一种安全可靠的集中式组播密钥管理方案

邓淑华 赵泽茂
(杭州电子科技大学通信与信息系统研究所 杭州 310018)

摘 要 利用Hash函数和二元单向函数,通过构造两个特殊的多项式,提出一种新的组播密钥管理方案——基于哈希函数的组播密钥管理(HFMKM)方案。此方案整合了集中式与分布式的特性,克服了传统集中式平面型管理方式的安全性完全依赖于组管理者的弊端。将其与GKMP方案进行了比较和分析。结果表明,HFMKM方案安全性明显提高,并提升了整体性能,是一种安全的新型组播密钥管理方案。

关键词 哈希函数,二元单向函数,密钥管理,安全组通信,前向加密,后向加密
中图法分类号 TP309.2 **文献标识码** A

Secure and Reliable Centralized Multicast Key Management Scheme

DENG Shu-hua ZHAO Ze-mao
(College of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract By constructing two special polynomials, a new multicast key management (MKM) scheme——Hash Function-based Multicast Key Management (HFMKM) scheme was proposed, which was based on hash function and two-variable one-way function. The properties of both centralized and distributed were combined in this scheme, so it overcame the defect of conventional centralized flat schemes, whose security totally depended on the Group Controller (GC). With Comparing and analyzing it with GKMP scheme, the results showed that the HFMKM scheme obviously improved the security and the overall performance. It is a new security MKM scheme.

Keywords Hash function, Two-variable one-way function, Key management, Secure group communication, Forward confidentiality, Backward confidentiality

1 引言

组播通信技术提高了数据的传送效率,降低了网络出现拥塞的可能,有效地节约了带宽,使服务器的负担减轻[1],可广泛应用于多媒体远程教育、分布式系统、网络视频会议等。组播是实现大规模信息资源共享的一种重要方式,而安全组播则是实现大规模信息资源共享的重要保证。然而,目前的组播协议缺乏足够安全的机制来满足组播应用的安全性需求,组播安全问题包括数据保密、组管理和源认证等多个方面。组播密钥管理解决组管理的安全问题,主要包括两个方面:一是密钥的产生与分发;二是对密钥进行管理以适应组成员关系的变化。组播密钥管理的重点是对参与组播的成员进行管理和更新组密钥[2]。

与单播密钥管理相比,前向加密、后向加密和同谋破解是组播密钥管理特有的问题[3]。前向加密指主动退出组播的节点或被强制退出的节点(比如被俘节点、恶意节点)无法利用它们所知的密钥解密后继的组播报文。后向加密则是新加入的组成员无法破解它加入之前的组播报文。同谋破解指掌握了足够密钥信息的几个恶意节点联合起来,使得系统无论如何更新密钥,都可以破解。组播密钥管理方式可以分为集中式(Centralized)和分布式(Distributed)两种管理方式。迄今为止,已经出现了许多组播密钥管理方案,如典型的集中式组密钥管理方案有:GKMP(Group Key Management Protocol)[4]、集中式分组型Iolus方案[5]。而基于身份认证的群组密钥协商协议见文献[6]。近年来国内学者也提出各种重要方案[7-10],它们在计算复杂性、安全性和通信复杂度方面各有长短。本文提出了一种新的集中式平面型的组播密钥管理方案——基于哈希函数的组播密钥管理(Hash Function-based Multicast Key Management,HFMKM)方案,给出了其实现过程,并分析了其性能。

2 预备知识

2.1 Hash函数

Hash函数或称哈希函数,又称散列函数、杂凑函数,是一个将可变长度的数据M通过单向算法压缩成另一个固定长度的数据(散列值)h的函数,符号表示为:$h=H(M)$,其中$H(\cdot)$就为Hash函数,h称为哈希值。Hash函数具有以下两个重要性质:

(1) 单向性:给定任意的散列值h,找到满足$H(x)=h$的x在计算上不可行;

本文受浙江省自然科学基金(Y1100818)资助。
邓淑华(1985-),男,硕士生,主要研究方向为密码学、通信安全,E-mail:shuhua.deng@163.com;赵泽茂(1965-),男,博士,教授,主要研究方向为密码学与信息安全。

