三级对称叛徒追踪新方案

苏加军 王新梅
(西安电子科技大学综合业务网国家重点实验室 西安 710071)

摘 要 提出了一种新的对称叛徒追踪方案。基于加密广播技术和Hash函数理论,构造了密钥方案、加密方案、解密方案和叛徒追踪算法,利用Chernoff界确定了系统参数值。新方案可以有效对抗加密广播业务中的共谋密钥攻击。相对于已有的CFN对称方案,新方案具有更低的个人密钥存储复杂度、用户计算复杂度和更少的数据冗余。

关键词 加密广播,叛徒追踪,共谋密钥攻击,Chernoff界,Hash函数
中图法分类号 TN918,TP393 **文献标识码** A

New Three-level Symmetry Scheme of Traitor Tracing

SU Jia-jun WANG Xin-mei
(State Key Lab. of Integrated Service Network, Xidian University, Xi'an 710071, China)

Abstract This paper presented a new symmetry scheme of traitor tracing. Based on the broadcast encryption techniques and Hash function theory, the corresponding key scheme, encryption scheme, decryption scheme and traitor tracing algorithm were constructed in this paper. The proper values of system parameters were obtained by the Chernoff bound. The scheme in this paper has the advantages of lower storing complexity of personal key, lower user computing complexity and less data redundancy than existing CFN symmetry schemes and it can efficiently combat against the collusion key attack in the broadcast encryption.

Keywords Broadcast encryption, Traitor tracing, Collusion key attack, Chernoff bound, Hash function

1 引言

在加密广播业务(如付费电视系统)中,数据供应商DS(Data Supplier)经常面临两种典型的安全威胁:(1)共谋密钥攻击,即部分授权用户利用他们的解密密钥合谋构造非法解密盒;(2)重放攻击,即授权用户(单个或多个)通过秘密网络非法发布有用信息。通常,把这两种攻击中的授权用户称为叛徒(叛逆者)。叛徒追踪方案就是DS为保护自己的合法权益(版权或机密信息),用来追踪及识别叛徒的有效措施。

1994年,依据A. Fiat等提出的如何保护加密广播中个人密钥的问题[1],B. Chor等在文献[2]中首次提出了叛徒追踪的概念,并在Hash函数存在及大整数难以素分解的基础上,构造了一级和二级对称叛徒追踪方案。1998年,M. Naor等在文献[3]中构造了门限叛徒追踪方案。同年,K. Kurosawa等在文献[4]中构造了非对称叛徒追踪方案。1999年,D. Boneh等在文献[5]中构造了改进的非对称叛徒追踪方案。这些方案都可以对抗至多k个叛徒的共谋密钥攻击,并可以追踪识别至少一个叛徒。1999年,通过引入实时反馈信道,并利用水印技术,A. Fiat等在文献[6]中首次提出了对抗重放攻击的动态叛徒追踪方案。2000年,基于动态叛徒追踪方案,R. Safavi-Naini等在文献[7]中提出了连续追踪方案。这两种方案可以对抗重放攻击并动态追踪识别所有参与重放攻击的叛徒。最近几年,随着多媒体广播业务需求的日益增加,广播加密和动态叛徒追踪再次成为信息安全业界关注的焦点。与此同时,一些新奇的构造思路也陆续出现。比如Mu Ning-bo等于2009年提出了一种基于RSA的广播加密方案[9],而Yong-Dong Wu等于2010年提出了一种基于多密钥策略的叛徒追踪方案[10]。

在构造有效的叛徒追踪方案时,如何降低个人密钥存储复杂度、数据冗余复杂度和用户计算复杂度3个性能参数的值是一个关键问题。基于文献[2]中构造对称叛徒追踪方案的基本思想,本文利用Hash函数类[8]理论构造了具有三级结构的密钥方案,并结合加密广播技术,给出了完整的三级对称叛徒追踪方案。

2 三级叛徒追踪方案

一个完整的叛徒追踪方案通常有以下4个组成部分:用户初始化方案(密钥方案)、加密方案、解密方案、追踪算法。在构造追踪方案时,我们将以Hash函数类[8]作为基本工具来构造密钥方案。Hash函数的作用体现在两个方面:一是为每个授权用户选择合理的个人密钥;二是降低系统子密钥集中的元素数目,降低授权分组EB的数据冗余。假定广播系统用户容量为n,系统中可能出现的合谋叛徒数目上限为k。

