التحديث الدوري لمفاتيح التشفير المستخدمة في الاتصالات الآمنة بين المجموعات في نظام مركز توزيع المفاتيح

*أ.د. أحمد الكردي،** محمد الرسلان

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

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

ملخص

في نظم الاتصالات الثنائية (طرف–إلى–طرف)، التي يديرها مركز توزيع المفاتيح، يؤدي التغيير الدوري لمفتاح الجلسة إلى زيادة مستوى الأمان، لأنه يقلل من فرص تحليل الرسائل المتبادلة من قبل أي متنصت. يتم ذلك بتحديث مفتاح الجلسة كل فترة زمنية محددة، تبعا لمستوى سرية المفتاح.

في نظم الاتصالات الجماعية، يتم تشفير اتصالات المجموعات باستخدام مفاتيح المجموعات، وبالتالي نحتاج إلى عملية تحديث دوري في مركز توزيع المفاتيح تتناسب مع هذه الأنظمة.

نسعى من خلال هذا البحث إلى اقتراح أسلوب لتمثيل الاتصالات الآمنة بين المجموعات في نظام مركز توزيع المفاتيح، إضافة لإجراءات عوديه لعملية التحديث الدوري لمفاتيح التشفير تتوافق مع هذا الأسلوب. نبين في كل إجراء عدد عمليات التشفير وعدد الرسائل اللازمة لإجراء التحديث.

الكلمات المفتاحية: التشفير، اتصالات المجموعات، البث المتعدد، البث الأحادي، التحديث الدوري، تأمين الاتصال

1- مقدمة:

في السنوات الأخيرة، انتشرت التطبيقات والبروتوكولات التعاونية والمعتمدة على المجموعات بشكل واسع [1,3,4,5,7,24,25]. تشمل هذه التطبيقات عادةً الاتصالات ضمن شبكة مفتوحة، لذلك يعتبر أمن الاتصال في هذه التطبيقات متطلباً مهماً [11,12]. لأجل ذلك ناقشت العديد من الأبحاث أساليب وطرائق تحقيق اتصالٍ آمنٍ بين المجموعات. طرح [9,10] نهجاً لتحقيق الاتصال الجماعي باستخدام خوارزمية **DH** (**Diffie-Hellman**) 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 **GDH** (**Group Diffie-Hellman**) وتم تطوير هذا البروتوكول على عدة مراحل، نظراً للكلفة الحسابية للنسخة الأولى. وتوالت الدراسات فقدم [5,6] نهجاً لإدارة المفاتيح اللازمة لتأمين اتصالات مجموعة باستخدام بروتوكول **GKMP** (**Group Key Management Protocol**). أما في [11]، فتم تطوير أسلوب لتحقيق الاتصال الآمن بين المجموعات، يعتمد على تطبيق خوارزمية **DH** وفق مبدأ الأشجار، ببروتوكول **TGDH** (**Tree Based Group DH**). اعتمدت أبحاث أخرى على مبدأ هرمي، بالتقسيم إلى تجمعات (**Clusters**)، كما هو الحال في [8]. وهناك أبحاث ركزت في محتواها على نظام غير مركزي لإدارة المفاتيح كما في [15]، فقد تم استخدام نظام إدارة متعدد المستويات، تُقسم فيه المجموعة إلى عدة مجموعات فرعية، يمكن تمثيلها بواسطة الأشجار، لكل مجموعة فرعية متحكم يدعى (**Group Security Initiator**) **GSI**، وتدار المجموعة الأساسية في المستوى الأعلى من خلال متحكم (**Group Security Controller**) **GSC**، حيث يكون متحكم المجموعة الفرعية مسؤولاً عن تأمين سرية الاتصالات التي تتم بين المستخدمين ضمن المجموعة الفرعية، ومع المجموعات الفرعية الأخرى. وفي [16] تم استعراض نهج لإدارة المفاتيح للمجموعات يعتمد 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،

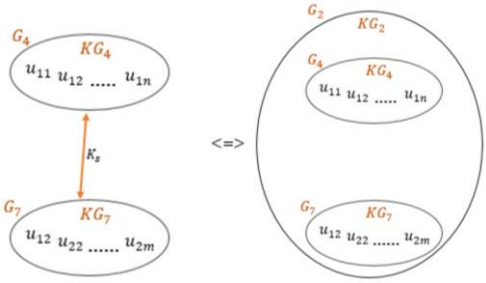
ومتحكم خاص بكل منطقة. كذلك الأمر اعتمد [17,18,22] على بنية شجرية عملية إدارة المفاتيح المستخدمة لتشفير البيانات المتبادلة بين المجموعات في بروتوكول **Logical Key Hierarchy (LKH)** وبروتوكول **Large-Group Key Distribution (ELK)**، وكذلك الأمر في بروتوكول **One-way Function Tree (OFT)** في [13].

في النظم التي يديرها مركز توزيع المفاتيح **(KDC)** [1,2,19,20]، يشترك المركز مع كل مستخدم في النظام بمفتاحٍ رئيسي K_m **(master key)**، ويمكن لكل زوج من المستخدمين تأسيس اتصال ثنائي **(end-to-end)** آمن بينهما، حيث يكون **KDC** مسؤولاً عن تأسيس جلسة موثوقة بين المستخدمين من خلال توزيع مفتاح مشترك **(shared secret key)** لكل منهما مشفرٍ بمفتاحه الرئيسي ويدعى هذا المفتاح بمفتاح الجلسة K_S **(session key)**. يُستخدم مفتاح الجلسة لتشفير البيانات المرسلة من الطرف الأول، ليتم إرسالها ضمن شبكة عامة، وفك تشفيرها عند الطرف الآخر(المستقبل). لا يقتصر عمل مركز توزيع المفاتيح على تأسيس الجلسة فقط، بل يكون مسؤولاً أيضا عن التحديث الدوري لمفاتيح الجلسات الثنائية الآمنة، التي تم تأسيسها بين المستخدمين في النظام [1,21,27].

