基于大型动态多播群组的分层虚拟动态子群密钥管理方案

Hierarchical virtual dynamic subgroup key management scheme based on the large dynamic multicast groups

赵　刚　（山东大学计算机科学与技术学院、山东省特殊教育中等专业学校　250022）
徐秋亮　张　忠　董世强　（山东大学计算机科学与技术学院　250061）

摘要：在互联网高速发展的今天，密钥管理越来越被人们重视，对组通信密钥方面的研究早已开始，而现实中密钥子群的研究成果较少。分层虚拟动态子群密钥管理方案是对大型动态多播群组提出了一种分别基于子组和子组成员两个层次建立一级虚拟动态子组子群和二级虚拟动态成员子群的方法，解决了大型群组子群建立效率低下的问题。

关键词：密钥树　分层虚拟动态子群　组密钥管理　子组　子组成员

1　引言

随着多播通信技术的广泛应用，组通信密钥的安全也成为急需解决的问题。有时为完成某项任务，需要在网络中组成由大型群组的部分子组或不同子组的部分成员参与构成的动态群组，通常情况下需重建一个新的群组。本文首先简单介绍了大型动态多播群组的密钥管理，然后给出了分层虚拟动态子群密钥管理方案。

2　大型动态多播群组的密钥管理方案简介[3]

2.1　一个多播组的子组加入时

此时，组安全控制器密钥树的更新算法和协议与协议1基本相同。以文献[3]图2为例（为简单起见，本文均以完全二叉树为例），组通信密钥更新协议（协议3）如下：

（1）$SGSC_{B1} \rightarrow GSC_A$：加入请求；

（2）$GSC_A <=> SGSC_{B1}$：相互认证并发送会话密钥 $k_{SGSC_{B1}}$（该密钥同时又为该子组的子组组密钥）；

（3）GSC_A：生成新的组通信密钥 K'，同时执行协议1更新组安全控制器密钥树；

（4）A 中各 $SGSC_i$（包括 $SGSC_{B1}$）→{子组 i 的所有成员}：$\{K'\}_{K_{SGSC_i}}$ 以多播方式发送。

2.2　一个子组离开时

此时，子组安全控制器密钥树的更新算法和协议与协议2基本相同。以文献[3]图2为例，组通信密钥更新协议（协议4）如下：

（1）$SGSC_k \rightarrow GSC_B$：{离开－请求}$_{k_{SGSC_k}}$；

（2）$GSC_B \rightarrow SGSC_i$：{离开－同意}$_{k_{SGSC_i}}$；

（3）GSC_B：生成新的组通信密钥 K'，执行协议2更新组安全控制器密钥树；

（4）B 中各 $SGSC_i$（$i \neq k$）→{子组 i 的所有成员}：$\{K'\}_{K_{SGSC_i}}$ 以多播方式发送。

2.3　子组内有成员变动（加入或离开）时

此时组安全控制器密钥树的更新算法有所不同。对组安全控制器密钥树而言，此时并没有成员的离开和加入情况。即不能删除 u 节点。组通信密钥更新算法（协议5）如下：

（1）$SGSC_k$：得知有成员变动（离开或加入）后分别执行协议2或协议1，更新子组密钥树，生成新的子组组密钥 K'_{SGSC_k}。

（2）$SGSC_k \rightarrow GSC$：$\{K'_{SGSC_k}\}_{K_{SGSC_i}}$；

(3) $SGSC_k \rightarrow GSC$:{密钥更新 - 请求}$_{K_{SGSC_k}}$;

(4) $GSC \rightarrow SGSC_k$:{密钥更新 - 同意}$_{K_{SGSC_k}}$,同时产生新的组通信密钥 K';

(5) $GSC \rightarrow$ 各{$SGSC_l$ | $SGSC_l \in$ userset{son_j(root key)}}集合:{K'}$_{son_j(rootkey)}$分别以多播方式发送(注:son_j(root key) 表示 root key 节点的第 j 个子节点所代表的密钥。j=1,2,…m(m 表示 root key 有 m 个子节点)。