2.1 三级结构的密钥方案

(1)系统子密钥集

到稿日期:2012-10-25 返修日期:2013-06-03 本文受国家973计划项目(2012CB316103)资助。
苏加军(1978—),男,博士生,主要研究方向为信息安全、信道编码和无线通信,E-mail:jiajun.su@hotmail.com;王新梅(1937—),男,教授,博士生导师,主要研究方向为信道编码和密码。

一级和二级结构密钥方案的构造可以参见文献[1,2]，本文将直接构造三级密钥方案。

在构造三级密钥方案时，数据供应商 DS 需要随机选取 3 个 Hash 函数集 $HS1$、$HS2$、$HS3$，要求如下：

r 个 Hash 函数构成集合 $HS1=\{h_1,h_2,\cdots,h_r\}$，其中每个 Hash 函数满足：

$$\{1,2,\cdots,n\}\xrightarrow{h_i}\{1,2,\cdots,N_1\} \tag{1}$$

$r\times s$ 个 Hash 函数构成集合 $HS2=\{g_{i,1},\cdots,g_{i,s}\}(1\leqslant i\leqslant r)$，其中每个 Hash 函数满足：

$$\{1,2,\cdots,n\}\xrightarrow{g_{i,j}}\{1,2,\cdots,N_2\} \tag{2}$$

$(1\leqslant i\leqslant r,1\leqslant j\leqslant s)$

$r\times s\times t$ 个 Hash 函数构成第 3 个集合 $HS3=\{f_{i,j,1},\cdots,f_{i,j,t}\}(1\leqslant i\leqslant r,1\leqslant j\leqslant s)$，其中每个 Hash 函数满足：

$$\{1,2,\cdots,n\}\xrightarrow{f_{i,j,e}}\{1,2,\cdots,N_3\} \tag{3}$$

$(1\leqslant i\leqslant r,1\leqslant j\leqslant s,1\leqslant e\leqslant t)$

r,s,t,N_1,N_2,N_3 为待定系统参数，其值将在第 3 节给出。

若系统选择 $r\times s\times t\times N_1\times N_2\times N_3$ 个随机数 $a_{i,j,e,h_i(u_1),g_{i,j}(u_2),f_{i,j,e}(u_3)}(1\leqslant i\leqslant r,1\leqslant j\leqslant s,1\leqslant e\leqslant t,u_1,u_2,u_3\in\{1,2,\cdots,n\})$作为子密钥，并按照如下方式排列即构成系统的三级子密钥集合。

$$BKS=\begin{Bmatrix}A^1_{1,h_1(1)} & \cdots & A^1_{1,h_1(n)}\\ \cdots & \cdots & \cdots\\ A^1_{r,h_r(1)} & \cdots & A^1_{r,h_r(n)}\end{Bmatrix}_{r\times n}$$

$$A^1_{i,h_i(u_1)}=\begin{Bmatrix}A^2_{i,1,h_i(u_1),g_{i,1}(1)} & \cdots & A^2_{i,1,h_i(u_1),g_{i,1}(n)}\\ \cdots & \cdots & \cdots\\ A^2_{i,s,h_i(u_1),g_{i,s}(1)} & \cdots & A^2_{i,s,h_i(u_1),g_{i,s}(n)}\end{Bmatrix}_{s\times n}$$

$$A^2_{i,j,h_i(u_1),g_{i,j}(u_2)}=\begin{Bmatrix}a_{i,j,1,h_i(u_1),g_{i,j}(u_2),f_{i,j,1}(1)} & \cdots & a_{i,j,1,h_i(u_1),g_{i,j}(u_2),f_{i,j,1}(n)}\\ \cdots & \cdots & \cdots\\ a_{i,j,t,h_i(u_1),g_{i,j}(u_2),f_{i,j,t}(1)} & \cdots & a_{i,j,t,h_i(u_1),g_{i,j}(u_2),f_{i,j,t}(n)}\end{Bmatrix}_{t\times n}$$