في هذه المقالة، نقدم أسلوباً لتمثيل الاتصالات الآمنة بين المجموعات في نظام مركز توزيع المفاتيح، وفق بنية الأشجار، إضافة لإجراءاتٍ عودية للتحديث الدوري لمفاتيح التشفير المستخدمة في الاتصالات بين المجموعات، تتلاءم مع هذا الأسلوب.

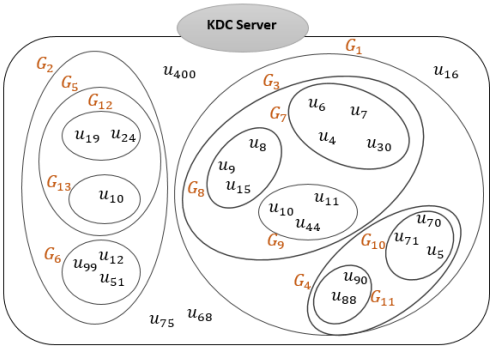
2- تمثيل الاتصال الآمن بين المجموعات

يمكن أن نعتبر أن الاتصال الآمن بين مجموعتين يتم بتكوين مفتاح مشترك بين المجموعتين [17,18]، (انظر الشكل **1**). كما هو الحال عند تكوين جلسة آمنة بين مستخدمين. هذا يكافئ منطقياً تكوين مجموعة، بحيث يكون عناصرها المجموعتان الراغبتان بالاتصال الآمن، ومفتاح هذه المجموعة هو المفتاح المشترك، الذي يستخدم لتشفير الاتصال بين المجموعتين.



الشكل 1. تأمين الاتصال بين مجموعتين.

إن المجموعة الناتجة تعتبر مجموعة حاوية للمجموعتين، ويمكن توسيع هذه المجموعة لتضم عدداً أكبر من مجموعات الإرسال المتعدد، بتزويد هذه المجموعات بمفتاح المجموعة الحاوية. لتحقيق أكثر من مستوى لاتصالات المجموعات يتم تمثيل المجموعات وفق علاقات احتواء (انظر الشكل **2**)، بحيث نجعل الخادم **KDC** يدير عدداً من المجموعات والمجموعات الفرعية، المحتواة ضمنها، وعلى عدة مستويات.



الشكل 2. أسلوب تمثيل الاتصال الآمن بين المجموعات في مركز توزيع المفاتيح.

يستعمل المستخدمون ضمن المجموعة **G9**، المفتاح KG_9 ، لتشفير البيانات ضمن المجموعة **G9**، وكذلك الأمر بالنسبة للمستخدمين ضمن المجموعات **G8, G7, G12, G13, G6, G10, G11**.

من أجل تشفير البيانات الموجهة لكلا المجموعتين **G11** و **G10**، يتم استخدام مفتاح المجموعة الحاوية للمجموعتين، أي المفتاح KG_4. يتم أيضاً استخدام مفتاح المجموعة **G3** من أجل الاتصال الآمن بين المجموعات G8,G9,G7. كما يُستخدم مفتاح المجموعة **G2** للاتصال الآمن بين المجموعات **G5,G6** وهكذا ...

يحفظ مركز توزيع المفاتيح لائحة بالمجموعات المشكلة في النظام (**GroupList**)، تتضمن هذه اللائحة معرف هذه المجموعة، ومفتاحها السري، ومعرف المجموعة التي تحوي هذه المجموعة (هو ما سميناه بأب المجموعة (**parent**)) لذلك نعتبر أن **null** هو الأب بالنسبة للمجموعات الكلية (غير محتواه في أي مجموعة). يوضح الجدول الآتي لائحة المجموعات:

Group id	Parent id	Group key
G_1	Null	KG_1
G_2	Null	KG_2
G_3	G_1	KG_3
G_4	G_1	KG_4
G_5	G_2	KG_5
G_6	G_2	KG_6
G_7	G_3	KG_7
G_8	G_3	KG_8
G_9	G_3	KG_9
G_{10}	G_4	KG_{10}
G_{11}	G_4	KG_{11}
G_{12}	G_5	KG_{12}
G_{13}	G_5	KG_{13}

جدول 1. لائحة المجموعات المشكلة في النظام

أما لائحة المستخدمين (**UserList**)، فإنها تتضمن معرف المستخدم، ومفتاحه الرئيسي المشترك مع مركز توزيع المفاتيح:

User Id	Master Key
u_9	K_{m9}
u_8	K_{m8}
u_4	K_{m4}
u_{24}	K_{m24}
u_{68}	K_{m68}
u_{400}	K_{m400}
.....	.....

جدول 2 . لائحة المستخدمين

أما بالنسبة للائحة الاتصالات الجماعية **(UserGroupList)**، فإنها تتضمن معرف المستخدم، ومعرف المجموعة التي ينتمي إليها. يتم تسجيل المستخدم في اللائحة، وذلك ضمن مجموعة البث المتعدد التي ينتمي إليها، أي المجموعة التي ليس لها أبناء (ليس لها مجموعات محتواة ضمنها)، ذلك لتحقيق أنواع الاتصالات الجماعية الآمنة. بالنسبة للمستخدمين غير المنضمين لأية مجموعه، يتم تسجيلهم ضمن المجموعة **null**، ولا يمكن لهؤلاء المستخدمين تشفير البيانات عبر الاتصالات الجماعية كونهم لا ينتمون لأية مجموعة. يوضح الجدول الآتي لائحة الاتصالات الجماعية:

User Id	Group id
u_9	G_8
u_8	G_8
u_4	G_7
u_{24}	G_{12}
u_{68}	$null$
u_{400}	$null$
.....	.....