(6) 各 $SGSC_l$(包括 $SGSC_k$)→{子组 i 的所有成员}:{K'}$_{K_{SGSC_i}}$,以多播方式发送。

3 分层虚拟动态子群密钥管理方案

令安全群组 G 有子组 K 个,其子组集合 G={$SGSC_1$,$SGSC_2$,…,$SGSC_K$},每个子组均有 L 个成员,G 相应密钥树为 T,群管理者为 GSC,采取的密钥管理方案是文献[3]中介绍的协议,GSC 与 $SGSC_l$ 的密钥树结构如图 1 所示(本文均以平衡二叉树为例)。取子组子群 $S \subseteq G$,$S=\{SGSC_1, SGSC_2, \cdots, SGSC_m\}$,其中 $1 \leq m \leq K$,在 G 上建立一级虚拟动态子组子群,在各子组 $SGSC_l[i \in [1,m]]$ 上建立二级虚拟动态成员子群。

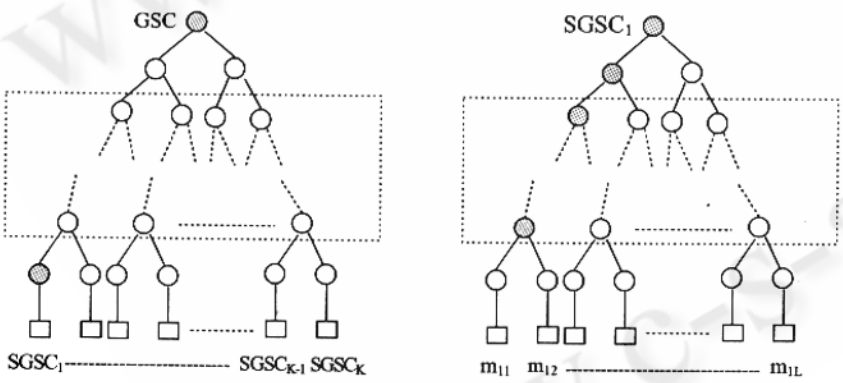

图 1 GSC 与 $SGSC_1$ 的密钥树结构

3.1 建立一级虚拟动态子组子群按如下协议

GSC 对组安全控制器密钥树进行有效裁减,保留与 S 中成员相关的从节点到根节点的路径,此外再建立一个虚拟的子组子群根节点,得到子组子群 S 的密钥树 T_S,从而子组与主根间的一级虚拟动态子组子群就建立起来了,GSC 任意选取一个随机数为密钥树 T_S 的根密钥 K_S,并进行密钥分配:

$GSC \rightarrow SGSC_i$:{K_S}$_{K_{SGSC_i}}$ $i \in [1,m]$

各 $SGSC_i \rightarrow${子组 i 的所有成员}:{K_S}$_{K_{SGSC_i}}$ $i \in [1,m]$

3.1.1 子组加入

分三种情况:

(1) 外部某子组 $SGSC_p$ 加入群组 G,但不加入某个一级虚拟动态子组子群。

用原文章方法,相当于单个子组的加入,参见文献[3] 3.1.2 1)。

(2) 外部某子组 $SGSC_p$ 加入群组 G,且加入某个一级虚拟动态子组子群 S。

先选中合适的插入节点,在尽量不增加树的高度的情况下,选择最浅最右端的节点为插入节点,将该子组 $SGSC_p$ 连接到密钥树 T_S 中,然后更新子组子群密钥 K_S 和群组密钥 K,更新后为 K'_S 和 K',并更新其他相应密钥。按照如下协议(协议 6):

① $SGSC_p \rightarrow GSC_A$:加入群组 A 的请求和加入一级虚拟动态子组子群 S 的请求。

② $GSC_A <=> SGSC_p$:相互认证并发送会话密钥 K_{SGSC_p}(该密钥同时又为该子组的子组组密钥);

③ GSC_A 生成新的组通信密钥 K',执行协议 1 更新组安全控制器密钥树;同时生成新的一级虚拟子组子群组通信密钥 K'_S,利用更新组安全控制器密钥树的 rekey 信息和 K_S 以及 $SGSC_p$ 的密钥 K_{SGSC_p} 传递 K'_S;