(2)用户个人密钥 $P(u)$

在三级追踪方案中，任何用户 $u\in\{1,2,\cdots,n\}$ 的个人密钥 $P(u)$ 所包含的子密钥的个数都为 $r\times s\times t$。即

$$P(u)=\left\{\begin{Bmatrix}a_{1,1,1,h_1(u),g_{1,1}(u),f_{1,1,1}(u)} & \cdots & a_{1,1,t,h_1(u),g_{1,1}(u),f_{1,1,t}(u)}\\ \cdots & \cdots & \cdots\\ a_{1,s,1,h_1(u),g_{1,s}(u),f_{1,s,1}(u)} & \cdots & a_{1,s,t,h_1(u),g_{1,s}(u),f_{1,s,t}(u)}\end{Bmatrix}_{s\times t}\right.$$
$$\cdots$$
$$\left.\begin{Bmatrix}a_{r,1,1,h_1(u),g_{r,1}(u),f_{1,1,1}(u)} & \cdots & a_{r,1,t,h_1(u),g_{r,1}(u),f_{r,1,t}(u)}\\ \cdots & \cdots & \cdots\\ a_{r,s,1,h_1(u),g_{r,s}(u),f_{r,s,1}(u)} & \cdots & a_{r,s,t,h_1(u),g_{r,s}(u),f_{r,s,t}(u)}\end{Bmatrix}_{s\times t}\right\}$$

2.2 加密方案

系统广播信息时，依然采用通用的加密广播解决方案[1]，即每次广播的信息组由授权分组 EB 和密文分组 CB 两部分构成。

由于保证安全广播的关键是非授权用户无法获取信息主密钥 MK，因此 MK 的生成方式是非常重要的。在三级叛徒追踪方案中，DS 可以随机选取 $r\times s\times t$ 个随机独立密钥份额 $S_{i,j,e}(1\leqslant i\leqslant r,1\leqslant j\leqslant s,1\leqslant e\leqslant t)$以如下方式来合成 MK：

$$\begin{aligned}&MK=XOR_{i=1}^{r}S_i\\&S_i=XOR_{j=1}^{s}S_{i,j}\\&S_{i,j}=XOR_{e=1}^{t}S_{i,j,e}\end{aligned} \tag{4}$$

(1)生成 EB：每个 $S_{i,j,e}$ 在 N_3 个不同的子密钥 $a_{i,j,e,h_i(u_1),g_{i,j}(u_2),f_{i,j,e}(u_3)}(u_3\in\{1,2,\cdots,n\})$作用下生成 N_3 个 EB 子分组 $E(s_{i,j,e},a_{i,j,e,h_i(u_1),g_{i,j}(u_2),f_{i,j,e}(u_3)})$，($E()$表示系统选定授权分组加密算法，后面用到的 $D()$表示授权分组解密算法)。显然，通过加密 $S_{i,j,e}$ 而生成的 EB 子分组的个数为 $N_1\times N_2\times N_3$。在完整的 EB 中通过加密随机密钥份额 $S_{i,j,e}$ $(1\leqslant i\leqslant r,1\leqslant j\leqslant s,1\leqslant e\leqslant t)$而生成的子分组总数为 $r\times s\times t\times N_1\times N_2\times N_3$。

(2)生成 CB：密文分组由明文分组 m 在主密钥 MK 下加密生成，即 $CB=E'(m,MK)$($E'()$表示密文分组加密算法，$D'()$表示密文分组解密算法)。

2.3 解密方案

(1)用户 u 首先利用个人密钥 $P(u)$解密 EB 中对应的$r\times s\times t$ 个子分组获取所有的

$$\begin{aligned}s_{i,j,e}=D(&E(s_{i,j,e},a_{i,j,e,h_i(u),g_{i,j}(u),f_{i,j,e}(u)}),\\&a_{i,j,e,h_i(u),g_{i,j}(u),f_{i,j,e}(u)})\end{aligned} \tag{5}$$

(2)计算信息主密钥 MK。

$$\begin{aligned}&S_{i,j}=XOR_{e=1}^{t}S_{i,j,e}\\&S_i=XOR_{j=1}^{s}S_{i,j}\\&MK=XOR_{i=1}^{r}S_i\end{aligned} \tag{6}$$

(3)利用 MK 解密密文分组 CB 获取明文信息 $m=D'(CB,MK)$。

2.4 叛徒追踪算法

数据供应商一旦捕获某个非法解密盒，利用黑盒测试方法[2,5]就可以获取该解密盒中的非法个人密钥 F。如果该解密盒可以正常工作，那么为了获取 MK，F 中至少包含 $r\times s\times t$ 个子密钥。分级追踪算法如下：