جدول 3 . لائحة الاتصالات الجماعية

يحتفظ كل مستخدم بمفتاحه الرئيسي المشترك مع الخادم **KDC**، ومفتاح المجموعة الكلية والمجموعات الفرعية الموجود فيها، حيث يستخدم مفتاح كل مجموعة لتشفير البيانات المراد توجيهها للعناصر (مستخدمين –مجموعات فرعية) ضمن نطاق هذه المجموعة.

فمثلا في الشكل **2**، المستخدم u_6 يحتفظ بالمفاتيح $\boldsymbol{KG_1, KG_3, KG_7, K_{m6}}$ فيستخدم المفتاح $\boldsymbol{KG_7}$ لتشفير البيانات المتبادلة مع نظرائه من المستخدمين الموجودين في المجموعة الفرعية $\boldsymbol{G_7}$ وهم $\boldsymbol{u_4, u_7, u_{30}}$ ويستخدم المفتاح $\boldsymbol{KG_3}$ لتشفير البيانات التي يريد توجيهها إلى كل المستخدمين والمجموعات الفرعية ضمن المجموعة الكلية $\boldsymbol{G_3}$.

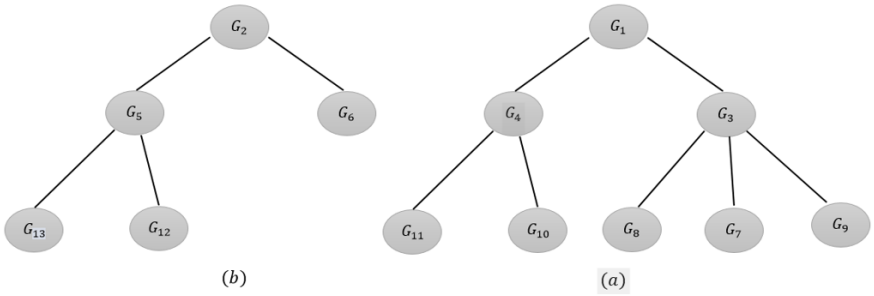
المستخدم $\boldsymbol{u_{70}}$ يحفظ المفاتيح $\boldsymbol{KG_1, KG_4, KG_{10}, K_{m70}}$. يستخدم المفتاح $\boldsymbol{KG_{10}}$ لتشفير البيانات مع نظرائه من المستخدمين الموجودين في المجموعة الفرعية $\boldsymbol{G_{10}}$، ويستخدم المفتاح $\boldsymbol{KG_4}$ لتشفير البيانات التي يريد توجيهها إلى كل المستخدمين والمجموعات الفرعية ضمن المجموعة الكلية $\boldsymbol{G_4}$

إن عملية إدارة توزيع المفاتيح هي المرحلة الأكثر تعقيداً، بحيث يعمل المركز **KDC** بخوارزمية تضمن توزيع وتحديث المفاتيح بشكل آمن وسليم، بحيث لا يسمح لأي مستخدم الحصول على مفتاح مجموعة لا ينتمي إليها، ومطلوب ألا تؤدي هذه الخوارزمية إلى إغراق النظام بعدد كبير من رسائل توزيع وتحديث المفاتيح، الأمر الذي يتسبب في هدر موارد النظام في علميات التشفير والإرسال.

في نظام **KDC** الذي يدير مجموعات كلية ومجموعات فرعية، يمكن تمثيل المجموعات ضمن النظام على أن كل مجموعة كلية هي عبارة عن شجرة، الأب الأساسي هو المجموعة الكلية نفسها، والأبناء هم المجموعات الفرعية المحتواة ضمنها، وهكذا حتى الوصول إلى المجموعات في المستوى الأدنى، التي تمثل الأوراق في الشجرة.

بالعودة إلى الشكل **2**، يمكن رسم أشجار المجموعات ضمن نظام **KDC** وفق الشكل **3**.

تمثل كل شجرة تطبيقاً منفرداً، وبالتالي يمكن لمركز **KDC** إدارة السرية في الاتصالات بين المجموعات لأكثر من تطبيق، مع استخدام لوائح مستخدمين ومجموعات واحدة لكل النظام.



الشكل 3. أسلوب تمثيل الاتصال الآمن بين المجموعات وفق بنية الأشجار.

3- التحديث الدوري

في نظم الاتصالات الثنائية (**one-to-one**)، التي يديرها مركز توزيع المفاتيح، يؤدي التغيير الدوري لمفتاح الجلسة [**1,2**] 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 [**21,27**]. يؤدي التحديث المتكرر لمفاتيح التشفير إلى تأخير تبادل البيانات، وإلى استهلاك أكبر في موارد الخادم، نظراً لعمليات التشفير والإرسال اللازمة للتحديث. في الأنظمة المعتمدة على الاتصالات الآمنة بين المجموعات تكون اتصالات البث المتعدد ضمن المجموعة والاتصالات بين المجموعات مشفرة باستخدام مفاتيح المجموعات، بدلًا من مفتاح الجلسة المشترك بين كل زوج من المستخدمين، وبالتالي يجب تحديث مفاتيح هذه المجموعات بشكل دوري لضمان أمن عالٍ في النظام [**28**]، إضافة لذلك قد لا تكون كل المجموعات بالمستوى نفسه من السرية، فقد يكون لمجموعة مستوى أمان، مطلوب توفيره، أعلى من بقية المجموعات، وبالتالي فإن الفترة الدورية لتحديث مفتاح هذه المجموعة ستكون أقل من بقية المجموعات في شجرة التطبيق. نسعى من خلال هذا البحث إلى اقتراح

إجراءات مناسبة لتحديث مفاتيح المجموعات، بحيث يكون الإجراء بشكل أساسي موثوقاً لا يؤدي لإرسال أي مفتاح تشفير إلى مستخدمين غير مخولين بالحصول عليه للحفاظ على أمن النظام كاملاً.