④ G 中各 $SGSC_i$(包括 $SGSC_p$)→{子组 i 的所有成员}:{K'}$_{K_{SGSC_i}}$ 以多播方式发送;

一级虚拟子组子群中各 $SGSC_i \rightarrow${子组 i 的所有成员}:{K'_S}$_{K_S}$ 以多播方式发送。

(3) G 中某子组 $SGSC_p$ 加入 S

GSC 将 T 中 $SGSC_p$ 关联的叶节点及其到根的路径上的中间节点合并到 T_S 中,并生成新的子组子群密钥 K'_S

$GSC \rightarrow SGSC_i$:{K'_S}$_{K_{SGSC_i}}$ $i \in [1,m]$

$GSC \rightarrow SGSC_p$:{K'_S}$_{K_{SGSC_p}}$

然后 T_S 中各 $SGSC_l$ 更新成员密钥:

各 $SGSC_l$(包括新加入子组)→{子组 i 的所有成员}:{K'_S}$_{K_{SGSC_i}}$

3.1.2 子组离开

(1) 被删除子组 $SGSC_p$ 是 GSC 的子组,但不是任何一级虚拟子组子群 S 的子组,相当于 GSC 中单个子组的删除,参见文献[3] 3.1.2 2)。

(2) 被删除子组 $SGSC_p$ 是 GSC 的子组,也是某个一级虚拟子组子群 S 的非唯一子组。

GSC 首先查找 $SGSC_p$ 所关联的子群 S（T_S 为其密钥树），选取 2 个随机数为 T 与 T_S 的新根密钥 K'与 K'$_S$，然后删除 $SGSC_p$ 在 T 与 T_S 中所关联的叶节点，并将从该节点到 T 的根节点这条路径上的所有中间节点所关联的密钥更新。按照如下协议（协议 7）：

① $SGSC_p \rightarrow GSC$：{离开 GSC－请求}$_{K_{sgsc_p}}$，{离开一级虚拟子组子群 S－请求}$_{K_{sgsc_p}}$；

② $GSC \rightarrow SGSC_p$：{离开 GSC－同意}$_{K_{sgsc_p}}$，{离开一级虚拟子组子群 S－同意}$_{K_{sgsc_p}}$；

③ GSC 生成新的组通信密钥 K'，执行协议 2 更新组安全控制器密钥树；同时生成新的一级虚拟子组子群组通信密钥 K'$_S$，利用更新组安全控制器密钥树的 rekey 信息和 K_S 传递 K'$_S$；

④ G 中各 $SGSC_i$（$i \neq p$）→{子组 i 的所有成员}：{K'}$_{K_{sgsc}}$ 以多播方式发送；

一级虚拟子组子群 S 中各 $SGSC_i$（$i \neq p$）→{子组 i 的所有成员}：{K'$_S$}$_{KS}$ 以多播方式发送。

（3）被删除子组 $SGSC_p$ 是 GSC 的子组，也是某个一级虚拟子组子群 S 的唯一子组。

此时删除子组 $SGSC_p$ 即相当于删除该子群，按原方案协议 4 更新组密钥树即可。

3.1.3　两个一级虚拟动态子组子群的合并

假设有两个一级虚拟动态子组子群 S 和 T，其密钥树分别为 T_S 和 T_T，虚拟根钥分别为 K_S 和 K_T，群管理员 GSC 重新选择一个新的虚拟根钥 K_{ST}，将 T_S 和 T_T 合并成一个新的子树，并生成新的一级虚拟子组子群为 ST，同时进行密钥分配。按如下协议（协议 8）：