(2) 抗碰撞性:对于任意的 x 和 y,若 $x\neq y$,则 $H(x)=H(y)$在实际环境中不可满足。

2.2 二元单向函数

在一元哈希函数的基础上,本文方案需要定义一个特定的二元函数。

定义 1(二元单向函数) 有两个自变量 r 和 s 的函数 $f(r,s)$称为二元单向函数,如果它满足以下性质:(1)已知 r 和 s,$f(r,s)$容易计算;(2)已知 s 和 $f(r,s)$,求 r 在计算上不可行;(3)在 r 未知的情况下,对任意的 s,难以计算 $f(r,s)$;(4)在已知 s 的情况下,找到 $r_1\neq r_2$ 使得 $f(r_1,s)=f(r_2,s)$在计算上不可行;(5)已知任意多的$(s_i,f(r,s_i))$对,对求 $f(r,s_j)$是没有帮助的,其中 $s_i\neq s_j$。

3 HFMKM 方案

组控制者(Group Controller,GC)和成员节点之间不存在密钥层次树结构,方案的结构是一种平面型结构(见图 1),与 GKMP 方案[4]的体系结构相同。为叙述方便,文中常用符号如表 1 所列。

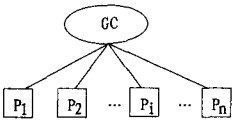

图 1 HFMKM 方案的体系结构

表 1 本文常用的符号

符号	含义
$k_i(i=1,2,\cdots,N)$	N 次多项式 $S(t)$的系数,也是候选的组密钥的集合
$h_{i,j}$	GC 与节点 P_i 第 j 次密钥更新时各方所取哈希值($j=0,1,\cdots$,下同。$h_{i,0}$表示密钥建立时所取的哈希值)
$u_{i,j}$	GC 生成的对应 P_i 的第 j 个秘密随机数($u_{i,0}$为密钥建立时的随机数)
$t_{i,j}$	GC 与成员 P_i 各自求得的第 j 个 t 值($t_{i,0}$为密钥建立时的 t 值)
σ_j	GC 随机生成的第 j 次密钥更新因子(σ_0 为密钥建立时的生成因子)
K_i	GC 与成员 P_i 共享的会话密钥(P_i 的私钥)
r_i	成员 P_i 的编号,是公开的
$\lfloor x\rfloor \bmod p$	对 x 取整后再对大素数 p 求模取最小正整数值
$\|\|$	数值连接符
$GC\rightarrow\{P\}:\{M\}_K$	GC 用密钥 K 加密消息 M 后发送给成员 P
$\{M\}$	将消息 M 打包,利于分组接收
K_g	组密钥
K_g'	更新后的组密钥

3.1 初始化阶段

(1)GC 预置多项式 $S(t)=k_1t+k_2t^2+\cdots+k_Nt^N$,其中 k_i $(i=1,2,\cdots,N)$为 GC 随机选取的 N 个大数。假设组中有 n 个成员 $P_1,P_2,\cdots,P_n$,GC 和成员 $P_i(i=1,2,\cdots,n)$通过密钥交换协议产生会话密钥 K_i(P_i 的私钥),GC 通过 K_i 加密给 P_i 发送一个随机种子 $Seed_i$(所有的 $Seed_i$ 互不相同)、一个相同的 Hash 函数 $H(\cdot)$和一个相同的二元单向函数 f:$GC\rightarrow\{P_i\}:\{Seed_i,H(\cdot),f\}_{K_i}$。

(2)GC 与每个成员 P_i 都需要计算哈希值,而哈希函数的执行由同步机制来协同一致,每更新一次密钥则在前一次存储的哈希值的基础上再执行一次哈希函数,如第 j 次密钥更新时节点 P_i 所取的哈希值为 $h_{i,j}=H^j(Seed_i)$,并约定各节点初始的哈希值取为 $h_{i,0}=Seed_i$;GC 随机生成 $u_{i,0}$ 并由 $h_{i,0}$计算 $t_{i,0}=f(u_{i,0},h_{i,0})$,再计算 $S(t_{i,0})=k_1t_{i,0}+k_2t_{i,0}^2+\cdots+k_Nt_{i,0}^N$;值得注意的是,GC 与成员 P_i 各自求得的第 j 个 t 值 $t_{i,j}$必须满足:$t_{i,j}>k_l$,$i=1,2,\cdots,n$;$j=0,1,2,\cdots$;$l=1,2,\cdots,N(t_{i,0}$为密钥建立时的 t 值,其中 $k_i(i=1,2,\cdots,N)$为多项式 $S(t)$的系数)。