(1)统计可疑用户的三级子密钥数量，并标识可疑的二级子密钥矩阵。

若 $A^2_{i,j,h_i(u_1),g_{i,j}(u_2)}\cap F=SK$(第三级上的可疑子密钥的集合)，则系统标识出 $P(u)\cap SK\neq\varnothing$ 的那些用户 u。为了减少可疑用户的数量并实现更高的追踪准确度，我们采取如下三级标识策略：标识出满足 $|A^2_{i,j,h_i(u),g_{i,j}(u)}\cap F|_{\max}$ 的用户 u(可以解密出秘密份额)和对应的 $A^2_{i,j,h_i(u),g_{i,j}(u)}$。

(2)统计可疑用户的二级子密钥矩阵数量，并标识可疑的一级子密钥矩阵。

对于每一个 $A^2_{i,j,h_i(u),g_{i,j}(u)}$ $(1\leqslant i\leqslant r,1\leqslant j\leqslant s,u\in\{1,2,\cdots,n\})$，系统重复(1)标识过程。于是可以得到可疑用户二级子密钥矩阵的数量统计表，并采取如下二级标识策略：对于特定的 i 和用户 u，统计可疑的 $A^2_{i,j,h_i(u),g_{i,j}(u)}$，并标识出满足 $|\{A^2_{i,j,h_i(u),g_{i,j}(u)}:1\leqslant j\leqslant s\}|_{\max}$的用户 u 和对应的 $A^1_{i,h_i(u)}$。

(3)统计可疑的一级子密钥矩阵数量，并标识可疑用户。

对于每一个 $A^1_{i,h_i(u)}$ $(1\leqslant i\leqslant r,u\in\{1,2,\cdots,n\})$，系统重复(2)的标识过程。在得到可疑用户的一级子密钥矩阵的数量统计表后，采取如下一级标识策略：统计表中满足 $|A^1_{i,h_i(u)}:1\leqslant i\leqslant r|_{\max}$的用户 u 被标识，并被系统判定为合谋制作非法解密盒的叛徒。

3 方案的安全性证明及其性能

在新方案的构造过程中，方案的有效性可以由以下两点保证：加密广播方案实现了信息的保密传输；追踪算法必定能判定某个用户为叛徒。但是，如果方案的追踪算法把一个守法用户误判为叛徒，那么这种方案即使有效也是不可用的。所以，方案安全性的证明将基于如下思路：如果能够证明一个守法用户被误判为叛徒的概率足够小(可忽略不计)，那么这个方案就是实际安全的。在构造新方案的过程中，6 个影响着方案性能的系统参数(r,s,t,N_1,N_2,N_3)一直待定，我们也将在本节给出其值。

考虑如下情形：系统用户容量为 n；系统中共谋叛徒集 T 满足$|T|\leqslant k$；随机选取一个无辜用户 $u\in\{1,2,\cdots,n\}-T$；一个具有非法个人密钥 F 的解密盒被捕获。

(1)用户 u 因为 F 能够解密 EB 获取 $S_{i,j,e}(1\leqslant e\leqslant t)$而被诬陷的概率

由于 $HS1$、$HS2$、$HS3$ 的作用，因此与每个 $A^2_{i,j,h_i(u_1),g_{i,j}(u_2)}$ 相关联的叛徒数目为 $\log\log k$。则为了获取 $S_{i,j,e}(1\leqslant e\leqslant t)$，平均意义上每个叛徒至少需要提供 $t/\log\log k$ 个子密钥参与共谋。

定义一个随机变量 X_m，其满足：如果用户 u 因为能够解密 EB 获取 $S_{i,j,e}(1\leqslant e\leqslant t)$而被标识为可疑用户，则 $X_m=1$；否则，$X_m=0$。X_m 的概率分布为：

$$\begin{aligned}&\Pr(X_m=1)=\log\log k/N_3\\&\Pr(X_m=0)=1-\log\log k/N_3\end{aligned}\tag{7}$$

用户 u 被标识为可疑用户的次数超过均值 $t/\log\log k$ 的概率可由 Chernoff 界得到：

$$\Pr_1=\Pr\Big(\frac{1}{t}\sum_{m=1}^{t}X_m\geqslant\beta\cdot\log\log k/N_3\Big)<\Big(\frac{e^{\beta-1}}{\beta^{\beta}}\Big)^{t\cdot\log\log k/N_3}\tag{8}$$

又

$$\begin{aligned}&t\cdot\beta\cdot\log\log k/N_3=t/\log\log k\\&\Rightarrow N_3=\beta\cdot(\log\log k)^2\end{aligned}\tag{9}$$