（1）S→GSC：{与 T 合并－请求}$_{KS}$；

（2）T→GSC：{与 S 合并－请求}$_{KT}$；

（3）GSC→S：{与 T 合并－同意}$_{KS}$；

（4）GSC→T：{与 S 合并－同意}$_{KT}$；

（5）GSC 生成新的合并后的一级虚拟子组子群 ST 的组通信密钥 K_{ST}和虚拟根，

GSC→{一级虚拟子组子群 ST 中的各 $SGSC_i$}：{K_{ST}}$_{K_{sgsc_i}}$分别以多播方式发送；

一级虚拟子组子群 ST 中个 $SGSC_i$→{子组 i 的所有成员}：{K_{ST}}$_{K_i}$，{K_{ST}}$_{K_T}$以多播方式发送。

3.2　对任一子组 $SGSC_i$，$SGSC_i \in G$，$SGSC_i$ 对应的密钥树为 T_{SGSC_i}，G_i 为该子组内所有成员集合，即 $G_i = \{m_{i1}, m_{i2}, \cdots m_{iL}\}$，在内部成员 m_{ij}（$1 \leqslant j \leqslant L$）上如上述方法建立二级虚拟动态成员子群。即取内部成员子群 $S_i \subseteq G_i$，$S_i = \{m_{i1}, m_{i2}, \cdots, m_{ih}\}$，且 $1 \leqslant h \leqslant L$。GSC 对该子组成员密钥树 T_{SGSC_i}进行有效裁减，保留与 S_i 中成员相关的从节点到根节点的路径，此外再对应该虚拟成员子群建立一个虚拟的成员子群根节点，从而得到成员子群 S_i 的密钥树 T_{S_i}，且任意选取一个随机数为密钥树 T_{S_i} 的根密钥 K_{S_i}，并且进行：

$SGSC_i$→{二级虚拟成员子群 S_i 的所有成员 m_{ij}}：{K_{S_i}}$_{K_{sgsc}}$，$i \in [1,h]$，从而二级虚拟动态成员子群就建立起来了。

3.2.1　成员加入

分三种情况：

（1）外部某成员 m_{ip}加入某子组 $SGSC_i$，但不加入某二级虚拟成员子群。

相当于单个成员的加入，参见文献[3] 3.1.2 3）。

（2）外部某成员 m_{ip}加入某子组 $SGSC_i$，且加入二级虚拟成员子群 S_i。

先选中合适的插入点，在尽量不增加树的高度的情况下，选择最浅最右端的节点为插入点，将该成员 m_{ip}连接到密钥树 T_{Si} 中，然后更新子群组密钥 K_{Si} 和子组组密钥 K_{SGSC_i}，更新后的密钥为 K'$_{Si}$ 和 K'$_{SGSC_i}$，并更新其他相应密钥，成员 m_{ip}加入子组的同时，$SGSC_i$ 也将它加入到二级虚拟成员子群 S_i 中。按如下协议（协议 9）进行：

① $SGSC_i$ 生成新的子组组密钥 K'$_{SGSC_i}$，按成员加入协议 1 更新子组密钥树；同时生成新的二级虚拟成员子群组密钥 K'$_{Si}$，利用更新子组密钥树的 rekey 信息和 K_{Si} 以及新加入成员的密钥传递 K'$_{Si}$；

② $SGSC_i \rightarrow GSC$：{K'$_{SGSC_i}$}$_{K_{sgsc}}$，{K'$_{S_i}$}$_{K_{sgsc}}$；

③ $SGSC_i \rightarrow GSC$：{密钥更新－请求}$_{K'_{sgsc}}$，{二级虚拟成员子群密钥更新－请求}$_{K'_{S_i}}$；

④ $GSC \rightarrow SGSC_i$：{密钥更新－同意}$_{K'_{sgsc}}$，{二级虚拟成员子群密钥更新－同意}$_{K'_{sgsc}}$；同时产生新的组通信密钥 K'和新的一级虚拟子组子群组通信密钥 K'$_S$；

⑤ GSC→各{$SGSC_i$ | $SGSC_i \in$ userset{son_i(rootkey)}}集合：{K'}$_{son_i(rootkey)}$分别以 多播方式发送；

各 $SGSC_i$→{子组 i 的所有成员 m_{ij}}：{K'}$_{K'_{sgsc}}$，以多播方式发送。

GSC→{一级虚拟子组子群中各 $SGSC_i$}：{K'$_S$}$_{K_S}$；

一级虚拟子组子群中各 $SGSC_i$→{子组 i 的所有成员 m_{ij}（其中包括 m_{ip}）}：{K'$_S$}$_{K'_{sgsc}}$以多播方式发送。