(3)GC 再构造一系列关于成员身份 r 的特殊多项式:$T_{i,j}(r)=I(r)\cdot u_{i,j}+h_{i,j}$,$i=1,2,\cdots,n$;$j=0,1,\cdots$,其中 $I(r)$是关于成员身份 r 的多项式,在不同阶段是动态变化的,在下面的过程中将详细说明。

3.2 密钥建立

(1)GC 产生密钥生成因子 $\sigma_0(\sigma_0\in\{1,2,\cdots,N\})$;

(2)GC 与所有成员 $P_i(i=1,2,\cdots,n)$约定成员身份 r 的多项式 $I(r)\equiv1$;

(3)GC 将 σ_0、$S(t_{i,0})$和 $T_{i,0}(r)=I(r)\cdot u_{i,0}+h_{i,0}=u_{i,0}+h_{i,0}$广播到所有的成员;

(4)所有成员 $P_i(i=1,2,\cdots,n)$收到 σ_0、$S(t_{i,0})$和 $T_{i,0}(r)$后,根据初始哈希值 $h_{i,0}=Seed_i$ 得 $u_{i,0}=T_{i,0}(r)-h_{i,0}$,从而 $t_{i,0}=f(u_{i,0},h_{i,0})$;

(5)$P_i(i=1,2,\cdots,n)$和 GC 都获得了 $S(t_{i,0})$、σ_0 以及 $t_{i,0}$,就可以根据下式求得组密钥:$K_g=\lfloor S(t_{i,0})/t_{i,0}^{\sigma_0}\rfloor(\bmod t_{i,0})=k_{\sigma_0}$。因为 $t_{i,0}>k_l$,$i=1,\cdots,n$;$l=1,\cdots,N$,$S(t_{i,0})/t_{i,0}^{\sigma_0}=(k_1t_{i,0}+\cdots+k_{\sigma_0-1}t_{i,0}^{\sigma_0-1})/t_{i,0}^{\sigma_0}+k_{\sigma_0}+(k_{\sigma_0+1}t_{i,0}+k_{\sigma_0+2}t_{i,0}^2+\cdots+k_Nt_{i,0}^{N-\sigma_0})$,故$\lfloor S(t_{i,0})/t_{i,0}^{\sigma_0}\rfloor=k_{\sigma_0}+(k_{\sigma_0+1}t_{i,0}+\cdots+k_Nt_{i,0}^{N-\sigma_0})$,从而 $k_{\sigma_0}=\lfloor S(t_{i,0})/t_{i,0}^{\sigma_0}\rfloor(\bmod t_{i,0})$,即每个成员和 GC 获得相同的组密钥 $K_g=k_{\sigma_0}$,至此组密钥生成结束。

3.3 密钥更新

在 HFMKM 方案中,有 3 种情况需要进行密钥更新:周期性更新、成员加入和成员离开。为了组通信安全,GC 应当周期性地更新组密钥(周期应大于同步时延);成员在加入组时向 GC 发送加入请求,GC 验证通过后允许成员加入时要更新组密钥;成员离开组时,GC 删除成员也要更新组密钥。

对于前述的 σ_j、$u_{i,j}$、$h_{i,j}$、$t_{i,j}$和 $T_{i,j}(r)$,在密钥更新前后不妨分别记为 σ_1、$u_{i,1}$、$h_{i,1}$、$t_{i,1}$和 $T_{i,1}(r)$,以及 σ_2、$u_{i,2}$、$h_{i,2}$、$t_{i,2}$和 $T_{i,2}(r)$;组密钥更新前后分别记为 K_g 和 K'_g。下面通过周期性更新、成员加入和成员离开 3 种情况来说明 HFMKM 方案的密钥更新方法。

3.3.1 周期性更新

组密钥周期性更新的具体过程如下:

(1)GC 随机生成密钥更新因子 $\sigma_2(\sigma_2\in\{1,2,\cdots,N\}$且 $\sigma_2\neq\sigma_1)$,并求得 $h_{i,2}=H(h_{i,1})$,$i=1,2,\cdots,n$;