取 $\beta=2$，即可得 $N_3=2(\log\log k)^2$，$\Pr_1<2^{-t/4\log\log k}$。

用户 u 因为 $A^2_{i,j,h_i(u),g_{i,j}(u)}$ 而被标识为可疑用户的次数达到最大值 $\log\log k$ 的概率为：

$$\begin{aligned}\Pr_1'&=\binom{\log k}{\log\log k}\cdot N_2^{-\log\log k}\\&\leqslant(e\log k/\log\log k)^{\log\log k}\cdot N_2^{-\log\log k}\end{aligned}\tag{10}$$

取 $N_2=2e\log k$，则

$$\begin{aligned}&\Pr_1\leqslant\Pr_1'=(2\log\log k)^{-\log\log k}\leqslant 1/\log k\\&\Rightarrow t\geqslant 4(\log\log k)^2\end{aligned}\tag{11}$$

(2)用户 u 因为 F 能够解密 EB 获取 $S_{i,j}(1\leqslant j\leqslant s)$而被诬陷的概率

定义一个随机变量 Y_m，其满足：如果用户 u 因为能够解密 EB 获取 $S_{i,j}(1\leqslant j\leqslant s)$而被标识为可疑用户，则 $Y_m=1$；否则，$Y_m=0$。Y_m 的概率分布为：

$$\begin{aligned}&\Pr(Y_m=1)=\log k/N_2\\&\Pr(Y_m=0)=1-\log k/N_2\end{aligned}\tag{12}$$

则用户 u 因为 $A^1_{i,h_i(u)}$ 而被标识为可疑用户的次数超过均值 $s/\log k$ 的概率可由 Chernoff 界得到：

$$\Pr_2=\Pr\Big(\frac{1}{s}\sum_{m=1}^{s}Y_m\geqslant\beta\cdot\log k/N_2\Big)<\Big(\frac{e^{\beta-1}}{\beta^{\beta}}\Big)^{s\cdot\log k/N_2}\tag{13}$$

取 $\beta=2$，并代入 $N_2=2e\log k$，则 $\Pr_2<2^{-s/4e}$。

用户 u 因为 $A^1_{i,h_i(u)}$ 而被标识为可疑用户的次数达到最大值 $\log k$ 的概率为：

$$\Pr_2'=\binom{k}{\log k}^{\log k}\cdot N_1^{-\log k}\leqslant(ek/\log k)^{\log k}\cdot N_1^{-\log k}\tag{14}$$

取 $N_1=ek$，则 $\Pr_2'<1/16k(k>4)$。

$$\Pr_2\leqslant\Pr_2'\Leftrightarrow 2^{-s/4e}\leqslant 1/16k\Rightarrow s\geqslant 4e\log 16k\tag{15}$$

(3)用户 u 因为 F 能够解密 EB 获取 $S_i(1\leqslant i\leqslant r)$而被诬陷的概率

定义一个随机变量 Z_m，其满足：如果用户 u 因为能够解密 EB 获取 $S_i(1\leqslant i\leqslant r)$而被标识为可疑用户，则 $Z_m=1$；否则，$Z_m=0$。Z_m 的概率分布为：

$$\begin{aligned}&\Pr(Z_m=1)=k/N_1\\&\Pr(Z_m=0)=1-k/N_1\end{aligned}\tag{16}$$

用户 u 因为 $S_i(1\leqslant i\leqslant r)$而被标识为可疑用户的次数超过均值 r/k 的概率可由 Chernoff 界得到：

$$\Pr_3=\Pr\Big(\frac{1}{r}\sum_{m=1}^{r}Z_m\geqslant\beta\cdot k/N_1\Big)<\Big(\frac{e^{\beta-1}}{\beta^{\beta}}\Big)^{r\cdot k/N_1}\tag{17}$$

取 $\beta=2$，并代入 $N_1=ek$，则 $\Pr_3<2^{-r/2e}$。

为了保证方案的安全性，必须要求 n 个用户中任意 k 个的叛徒无法诬陷任意一个无辜用户，即

$$\binom{n}{1}\cdot\binom{n}{k}\cdot\Pr_3<1\Leftrightarrow 2^{r/2e}>n\cdot n^{k-1}\Leftrightarrow r>2ek\log n\tag{18}$$