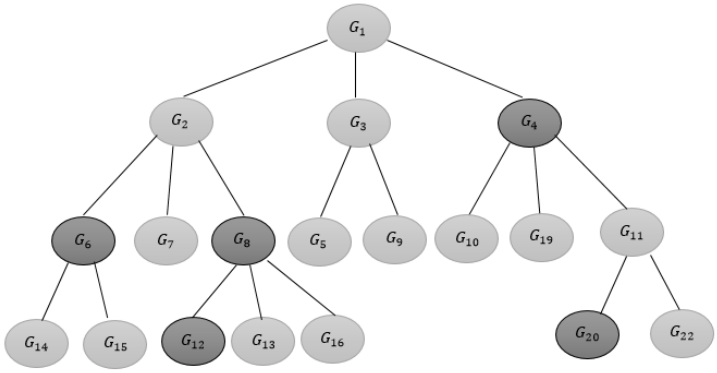
نفرض أن مركز توزيع المفاتيح يحفظ الفترة الزمنية لآخر تحديث لكل مفتاح خاص بمجموعة وفق اللائحة:

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

جدول 4 . لائحة المجموعات وتحديثاتها

على اعتبار أن المجموعات تمثل شجرة (الشكل **3**)، بالتالي نعتمد على العبور على الأشجار بحيث يتم تكوين رسائل تحديث سليمة، وتوزيعها بحيث لا يتطلب التحديث عمليات تشفير وإرسال كبيرة، تؤدي إلى استهلاك موارد النظام وحزمة الاتصال.

يشير الشكل 4 إلى شجرة مجموعات لتطبيق ما، يديره مركز توزيع المفاتيح. إن فترة الصلاحية لمفاتيح المجموعات **G4,G6,G8,G12,G20** انتهت، وتتطلب تحديثاً لمفاتيحها. فالمستخدمون في المجموعة **G14**، مثلاً، يجب أن يستلموا مفتاحاً محدثاً للمجموعة **G6**، والمستخدمون في المجموعة **G16** يجب أن يستلموا مفتاحاً محدثاً واحداً للمجموعة **G8**، والمستخدمون في المجموعة **G12** يجب أن يستلموا مفاتيح محدثة للمجموعات **G8,G12**، وكذلك المستخدمون في المجموعة **G19,G10**، يجب أن يستلموا نسخة جديدة لمفتاح المجموعة **G4**، والمستخدمون في المجموعة **G20** يجب أن يستلموا نسخاً جديدة لمفاتيح المجموعتين **G4,G20** وهكذا...



الشكل 4. شجرة مجموعات التطبيق

لأجل ذلك، نعرف بنية تسمى المجموعة المتأثرة بالتحديث تتضمن معرف المجموعة ومفتاحها القديم (**old key**) ومفتاحها الجديد أو المُحدث (**new key**):

Group id	Old key	New key

جدول 5 . بنية معلومات مجموعة

لكل مجموعة متأثرة بالتحديث، والذي يعني أن مفتاح هذه المجموعة ســـيتعرض للتحديث نتيجة لانتهاء مدة صلاحيته، يتم توليد مفتاح جديد (**new key**).

3-1- إجراء 1 (التحديث عند الانتهاء)

في هذا الإجراء نقوم بتحديث مفتاح أية مجموعة بمجرد انتهاء فترة صلاحيته، سواءً كانت هذه المجموعات بنفس مستوى السرية (نفس مدة الصلاحية للمفتاح)، أو بمستوى سرية متفاوت، فيتم اختبار مدة الصلاحية لكل المجموعات في النظام، المسجلة في لائحة المجموعات، وبمجرد تحقق شرط انتهاء الصلاحية لمفتاح المجموعة المختبرة، يتم تحديث مفتاحها، بتوليد مفتاح جديد، وتشفيره بالمفتاح القديم للمجموعة، ومن ثم إرساله برسالة بث متعدد إلى جميع عناصر المجموعة:

```
periodicUpdate()
{
    foreach (Group G in GroupList) {
        if (G.expired){
            KG′=generatNewKey()
            message= E(KG , KG′)
            multicast : KDC ──message──→ G
        }
    }
}
```

بالعودة إلى الشكل **4**، نجد أنه وفق الحلقة التكرارية في الإجراء السابق سيتم تحديث المفاتيح لخمس مجموعات.

يبين الجدول التالي عدد عمليات التشفير والإرسال اللازمة لعملية التحديث وفق الإجراء **1**:

message	message → Group	encryptions	type	sends
1	E (KG_{20} ,KG'_{20}) → G20	1	multicast	1
2	E (KG_4 ,KG'_4) → G4	1	multicast	4
3	E (KG_8 ,KG'_8) → G8	1	multicast	3
4	E (KG_{12} ,KG'_{12}) → G12	1	multicast	1
3	E (KG_6 ,KG'_6) → G6	1	multicast	2

جدول 6 . احصائيات الارسال والتشفير

<u>مناقشة الميزات:</u>

يتناسب هذا الإجراء مع مختلف الحالات، ولكن قد ينتج عنه أسلوب غير منتظم لرسائل التحديث. في الشكل **4**، نجد أن مفتاحي المجموعتين **G4,G20** منتهيا الصلاحية،

ونتيجة لإجراء التحديث سيتم تحديث مفتاح المجموعة **G20** عبر تشكيل رسالة تحديث، بتوليد مفتاح جديد KG'_{20}، وتشفيره بالمفتاح القديم للمجموعة KG_{20}، وإرسالها ببث متعدد إلى جميع المستخدمين في المجموعة **G20**. من جهة أخرى نجد أن مفتاح المجموعة **G4** أيضا يتطلب تحديثاً، وعند وصول التكرار في الإجراء إلى المجموعة **G4**، سيقوم المركز **KDC** بتحديث مفتاحها أيضا برسالة بث متعدد إلى المجموعات الأوراق أسفل المجموعة **G4**. المستخدمون ضمن المجموعة **G20** يستلمون رسالة التحديث هذه، وبالتالي يستلم عناصر المجموعة **G20** أكثر من تحديث متتالٍ.