（3）$SGSC_i$ 中某成员 m_{ip}加入 S_i，令 $SGSC_i$ 对应的密钥树为 T_{SGSC_i}。$SGSC_i$ 将中 m_{ip}关联的叶节点及其到

$SGSC_1$ 的路径上的中间节点合并到成员子群 S_1 的密钥树 T_{S_1} 中，并生成新的成员子群密钥 K'_{S_1}。

$SGSC_1 \rightarrow m_{1j}: \{K'_{S_1}\}_{K_{m_{1j}}}$　$j \in [1,h]$，$j \neq m_{1p}$ 为成员 mij 对应的密钥。

$SGSC_1 \rightarrow m_{1p}: \{K'_{S_1}\}_{K_{m_{1p}}}$

3.2.2 成员离开

分两种情况：

(1) 被删除成员 m_{1p} 是 $SGSC_1$ 的成员，但不是任何成员子群 S_1 的成员，相当于单个成员的删除，参见文献[3] 3.1.2 3)。

(2) 7 被删除成员 m_{1p} 是 $SGSC_1$ 的成员，也是某个二级虚拟成员子群 S_1 的非唯一成员。按如下协议(协议 10)：

① $SGSC_1$ 得知成员离开，生成新的子组组密钥 K_{SGSC_1}，执行协议 2 更新子组密钥树；同时生成新的二级虚拟成员子群组密钥 K'_{S_1}，在更新子组密钥树的同时利用 rekey 信息和 K_{S_1} 传递 K'_{S_1}

② $SGSC_1 \rightarrow GSC: \{K'_{SGSC_1}\}_{K_{SGSC_1}}, \{K'_{S_1}\}_{K_{ij}}$

③ $SGSC_1 \rightarrow GSC: \{$密钥更新 − 请求$\}_{K'_{SGSC_1}}, \{$二级虚拟成员子群密钥更新 − 请求$\}_{K_{S_1}}$；

④ $GSC \rightarrow SGSC_1: \{$密钥更新 − 同意$\}_{K'_{SGSC_1}}, \{$二级虚拟成员子群密钥更新 − 同意$\}_{K_{S_1}}$；

同时生成新的组通信密钥 K'和新的一级虚拟子组子群组通信密钥 K'_S；

⑤ $GSC \rightarrow$ 各 $\{SGSC_1 \mid SGSC_1 \in userset\{son_1(rootkey)\}\}$ 集合：$\{K'\}_{son_1(rootkey)}$ 分别以多播方式发送；

各 $SGSC_1 \rightarrow \{$子组 i 的所有成员 $m_{1j}(j \neq p, j \in [1,L])\}: \{K'\}_{K'_{SGSC_1}}$，以多播方式发送；

$GSC \rightarrow \{$一级虚拟子组子群中的 $SGSC_1\}: \{K'_S\}_{K_{S_1}}$；

一级虚拟子组子群中的 $SGSC_1 \rightarrow \{$子组 i 所有成员 $m_{1j}(j \neq p, j \in [1,h])\}: \{K'_S\}_{K'_{S_1}}$ 以多播方式发送。

(3) 被删除成员 m_{1p} 是 $SGSC_1$ 的成员，同时是某个二级虚拟成员子群 S_1 的唯一成员。

此时删除该成员 m_{1p} 即相当于删除该子群 S_1，按原方案协议 5 更新子组密钥树和组密钥树即可。

3.2.3 两个二级虚拟成员子群的合并

假设有子组 $SGSC_1$ 的两个二级虚拟成员子群 S 和 T，其密钥树分别为 T_S 和 T_T，虚拟根钥分别为 K_S 和 K_T，群管理员 GSC 重新选择一个虚拟根 K_{ST}，并将 T_S 和 T_T 合并成一棵新的子树，生成新的二级虚拟成员子群 ST，同时进行密钥分配。按如下协议(协议 11)：