考虑 6 个系统参数的整数性要求，参数取值如下：

$r=\lceil 2ek\log n\rceil$，$s=\lceil 4e\log 16k\rceil$

$t=\lceil 4(\log\log k)^2\rceil$，$N_1=\lceil ek\rceil$

$N_2=\lceil 2e\log k\rceil$，$N_3=\lceil 2(\log\log k)^2\rceil$

取定上述参数之后，则有 $\Pr_1<2^{-\log\log k}$，$\Pr_2<2^{-\log 16k}$，$\Pr_3<2^{-k\log n}$。在追踪过程中守法用户是否被诬陷，$\Pr_3$ 起着最关键作用。依据式(19)可知，$\Pr_3$ 足够小是可以保证的，由此可以得到如下结论：基于所取的参数值，守法用户被诬陷为叛徒的可能性是可以忽略不计的。

对于一个叛徒追踪方案，通常有 3 个衡量方案性能的参数：个人密钥存储复杂度、数据冗余复杂度和用户计算复杂度。以上述 3 个参数为标准，给出了三级新方案与已有的一级、二级方案[2]的性能比较。

表 1 新方案与已有方案的性能比较：密钥存储复杂度

类型	个人密钥存储复杂度
一级	$O(k^2\log n)$
二级	$O(k^2\log^2 k\log(n/k))$
三级	$O(k\log n\log k\log^2\log k)$

表 2 新方案与已有方案的性能比较：数据冗余复杂度

类型	数据冗余复杂度
一级	$O(k^4\log n)$
二级	$O(k^3\log^4 k\log(n/k))$
三级	$O(k^2\log n\log^2 k\log^4\log k)$

表 3 新方案与已有方案的性能比较：用户计算复杂度

类型	用户计算复杂度
一级	$O(k^2\log n)$
二级	$O(k^2\log^2 k\log(n/k))$
三级	$O(k\log n\log k\log^2\log k)$

由表 1－表 3 可以看出，随着分级数目的增大，方案的性能也逐渐提高。对于个人密钥存储复杂度和解密盒计算复杂度，本文方案相对于已有二级方案有一个接近于 k 因子的优化；而对于 EB 中的数据冗余复杂度，本文方案相对于已有二

级方案有一个接近于 $k\log k$ 因子的优化。

结束语 共谋密钥攻击是加密广播业务中一种常见的攻击。本文提出了一种有效对抗该攻击的新方案，并基于加密广播技术和 Hash 函数理论，构建了三级结构的密钥方案、加密方案、解密方案和三级叛徒追踪算法。在利用 Chernoff 界论证方案安全性的过程中，给出了合理的系统参数值。对比已有的 CFN 对称方案，本文方案显著优化了 3 个主要性能参数：个人密钥存储复杂度、数据冗余复杂度和个人解密盒计算复杂度。新方案的构造方法具有较高的理论参考价值。利用具体的 Hash 函数和子密钥值实例化本方案，可以构建一个应用型的加密广播解决方案。

考虑到系统端的实际计算能力，多级方案中的级数必定存在实际上限。如何取得最优的实际上限是个值得深入研究的课题。

参 考 文 献

[1] Fiat A, Naor M. Broadcast encryption [A]// Advance in Cryptology-CRYPTO'93 [C]. vol. 773, Berlin, Germany: Springer-Verlag, 1994: 480-491

[2] Chor B, Fiat A, Naor M. Tracing traitors [A] // Advance in Cryptology-CRYPTO'94 [C]. vol. 839, Berlin, Germany: Springer-Verlag, 1994: 257-270

[3] Naor M, Pinkas B. Threshold traitor tracing [A]// CRYPTO'98 [C]. vol. 1462, Berlin: Springer-Verlag, 1998: 502-517

[4] Kurosawa K, Desmedt Y. Optimum traitor tracing and asymmetric scheme [A] // Advances in Cryptology-EUROCRYPT'98 [C]. vol. 1403, Berlin: Springer-Verlag, 1998: 145-157

[5] Boneh D, Franklin M. An efficient public key traitor tracing scheme [A] // CRYPT'99 [C]. vol. 1666, Berlin, Germany: Springer-Verlag, 1999: 338-353

[6] Fiat A, Tassa T. Dynamic traitor tracing [A]// Advance in Cryptology-CRYPTO'99 [C]. vol. 1999, Berlin, Germany: Springer-Verlag, 1666: 354-371