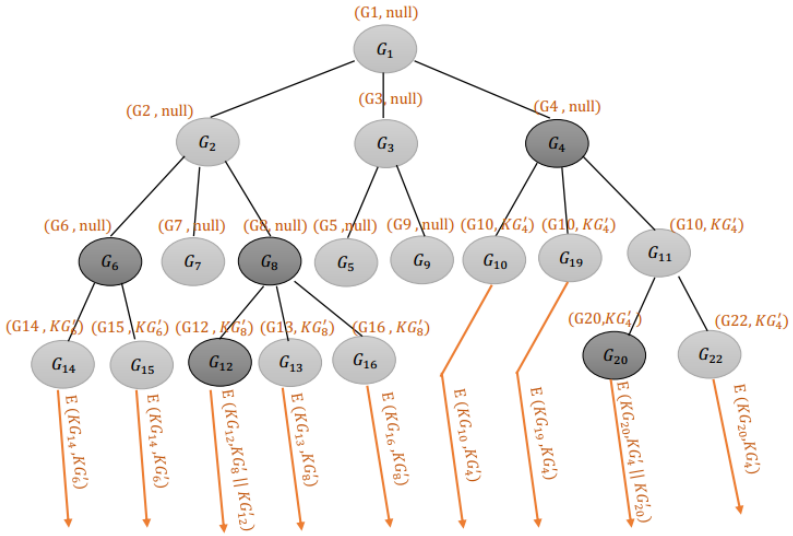
3-2- الإجراء 2 (الدمج التعاودي من الأعلى إلى الأسفل)

في هذا الإجراء، نسعى لحل مشكلة الأسلوب غير المنتظم لرسائل التحديث في الإجراء **1** المقترح آنفاً، وذلك بتجميع رسائل التحديث على كل مسار، ومن ثم إرسالها برسالة واحدة، بحيث لا يصل لأي مجموعة أكثر من رسالة تحديث متتالية. نعرف لائحة المفاتيح المحدثة (**updatedKeys**)، نستخدمها لتجميع المفاتيح المحدثة الواقعة على نفس المسار من الجذر للمجموعة الورقة. عندما يصل التمرير إلى المجموعة الورقة، يتم تشكيل رسالة تحديث تتضمن لائحة المفاتيح المحدثة، مشفرةً بمفتاح المجموعة الورقة، وإرسالها إلى المجموعة الورقة برسالة واحدة بث متعدد.

```
periodicSystemUpdate()
{
      updatedKeys = null     //list of keys
      foreach (Group G in GroupList where G.parent=null) {
          if (G.expired)
              periodicUpdate(G, updatedKyes)
      }
}
periodicUpdate(affectedGroup g, keys updatedKyes)
{
      if (g.expired){
          Kg' =generatNewKey()
          updatedKeys = updatedKeys || Kg'
```

```
    }
    childs = getChilds(g)                          //get childs of node
g
    if (childs = null) {                           //stop condition
          if(updatedKyes ≠ null){                  //
               message = E(Kg, updatedKeys)
               multicast : KDC  ──message──→  g
          }
    }
  else{
       foreach group C in childs
          periodicUpdate(C , updatedKeys)
  }
}
```

يوضح الشكل **5** التمرير التعاودي للإجراء **2** من أجل المجموعة الجذر **G1**.



الشكل 5 . التحديث الدوري باستخدام الدمج التعاودي من الأعلى للاسفل.

يبين الجدول الآتي عدد عمليات التشفير والإرسال اللازمة لإجراء التحديث وفق الإجراء **2**:

message	message → Group	encryptions	type	sends

1	E (KG_{14} ,KG'_6) → G14	1	multicast	1
2	E (KG_{15} ,KG'_6) → G15	1	multicast	1
3	E (KG_{12},KG'_8 || KG'_{12}) → G12	1	multicast	1
4	E (KG_{13} ,KG'_8) → G13	1	multicast	1
5	E (KG_{16} ,KG'_8) → G16	1	multicast	1
6	E (KG_{10} ,KG'_4) → G10	1	multicast	1
7	E (KG_{19} ,KG'_4) → G19	1	multicast	1
8	E (KG_{20},KG'_4 || KG'_{20}) → G20	1	multicast	1
9	E (KG_{22},KG'_4) → G22	1	multicast	1

جدول 7 . احصائيات الارسال والتشفير

- ✓ عدد عمليات التشفير = **9**
- ✓ عدد عمليات الإرسال =**9**

مناقشة الميزات:

- يوفر هذا الإجراء تجميع مفاتيح المجموعات المطلوب تحديثها في لائحة، بحيث نصل إلى المجموعات الأوراق في الشجرة، وقد تم إضافة كل المفاتيح القابلة للتحديث، والواقعة على المسار من العقدة الجذر إلى العقدة الورقة، إلى هذه اللائحة ليتم تشفيرها بمفتاح العقدة الورقة، وإرسالها إلى جميع عناصرها، برسالة واحدة بث متعدد.
- يحل هذا الإجراء مشكلة عدم الانتظام في إرسال رسائل التحديث المتوقع حدوثها بالإجراء **1**، كما يضمن وصول المفاتيح المحددة للمستخدمين في المجموعات بطريقة آمنة.

- يتجنب هذا الإجراء إرسال رسائل إلى عقد (مجموعات) أوراق في حال لم يكن أي من آباء هذه العقد يتطلب تحديثاً لمفتاحه.
- من مساوئ هذا الإجراء أن عدد عمليات التشفير والإرسال مرتبط بعدد العقد الأوراق، التي يتضمن المسار من الجذر إليها مفاتيح محدثة، ولا يرتبط بعدد المجموعات التي تطلب تحديث مفاتيحها، وهذا يعني عدد عمليات تشفير أكبر من عدد المفاتيح المطلوب تحديثها.