① $S \rightarrow SGSC_1: \{$与 T 合并 − 请求$\}_{K_S}$；

② $T \rightarrow SGSC_1: \{$与 S 合并 − 请求$\}_{K_T}$；

③ $SGSC_1 \rightarrow S: \{$与 T 合并 − 同意$\}_{K_S}$；

④ $SGSC_1 \rightarrow T: \{$与 S 合并 − 同意$\}_{K_T}$；

⑤ $SGSC_1$ 生成新的合并后的二级虚拟成员子群 ST 的组通信密钥 K_{ST}；

$SGSC_1 \rightarrow \{$新的二级虚拟成员子群 ST 中的全体成员$\}: \{K_{ST}\}_{K_S}, \{K_{ST}\}_{K_T}$ 以多播方式发送。

项目 \ 方案		本方案	重新建立新群体	
			原方案	WGL 98 方案
子群的组建		极快	慢	慢
m_{11} 存储的密钥信息		2	$1+\log_2 L$	$1+\log_2 L+\log_2 K$
$SGSC_1$ 维护的密钥数		2	$2L-1$	-
GSC_1 维护的密钥数		$K+1$	$2K-1$	$2LK-1$
m_{11} 离开 GSC_1	更新的密钥数	2	$1+\log_2 L$	$\log_2 L+\log_2 K$
	$SGSC_1$ rekey 的信息包数	1	$2+\log_2 L$	-
	$SGSC_1$ 的加密计算开销	3	$2+2\log_2 L$	-
m_{11} 由 GSC_2 加入 GSC_1 中的 $SGSC_1$，且加入相应二级虚拟动态子群	更新的密钥数	2	$2+\log_2 L$	$1+\log_2 L+\log_2 K$
	$SGSC_1$ rekey 的信息包数	1	$3+\log_2 L$	-
	$SGSC_1$ 的加密计算开销	4	$2+2\log_2 L$	-
$SGSC_1$ 离开 GSC_1	更新的密钥数	1	$\log_2 K$	$L(\log_2 L+\log_2 K)$
	GSC_1 rekey 的信息包数	0	$\log_2 K$	$L(\log_2 L+\log_2 K)$
	GSC_1 的加密计算开销	1	$2\log_2 K$	$2(\log_2 L+\log_2 K)$
$SGSC_1$ 由 GSC_2 加入 GSC_1，且加入相应一级虚拟动态子组子群	更新的密钥数	1	$1+\log_2 K$	$L(1+\log_2 L+\log_2 K)$
	GSC_1 rekey 的信息包数	0	$1+\log_2 K$	$L(1+\log_2 L+\log_2 K)$
	GSC_1 的加密计算开销	2	$2\log_2 K$	$2L(\log_2 L+\log_2 K)$

(下转第 62 页)

(上接第 58 页)

4 性能分析

分层虚拟动态子群密钥管理方案是基于大型动态多播群组的,其安全性等价于该方案,对于选择大型群组的不同的多个子组和在不同子组内选取若干成员建立子群这一要求,原方案需利用现有密钥分配方案进行重建,而我们的方案是在原有密钥树的基础上进行子群重构,实现不同层次的虚拟,可减少信息传输和密钥存储空间。

令有群组 GSC_1 和 GSC_2,各有 K 个子组,每个子组内有 L 个成员,GSC_1 中有子组 $SGSC_1$,$SGSC_1$ 中有成员 m_{11},令 m_{11} 在 $SGSC_1$ 所属的一级虚拟动态子组子群的二级虚拟动态成员子群中,以表 1 为例进行分析。

5 结束语

本文设计的分层虚拟动态子群密钥管理方案,能够高效快速地完成大型群组的各种子群的重建,与原方案重建子群相比,节省了大量的密钥存储空间和加密计算开销,具有更强的可扩展性。下一步需解决的问题是:分层多级虚拟的实现问题,同时要求达到密钥存储最小、效率最高,这有待于进一步研究。

参考文献

1 C. K. Wong, M. G. Gouda, and S. S. Lam. Secure group communications using key graphs. In Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication, 1998. 68 – 79.

2 李先贤、怀进鹏、刘旭东,群密钥分配的动态安全性及其方案. 计算机学报,2002,25(4):337 – 345.

3 刘璟、周明天,大型动态多播群组的密钥管理和访问控制. 软件学报, 2002,13(2):291 – 297.

4 张忠、徐秋亮、原变青,虚拟动态子群密钥管理方案,计算机工程与设计,2006,27(5):731 – 734.