[7] Safavi-Naini R, Wang Ye-jing. Squential traitor tracing [A]// Advance in Cryptology-CRYPTO2000 [C]. vol. 1880, Berlin, Germany: Springer-Verlag, 2000: 316-332

[8] Goldreich O, Goldwasser S, Micali S. How to construct random functions [J]. J. Assoc. Comput, 1986, 33(4): 792-807

[9] Mu Ning-bo, Hu Yu-pu, Ou Hai-wen. Broadcast encryption schemes based on RSA [J]. The Journal of China Universities of Posts and Telecommunications, 2009, 16(1): 69-75

[10] Wu Yong-dong, Deng R H. A Multi-Key Pirate Decoder against Traitor Tracing Schemes [J]. Journal of Computer Science and Technology, 2010, 25(2): 362-374

(上接第 65 页)

表 4 基于分片迭代算法高负载动态迁移结果

内存(M)	迭代次数	停机时间(s)	总迁移时间(s)
128	30	3.917	31.296
256	30	3.756	44.284
512	29	3.872	62.246
1024	30	4.039	70.012
2048	29	4.078	91.259

高负载情况下的实验结果对比如图 5 和图 6 所示。

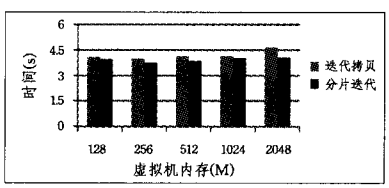

图 5 高负载时停机拷贝时间比对

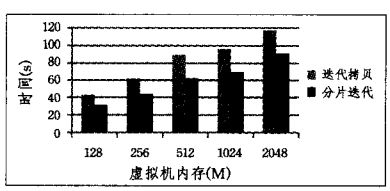

图 6 高负载时总迁移时间比对

在高负载情况下，比较而言，基于内存分片迭代拷贝机制不仅停机拷贝时间有一定的减少，而且总迁移时间也减少得较为明显，这是因为分片的迭代拷贝机制迭代过程中产生的脏页数小于额定脏页数的概率变大，从而加快了迭代拷贝的终止。

综上所述，在零负载或低负载情况下，基于分片内存迭代拷贝机制保持了原有 Pre-Copy 算法的性能，总的迁移时间略有改善；但在高负载情况下，内存分片迭代机制优越性明显，提升了虚拟机动态迁移的效率。

结束语 本文分析了 Xen 动态迁移中的内存迭代拷贝机制，提出了基于分片迭代的动态迁移机制，通过分析和实验验证了系统在不同负载环境下的迁移性能。下一步将继续优化 Xen 的动态迁移，将分片迭代拷贝中修改过于频繁的内存页直接放到停机拷贝的脏页中，提出缺页错误检测机制，以解决虚拟机迁移时因停机拷贝过长而导致缺页的问题。

参 考 文 献

[1] 刘铭，张炯，龙翔. 基于 Xen 虚拟机全系统在线增量迁移的设计与实现[J]. 微计算机信息，2010，26(10-3)：162-164

[2] Intel Corporation. System virtualization: principle & implementation[M]. Beijing: Tsinghua University Press, 2008

[3] 董耀祖，周正伟. 基于 X86 架构的系统虚拟机技术与应用[J]. 计算机工程，2006，32(13)：71-73

[4] 刘鹏程，陈榕. 面向云计算的虚拟机动态迁移框架[J]. 计算机工程，2010，36(5)：37-39

[5] 刘可超，李小勇. 基于 Xen 的虚拟磁盘调度算法改进[J]. 微型电脑应用，2010，26(4)：51-53

[6] Barham P, Dragovic B, Fraser K, et al. Xen and the art of virtualization[C]// Proceedings of the nineteenth ACM symposium on Operating Systems Principles (SOSP19). New York: ACM Press, 2003: 164-177

[7] 阮敏. Xen 环境下实时迁移结构和算法研究[D]. 大连：大连海事大学，2009

[8] 李建彬. 基于虚拟机的分布式容灾备份技术研究[D]. 长沙：国防科学技术大学，2010

[9] 高翔. 基于 Xen 的虚拟机动态迁移算法优化[D]. 深圳：哈尔滨工业大学，2010

[10] 付超. 基于 Excalibur 系统的 Xen 动态迁移优化与研究[D]. 北京：北京邮电大学，2011

[11] Clark C, Fraser K, Hand S, et al. Live migration of virtual machines[C]// Proceedings of the 2nd Conference on Networked Systems Design & Implementation. 2005: 273-286