3-3- الإجراء 3: (التشفير والدمج التعاودي من الأسفل إلى الأعلى)

في هذا الإجراء، يقوم الخادم بجمع المفاتيح المحدثة لكل العقد الأبناء، ذلك عن طريق العبور على الشجرة بدءاً من العقد الأوراق. حيث يتم إعادة مقطع رسالة من كل عقدة ابن، يتضمن هذا المقطع مفتاح هذه العقدة (المجموعة) التي انتهت فترة صلاحية مفتاحها، مشفراً بمفتاحها القديم، إلى أب هذه العقدة. يقوم هذا الأب بدمج المقاطع المعادة من أبنائه مع المقطع الذي يولده هو، إن تطلب تحديثا لمفتاحه، ثم يعيدها إلى أبيه، وهكذا حتى الوصول إلى الجذر. بذلك يتكون لدينا رسالة ناتجة عن دمج عدة مقاطع مشفرة، مقطع أو رسالة لكل مجموعة عقدة انتهت فترة صلاحية مفتاحها. بالنهاية يقوم الخادم بتوجيه هذه الرسالة إلى المجموعة الجذر، التي تمثل المجموعة الكلية، أي يتم ارسالها برسالة بث متعدد إلى كل مجموعة ورقة، وبالتالي إلى كل المستخدمين في التطبيق، حيث يستلم كل مستخدم هذه الرسالة. لن يتمكن المستخدم من فك تشفير كل مقاطع هذه الرسالة، بل سيستخدم المفاتيح المتوفرة والمحفوظة لديه لفك تشفير أجزاء (مقاطع) الرسالة المهمة له، والتي تتضمن فقط المفاتيح الجديدة للمجموعات الواقعة على المسار من الجذر إلى المجموعة الورقة التي ينتمي إليها المستخدم، وبالتالي يحصل على المفاتيح المخول بالحصول عليها فقط، وباقي أجزاء الرسالة لن يتمكن من فك تشفيرها والحصول على المفاتيح المحدثة التي لا تهمه، كونه لا يمتلك المفتاح الذي تم استخدامه لتشفير رسالة التحديث.

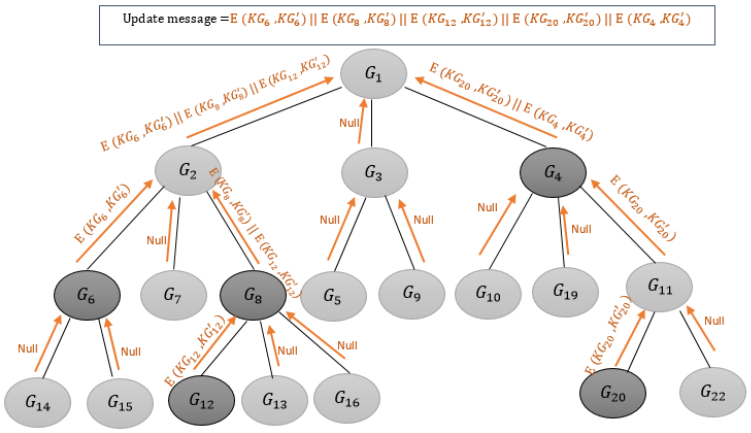
```
periodicSystemUpdate()
{
    Group G
```

```
      updateMessage = null
      foreach (Group G in GroupList where G.parent=null) {
            updateMessage =periodicUpdate(G)
     if(message1 ≠ null)
            multicast : KDC ──updateMessage──▶ G
     }
}
string periodicUpdate(Group g)
{
     message =null
     if (g.expired){
        Kg′=generatNewKey()
        message = E(Kg , Kg′)
     }
     childs = getChilds(g)   //get childs of node g
     if (childs = null) {        //stop condition
        return message
     }
    else{
       foreach (Group C in childs)
          message = message || periodicUpdate(C )
    }
}
```

يوضح الشكل **6**التمرير التعاودي للإجراء **3**. نجد أن الخادم يقوم بإرسال رسالة التحديث الوحيدة والمكونة من دمج عدة مقاطع إلى جميع المستخدمين في المجموعات. المستخدمون في المجموعة $G14$، مثلاً، يقومون بفك تشفير مقطع من الرسالة الكلية، يتضمن هذا المقطع المفتاح المحدث KG_6'، من خلال فك تشفير هذا المقطع بالمفتاح المحفوظ لديهم KG_6، ولا يهتمون ببقية أجزاء الرسالة. المستخدمون في المجموعة **G20** يقومون بفك تشفير مقاطع من الرسالة للحصول على المفاتيح KG_{20}', KG_4' ولا يهتمون ببقية الأجزاء وهكذا...

نلاحظ أن المستخدمين في المجموعات **G7,G5,G9** يستلمون رسالة التحديث، رغم عدم فائدة الرسالة بالنسبة لهم كون المجموعات الواقعة على المسار من الجذر إلى هذه

المجموعات لا تتطلب تحديث مفاتيحها، وهذا يعتبر أهم مساوئ هذا الإجراء، مع الإشارة أن هؤلاء المستخدمون لن يتمكنوا من فك تشفير أي جزء من الرسالة، كونهم لا يحتفظون بالمفاتيح اللازمة لفك التشفير، وهذا يحافظ على أمن النظام كاملاً.



الشكل 6. التحديث الدوري باستخدام التشفير والدمج التعاودي من الأسفل للاعلى.

يبين الجدول الآتي عدد عمليات التشفير والإرسال اللازمة لإجراء التحديث وفق الإجراء **3**:

essage	Updated Keys → Root	encryptions	type	sends
1	E (KG_6 ,KG'_6) || E (KG_8 ,KG'_8) || E (KG_{12} ,KG'_{12}) || E (KG_{20} ,KG'_{20}) || E (KG_4 ,KG'_4) → G1	5	multicast	11

جدول 8 . احصائيات الارسال والتشفير

- ✓ عدد عمليات التشفير = **5**
- ✓ عدد عمليات الإرسال =**11**

<u>مناقشة الميزات:</u>

- عدد عمليات التشفير في هذا الإجراء مرتبط بعدد المجموعات المطلوب تحديث مفاتيحها، ولا يتعلق بعدد العقد الأوراق كما في الإجراء **2**.