(2)GC 根据 $u_{i,2}$ 和 $h_{i,2}$ 计算 $t_{i,2}=f(u_{i,2},h_{i,2})$,进而 $S(t_{i,2})=k_1t_{i,2}+k_2t_{i,2}^2+\cdots+k_Nt_{i,2}^N$以及 $T_{i,2}(r)=I(r)\cdot u_{i,2}+h_{i,2}$,$i=1,2,\cdots,n$(注意此时 $I(r)$为组密钥更新前的值);

(3)GC 将 σ_2 以及 $S(t_{i,2})$和 $T_{i,2}(r)$组成的数据包连成的数据串组播发送到所有组成员节点:$GC\rightarrow\{P_1,\cdots,P_n\}:\{\sigma_2,\{S(t_{1,2}),T_{1,2}(r)\}\|\cdots\|\{S(t_{n,2}),T_{n,2}(r)\}\}_{K_g}$;

(4)节点 $P_i(i=1,2,\cdots,n)$接收 σ_2、数据包$\{S(t_{i,2}),T_{i,2}(r)\}$后,根据存储的 $h_{i,1}$,求得 $h_{i,2}=H(h_{i,1})$,再由 $h_{i,2}$及 $I(r)$求得 $u_{i,2}=(T_{i,2}(r)-h_{i,2})/I(r)$,从而得到 $t_{i,2}=f(u_{i,2},h_{i,2})$;

(5)最后，所有成员 $P_i(i=1,2,\cdots,n)$ 和 GC 都可以算出更新后的组密钥：$K'_g=\lfloor S(t_{i,2})/t_{i,2}^{\sigma_2}\rfloor(\mathrm{mod}\ t_{i,2})=k_{\sigma_2}$。至此，组密钥周期性更新完成。

3.3.2 成员加入

假设成员 P_{n+1} 要求加入组，此时组中已有 n 个成员 P_1，P_2，$\cdots$，P_n，如图 2 所示。成员 P_{n+1} 向 GC 发送加入请求，GC 接收到成员的加入请求后，进行成员身份验证以决定是否接受该成员加入这个组。如果决定接受这个成员加入组，GC 需要进行密钥更新。成员加入时的密钥更新过程如下：

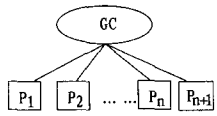

图 2 成员 P_{n+1} 加入

(1)GC 和成员 P_{n+1} 通过密钥交换协议产生会话密钥 K_{n+1}，GC 随机生成一个种子 $Seed_{n+1}$；

(2)GC 生成密钥更新因子 $\sigma_2(\sigma_2\in\{1,2,\cdots,N\})$，并求得 $h_{i,2}=H(h_{i,1})(i=1,2,\cdots,n)$ 以及 $h_{n+1,2}=Seed_{n+1}$；GC 生成随机数 $u_{n+1,2}$（$h_{n+1,2}$ 与 $u_{n+1,2}$ 中 j 值取 2 只是为了形式上与叙述上的方便，下面的 $T_{n+1,2}(r)$ 同理），对 $i=1,2,\cdots,n+1$，GC 根据 $u_{i,2}$ 和 $h_{i,2}$ 计算 $t_{i,2}=f(u_{i,2},h_{i,2})$，进而计算 $S(t_{i,2})=k_1t_{i,2}+k_2t_{i,2}^2+\cdots+k_Nt_{i,2}^N$；

(3)GC 再计算一个关于成员身份 r 的多项式：$T_{n+1,2}(r)=I(r)\cdot u_{n+1,2}+h_{n+1,2}$（$I(r)$ 为组密钥更新前的值），然后：$GC\rightarrow\{P_{n+1}\}:\{\sigma_2,Seed_{n+1},H(\cdot),f,I(r),S(t_{n+1,2}),T_{n+1,2}(r)\}_{K_{n+1}}$；

(4)$GC\rightarrow\{P_1,P_2,\cdots,P_n\}:\{\sigma_2,\{S(t_{1,2}),T_{1,2}(r)\}||\cdots||\{S(t_{n,2}),T_{n,2}(r)\}\}_{K_g}$；

(5)节点 $P_i(i=1,2,\cdots,n)$ 根据存储的 $h_{i,1}$，求得 $h_{i,2}=H(h_{i,1})$，P_{n+1} 则是 $h_{n+1,2}=Seed_{n+1}$；然后 $P_i(i=1,2,\cdots,n+1)$ 再由 $h_{i,2}$ 及 $I(r)$ 求得 $u_{i,2}=(T_{i,2}(r)-h_{i,2})/I(r)$，从而 $t_{i,2}=f(u_{i,2},h_{i,2})$，最后所有成员和 GC 可求得更新的组密钥：$K_g'=\lfloor S(t_{i,2})/t_{i,2}^{\sigma_2}\rfloor(\mathrm{mod}\ t_{i,2})=k_{\sigma_2}$。至此，成员加入的组密钥更新完成。

