرسال اللازمة لإجراء التحديث وناقشنا توفير عامل السرية السابقة واللاحقة.

المراجع

[1] W.Stallings., **2011 - CRYPTOGRAPHY AND NETWORK SECURITY PRINCIPLES AND PRACTICE.**

[2] J.Menezes and C.van and A.Vanstone., June **1996 - HANDBOOK of APPLIED CRYPTOGRAPHY**, Massachusetts Institute of Technology.

[3] S.Kruus and P.Macker., **1998 - TECHNIQUES AND ISSUES IN MULTICAST SECURITY**, Naval Research Labortory.

[4] D.Wallner, Er.Harder, and R.Agee., July **1999 - Key Management for Multicast: Issues and Architecturest**, National Security Agency.

[5] H.Harney and C.Muckenhirn., **1997 - Group Key Management Protocol (GKMP) Architecture**, **RFC 2094**.

[6] H.Harney and C.Muckenhirn., **1997 - Group Key Management Protocol (GKMP) Specification**. **RFC 2093**.

[7] Y.Wang., **2008 - Key Management for Secure Group Applications in Wireless Networks**.

[8] S.Banerjee, and B.Bhattacharjee., **2002 - Scalable secure group communication over IP multicast**, IEEE Journal on Selected Areas in Communications.

[9] M. Steiner, G. Tsudik, and M. Waidner., **1997 - Diffie-Hellman Key Distribution Extended to Group Communication,Proc**. Third ACM Conf. Computer and Comm. Security.

[10] W.Die and M.Hellman., **1976 - New Directions In Cryptography,IEEE Transactions on Information Theory**.

[11] Y.Kim , A.Perrig, , and G.Tsudik., **2004 - Group Key Agreement Efficient in Communication** , IEEE Transactions on Computers.

[12] A.Bakkardie., **1996 - Scalable multicast key distribution**, **RFC 1949**.

[13] A.Sherman, and D.McGrew., **2003 - Key Establishment in Large Dynamic Groups Using One-Way Function Trees**. IEEE Transactions on Software Engineering.

[14] R.Molva, and A.Pannetrat., **1999 - Scalable multicast security in dynamic groups**, **6th** ACM Conference on Computer and Communications Security ACMCCS Singapore.

[15] S.Mittra., **1997 - Iolus: A framework for scalable secure multicasting**, ACM SIGCOMM.

[16] T.Hardjono, B.Cain and I.Monga., **2000 - Intra-Domain Grouop Key Management Protocol**, IETF draft.

[17] A.Perrig, D.Song, and J.Tygar., **2001 - ELK,a New Protocol for Efficient Large-Group Key Distribution**. In Proceedings of the IEEE Symposium on Security and Privacy.

[18] C.Wong, M.Gouda and S.Lam., **2000 - Secure Group Communications Using Key Graphs**.

[19] Y.Challal and H.Seba., **2008 - Group Key Management Protocols: A Novel Taxonomy**.

[20] M.Stamp., **2011 - INFORMATION SECURITY Principles and Practice** .Second Edition, John Wiley & Sons, Inc., Hoboken, New Jersey.

[21] R.Dondeti., **2000 - A Distributed Group Key Management Scheme for Secure Many-to-many Communication**.

[22] B.DeCleene, L.Dondeti and S.Griffin., **2001 - SECURE GROUP COMMUNICATIONS FOR WIRELESS NETWORKS**, IEEE **0-7803-7225-5/01**.

[23] C. E. Shannon., **1948 - Communication Theory of Secrecy Systems**, Bell Systems Technical Journal,Vol**. 28, pp. 656–715**.

[24] A.Ballardie., May **1996 - Scalable Multicast Key Distribution**, University College London.

[25] T.Harddjono and GTsudik., **2000 - IP Multicast Security:Issu and Directions**.

[26] s.Deering., **1989 - Host extension for IP multicast**, **RFC 1112**,IETF.

[27] P.Karn and W.Simpson., March **1999 - Session-Key Management Protocol**,**RFC 2522**.

[28] S.Donovan and J.Rosenberg., **2005 - Session Timers in the Session Initiation Protocol (SIP)**, **RFC 4028**