- عدد عمليات الإرسال بعدد المجموعات الأوراق، وبالتالي تصل الرسالة إلى كل المستخدمين في التطبيق، حيث يقوم كل مستخدم بفك تشفير أجزاء من هذه الرسالة، تبعا للمفاتيح التي يحتفظ بها ليحصل على المفاتيح المحدثة.
- من مساوئ هذا الإجراء، وصول رسالة التحديث الكلية إلى مستخدمين غير مهتمين بهذه الرسالة، كون المجموعات الواقعة على المسار من الجذر إلى المجموعات الأوراق التي ينتمي لها هؤلاء المستخدمون لا تتطلب تحديث لمفاتيحها.
- من مساوئ هذا الأسلوب في التحديث، وجود مقاطع من الرسالة لا تهم المستخدم ووجود كل المفاتيح برسالة واحدة (عدة مقاطع مشفرة) قد يعطي لمهاجم فرصة أكبر في الحصول على مفاتيح التشفير.

3-4- الإجراء 4 (التشفير والدمج أسفل الأب الفرعي)

periodicUpdate(Group g , bool flag)				
state	childs	flag	g.expired	Behavior
1	null	false	false	do nothing
2	null	false	true	Kg'=generatNewKey() message= $E(Kg\ ,\ Kg')$ multicast : KDC $\xrightarrow{\text{message}}$ g
3	null	true	false	return null
4	null	true	true	return $E(Kg\ ,\ Kg')$
5	not null	false	false	foreach (Group C in childs) periodicUpdate(C , false)

6	not null	false	true	Kg'=generatNewKey() message= $E(Kg\ ,\ Kg')$ foreach (Group C in childs) message = message ‖ periodicUpdate(C , true) multicast : KDC $\xrightarrow{\text{message}}$ g
7	not null	true	false	foreach (Group C in childs) message = message ‖ periodicUpdate(C , true) return message
8	not null	true	true	Kg'=generatNewKey() message= $E(Kg\ ,\ Kg')$ foreach (Group C in childs) message = message ‖ periodicUpdate(C , true) return message

يمثل هذا الإجراء عبوراً على الشجرة من الجذر باتجاه الأوراق، كما في الإجراء **2**، ولكن بمجرد مصادفة عقدة تتطلب تحديثاً لمفتاحها، يتحول الإجراء إلى إجراء تعاودي من الأسفل عند الأوراق للأعلى، وصولاً إلى هذه العقدة، والتي نطلق عليها مسمى الأب الفرعي، فيتم تجميع رسائل التحديث أسفل هذا الأب الفرعي بنفس الأسلوب المتبع في الإجراء **3**، ومن ثم إرسال رسالة واحدة بث متعدد موجهة إلى مجموعة الأب الفرعي.

نناقش ثلاثة بارامترات في الاجراء وهي:

1- وجود أبناء للعقدة المُمررة بالاستدعاء العودي:

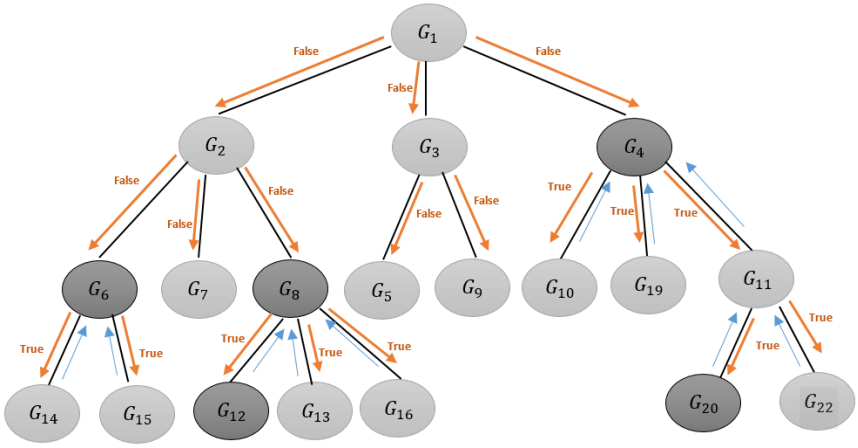
childs=(null , not null)

2- انتهاء مدة صلاحية المفتاح للعقدة المُمررة بالاستدعاء العودي:

g.expired=(true , false)

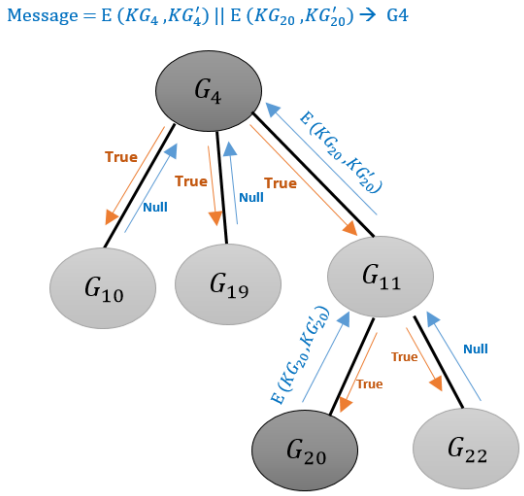
3- عاملا إضافي هو العلم **flag=(true,false)**. فمن أجل كل عقدة، إذا كانت هذه العقدة لا تتطلب تحديث لمفتاحها، تمرر **flag=false** إلى أبنائها، للدلالة على أنها لن تكون أبا فرعيا. عند وصول التمرير التعاودي إلى عقدة، وكانت فترة مفتاحها قد انتهت، فإنها تعتبر نفسها أبا فرعيا، وتمرر **flag=true**، إلى أبنائها لتشير لهم أن رسالة التحديث ستكون موجهة عبرها، عندها تقوم العقد الأبناء بإعادة رسائل التحديث الخاصة بها بحال كانت تتطلب تحديثا لمفاتيحها وذلك بمقاطع تتضمن مفتاحها الجديد مشفراً بمفتاحها القديم، وتعيد هذا المقطع المشفر إلى الأب الذي يدمج المقاطع المعادة من أبنائه، ويعيدها إلى أبيه، وهكذا حتى الوصول للأب الفرعي، أي تعاود الخوارزمية التعاودية دمج رسائل التحديث صعوداً من الأوراق إلى هذا الأب الفرعي، الذي يدمج الرسائل برسالة واحدة ويرسلها برسالة بث متعدد موجهة لمجموعة الأب الفرعي.