3.3.3 成员离开

假设组中有 n 个成员 P_1，P_2，$\cdots$，P_n，成员 P_l 要求离开组、被俘获或为恶意节点时，GC 需要将成员 P_l 删除，并进行密钥更新，如图 3 所示。成员离开时的密钥更新过程如下：

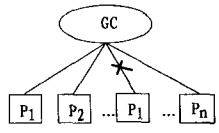

图 3 成员 P_l 离开

(1)GC 删除成员 P_l 的节点，即不再保存与 P_l 有关的信息，并将关于成员身份 r 的多项式 $I(r)$ 更新为 $I'(r)=I(r)\cdot(r-r_l)$，其中 r_l 是 P_l 的身份编号；同时将 $T_{i,1}(r)(i=1,2,\cdots,n,i\neq l)$ 更新为 $T_{i,2}(r)=I'(r)\cdot u_{i,2}+h_{i,2}$，其中 $u_{i,2}$ 是 GC 产生的随机数；

(2)GC 生成 $\sigma_2(\sigma_2\in\{1,2,\cdots,N\}$ 且 $\sigma_2\neq\sigma_1)$，并求得 $h_{i,2}=H(h_{i,1})(i=1,2,\cdots,n,i\neq l)$，再由 $u_{i,2}$ 和 $h_{i,2}$ 计算 $t_{i,2}=f(u_{i,2},h_{i,2})$，从而 $S(t_{i,2})=k_1t_{i,2}+k_2t_{i,2}^2+\cdots+k_Nt_{i,2}^N$；

(3)$GC\rightarrow\{P_1,\cdots,P_{l-1},P_{l+1},\cdots,P_n\}:\{I'(r),\sigma_2,\{S(t_{1,2}),T_{1,2}(r)\}||\cdots||\{S(t_{n,2}),T_{n,2}(r)\}\}_{K_g}$；

(4)节点 $P_i(i=1,2,\cdots,n,i\neq l)$ 根据存储的 $h_{i,1}$，求得 $h_{i,2}=H(h_{i,1})$，再由 $h_{i,2}$ 及 $I'(r)$ 求得 $u_{i,2}=(T_{i,2}(r)-h_{i,2})/I'(r)$，从而求得 $t_{i,2}=f(u_{i,2},h_{i,2})$，最后各方求得更新的组密钥：$K'_g=\lfloor S(t_{i,2})/t_{i,2}^{\sigma_2}\rfloor(\mathrm{mod}\ t_{i,2})=k_{\sigma_2}$，成员离开的组密钥更新完成。

4 分析与讨论

4.1 安全性

从以上密钥生成和更新过程可以看出，方案中组密钥的安全性主要依赖于求取 t 值 $t_{i,j}=f(u_{i,j},h_{i,j})$ 的困难性。具体而言，一方面 GC 通过发送多项式值 $S(t_{i,j})$ 和密钥更新因子 σ_j 来控制组密钥的生成；另一方面，各合法成员根据自身的 Hash 函数 $H(?)$ 与二元单向函数 f 安全地生成 t 值 $t_{i,j}=f(u_{i,j},h_{i,j})$，其中 $u_{i,j}$ 是 GC 生成的对应 P_i 的第 j 个随机数（$j=0,1,\cdots$），$h_{i,j}=H(h_{i,j-1})$（即每次哈希值的求取是以前一次的哈希值作为输入的）；第三，由二元单向函数 f 的特性可以看出，若无法通过 $I(r)$ 求出 $u_{i,j}$，即使 $h_{i,j}$ 被敌手破解，依然无法破解 $t_{i,j}$。

在后向加密性方面：①新加入的成员 P_{n+1} 只能获取 $T_{n+1,2}(r)$ 和 $S(t_{n+1,2})$ 而无法“虚构”出原本不存在的“$T_{n+1,1}(r)$ 和 $S(t_{n+1,1})$”；②由于 Hash 函数良好的单向性，P_{n+1} 无法由 $h_{n+1,2}=Seed_{n+1}=H(h_{n+1,1})$ 恢复出所谓的“$h_{n+1,1}$”，P_{n+1} 对“$u_{n+1,1}$”的求解依赖于 $h_{n+1,1}$ 以及 $T_{n+1,1}(r)$：$u_{n+1,1}=(T_{n+1,1}(r)-h_{n+1,1})/C(r)$，从而 $u_{n+1,1}$ 也无法求出，又由于二元单向函数 f 的良好特性，因此 $t_{n+1,1}$ 更难求出；③P_{n+1} 也无法获取 $S(t_{n+1,1})$ 和前一次的密钥更新因子 σ_1。总之，P_{n+1} 恢复出更新前的组密钥 P_i 是难以想象的。

由于关于成员身份 r 的多项式 $C(r)$ 更新为 $I'(r)=I(r)\cdot(r-r_l)$，非法成员 P_l 就无法通过 $u_{i,2}=(T_{i,2}(r)-h_{i,2})/I'(r)$ 来求得 $u_{i,2}$（因为对 P_l 而言，$I'(r)=0$），从而就无法求得 $t_{i,2}=f(u_{i,2},h_{i,2})$，也就无法获得更新后的组密钥 K_g'，这显然很好地满足了前向加密性。

该方案中每个成员是独立地验证计算组密钥，系统针对非法成员更新了多项式 $I(r)$，这在求取 $u_{i,1}$ 的问题上设置了障碍，而且组密钥的获取还由成员 P_i 自身存储的前一次的哈希值 $h_{i,1}$（或初始的种子 $Seed_i$）和密钥更新因子 σ_2 所决定，所以也不存在同谋破解问题。

4.2 效率分析

为评价 HFMKM 方案的性能，我们从更新开销、计算开销和存储开销三方面，将其与 GKMP 方案进行对比分析。其中，更新开销指的是密钥更新时在网络上所传输的更新消息数量；计算开销指的是为了达到密钥更新而所需要的计算次数或计算时间；存储开销指的是密钥管理方案需要的存储空间大小。具体如表 2、表 3 所列。

表 2 组密钥更新

	GKMP	HFMKM
更新开销(个)	n	n
GC 计算开销	nC_e+C_r	$C_p+C_e+C_r$
成员计算开销	C_e	C_e+C_s

(下转第 65 页)

因是在提取水印时需要利用在嵌入水印时生成的两个决定图像几何结构的正交矩阵，而奇异值只是决定了特征图像在重构图像中的权重。因此为了克服该类算法存在较高的虚警率问题，后续的水印算法设计应另辟蹊径，而不应再沿用这类算法思想。改进的水印算法采用量化的嵌入方式将预处理后的水印图像嵌入原始图像经 DWT 和 SVD 分解后的奇异值中，提取时不需要原始水印的参与，是一种盲水印技术。实验表明，改进算法解决了典型的 SVD 水印算法过高虚警率的问题，具有较高的安全性、透明性及抗攻击的能力。

参 考 文 献

[1] 熊祥光，杨锦尊，崔巍，等. 基于三维小波变换和 HVS 的视频水印算法[J]. 武汉大学学报：工学版，2010，43(3)：357-360

[2] Liu R，Tan T. An SVD-based watermarking scheme for protecting rightful ownership [J]. IEEE Transactions on Multimedia，2002，4(1)：121-128

[3] 张秋余，李凯，袁占亭. 基于混沌和 SVD-DWT 的稳健数字图像水印算法[J]. 计算机应用研究，2010，27(2)：718-720

[4] 周波，陈健. 基于奇异值分解的、抗几何失真的数字水印算法[J]. 中国图象图形学报，2004，9(4)：506-512

[5] Shieh J-M，Lou D-C，Chang M-C. A semi-blind digital watermarking scheme based on singular value decomposition [J]. Computer Standards and Interfaces，2006，28：428-440

[6] Bhatnagar G，Raman B. A new robust reference watermarking scheme based on DWT-SVD [J]. Computer Standards & Interfaces，2009，31：1002-1013

[7] Roman R. Comment on an SVD-based watermarking scheme for protecting rightful ownership[J]. IEEE Transactions on Multimedia，2007，9(2)：421-423

[8] 周鹏颖，沈磊，田小林，等. 基于小波—奇异值分解的数字水印新算法[J]. 计算机应用研究，2010，27(5)：1896-1897

(上接第 52 页)