يظهر الشكل **7** استدعاء الإجراء **4** للعقدة الجذر **G1**، حيث يتشكل لدينا ثلاث عقد كأب فرعي هي العقد **G6,G8,G4**.



الشكل 7. التحديث الدوري باستخدام التشفير والدمج أسفل الأب الفرعي.

الشكل **8** يوضح التمرير أسفل الأب الفرعي **G4**



الشكل 8. التشفير والدمج اسفل الأب الفرعي G4.

يتم التمرير بنفس الأسلوب بالنسبة لبقية فروع الشجرة، فعند وصول الاستدعاء إلى العقد **G8,G6**، ستكون هاتان العقدتان أبا فرعيا (أنظر الشكل **7**)، ويتم دمج رسائل التحديث الناتجة عن الأبناء كما وضحنا في حالة الاستدعاء من أجل العقدة **G4**.

يبين الجدول الآتي عدد عمليات التشفير والإرسال اللازمة لإجراء التحديث وفق الإجراء **4**:

message	message → Group	encryptions	type	sends
1	E (KG_4, KG'_4) || E (KG_{20}, KG'_{20}) → G4	2	multicast	4
2	E (KG_8, KG'_8) || E (KG_{12}, KG'_{12}) → G8	2	multicast	3
3	E (KG_6, KG'_6) → G6	1	multicast	2

جدول 9 . احصائيات الارسال والتشفير

- ✓ عدد عمليات التشفير = **5**

- ✓ عدد عمليات الإرسال= **9**

مناقشة الميزات:

- يحل هذا الإجراء العشوائية في إرسال رسائل التحديث المتوقع حدوثها بالإجراء **1** كما يضمن وصول المفاتيح المحددة للمستخدمين في المجموعات.
- عدد عمليات التشفير في هذا الإجراء مرتبط بعدد المجموعات المحدثة كما في الإجراء **3** وليس بعدد المجموعات الأوراق كما في الإجراء **2**.
- تم تفادي توجيه رسائل تحديث إلى المجموعات لا تتطلب تحديثاً كما في المجموعة **G3** وفروعها وبالتالي تم استثناء مشكلة الإجراء **3**.
- في حال كان مفتاح العقدة الجذر **G1** مطلوباً تحديثه فإن هذه الإجراء يؤول إلى الإجراء **3**
- في حال كان هناك عقدة واحدة فقط في الشجرة مطلوب تحديثها فإن هذا الإجراء يؤول إلى الإجراء **1** (التحديث عند الانتهاء).
- يحقق هذا الإجراء كما ذكرنا كل خواص الإجراء **2**.
- تم تخفيض عدد المقاطع المشفرة في رسلة التحديث.

4- الخاتمة:

قدمنا في هذه المقالة نهجا يعتمد على الأشجار لتمثيل الاتصالات بين المجموعات في مركز توزيع المفاتيح. علاوة على ذلك اقترحنا إجراءات عودية مناسبة للتمثيل الشجري لعملية التحديث الدوري لمفاتيح التشفير الخاصة بالاتصالات بين المجموعات، اعتمدنا في الإجراء الأول على تحديث مفتاح أية مجموعة ضمن النظام في حال انتهت صلاحيته وعرضنا مشكلة هذا الأسلوب في عدم الانتظام الممكن حدوثه في رسائل التحديث، قدمنا الإجراء **2** الذي يعتمد على تجميع المفاتيح المحدثة على كل مسار في الشجرة حتى الوصول إلى المجموعات التي تمثل عقد أوراقٍ بحيث يضمن إرسال رسالة تحديث واحدة إلى المستخدمين في كل مجموعة ورقة، وتجنبنا في هذا الإجراء إرسال رسائل غير مفيدة

لبعض المجموعات التي لا يجب ان تتأثر بالتحديث كما شرحنا مشكلة هذا الإجراء في أن عدد عمليات التشفير والإرسال اكبر من عدد المجموعات التي يشملها التحديث. بعد ذلك قدمنا الإجراء **3** والذي يقوم بتحديث النظام كاملاً برسالة واحدة ناتجة عن الدمج العودي لعدة أجزاء مشفرة حيث يستخلص كل مستخدم الجزء المهم بالنسبة له لعملية التحديث، وعرضنا مشكلة هذا الإجراء في وصول هذه الرسالة إلى مجموعات لا تهتم بمحتواها ولا يجب أن تتأثر بالتحديث هذا. وفي النهاية قدمنا الإجراء **4** الذي يحل مشكلات الإجراءات السابقة ويعتبر الأفضل من حيث عدد عمليات التشفير والإرسال اللازمة لإجراء التحديث الدوري لمفاتيح التشفير. ختمنا كل إجراء قمنا به في هذه المقالة بتوضيح عدد عمليات التشفير والإرسال اللازمة لإجراء التحديث وبمناقشة محاسنه ومساوئه.

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, November.

[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** - lolus: 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.Rhee., **2003** - Internet Security, Cryptographic Principles,Algorithms and Protocols. John Wiely & Sons, USA, Man.

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

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

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

[28] X.Zou, B.Ramamurthy and S.Magliveras., **2005** - SECURE GROUP COMMUNICATIONS OVER DATA NETWORKS, Springer Science+Business Media,ISBN **0-387-22970-1**.