表 3 存储开销

	GKMP	HFMKM
GC 存储开销	$(n+1)K$	$(N+1)K+C_p$
成员存储开销	$2K$	$2K+\frac{C_p}{n}$

其中，n 为组规模(成员数量)，C_e 为一次加/解密的计算开销，C_r 为 GC 生成一个新密钥的计算开销，C_p 为计算 $S(t_{i,j})$和 $T_{i,j}(r)$组成的数据包的平均计算开销，C_s 为成员更新一次组密钥的计算开销，K 为密钥大小(比特)。

从以上分析可以看出，HFMKM 方案在更新开销和计算开销方面有一定优势；存储开销方面，组成员的开销稍微增加可以带来 GC 的存储开销的减少，从而在一定程度上减轻 GC 的负担。

4.3 方案优势

与典型的集中式组播密钥管理[4]不同，方案中 GC 与各个成员之间的关系不是简单的主次关系，组密钥由 GC 以及每个合法成员各自对等地产生，而不是由 GC 产生后再由会话密钥加密分发给各成员，这样就避免了组密钥在传输过程中的泄漏。

对于每个组成员，其组密钥的生成依赖于自身存储的哈希值(或哈希种子)和密钥更新因子，融合了集中式与分布式组密钥管理的双重特性，从而极大地提高了安全性，非常适合于安全需求较高的组播通信。

在组密钥生成阶段和每个成员的加入过程中都仅使用一次会话密钥就可确保后续的安全组播通信，这也使得通信开销有所降低。对多项式求值的运算由 GC 完成，不给成员节点带来负担；在成员节点的运算开销方面，对哈希函数和二元单向函数的运算具备很好的单向性，而且其计算开销不大，带来更好的安全性，同时不增加过多的计算开销。

另外，删除成员的组密钥更新方法的一个显著优点是：可以很方便地一次性删除多个成员。例如，若一次性要删除的成员为 $P_1, P_2, \cdots, P_l$，则只需将 $I(r)$更新为 $I'(r)=I(r)\cdot(r-r_1)\cdot(r-r_2)\cdots(r-r_l)$即可。

结束语 组密钥管理是安全组播通信的前提。传统的许多集中式平面型组密钥管理方案将几乎所有计算、更新、传输等操作任务归于 GC，在带来管理方便的同时也存在一定风险。本文正是基于此，设计出另一种由各组成员与 GC 平等地生成组密钥的密钥管理模型，旨在探索出一种新的组密钥管理体制。本方案体现出集中式、分布式与验证性的统一，是一种新型的、安全可靠的集中式平面型组播密钥管理方案。当然，如何将 GC 的各项开销继续缩小以及如何优化对数据的分组接收模式，是需要进一步研究的问题。

参 考 文 献

[1] 王琳，解冲锋，杨明川. IP 组播的关键技术 [J]. 信息网络，2003(01)：28-33

[2] 赵膺，宋佳兴，徐万鸿，等. 安全组播综述 [J]. 小型微型计算机系统，2003，24(10)：1873-1877

[3] Eskicioglu A M. Multimedia security in group communications：Recent progress in key management，authentication，and watermarking [J]. ACM Multimedia Systems，Special Issue on Multimedia Security，September 2003：239-248

[4] Harney H，Muckenhirn C. Group key management protocol (GKMP) specification[S]. RFC2093. 1997

[5] Suvo M. Iolus：A framework for scalable secure multicasting [J]. ACM SIGCOMM Computer Communication Review，1997，27(4)：277-288

[6] 李国民，何大可. 基于身份的认证群密钥协商协议 [J]. 计算机科学，2009，36(01)：60-64

[7] 孙海波，张权，唐朝京. 基于密钥矩阵的组播密钥管理方案 [J]. 计算机工程，2008，34(21)：112-114

[8] 柳秀梅，周福才，常桂然，等. 使用双线性配对实现组播密钥管理协议 [J]. 东北大学学报：自然科学版，2009，30(08)：1119-1123

[9] 吕远方. 基于秘密共享的无线传感器网络组密钥管理方案 [J]. 微计算机应用，2010，31(03)：35-41

[10] 许建真，董永先，梁克会. 一种高效的动态组播密钥管理方案 [J]. 计算机应用研究，2010，27(03)：1061-1063