Computer Science and Application 计算机科学与应用, 2013, 3, 297-301
http://dx.doi.org/10.12677/csa.2013.37052 Published Online October 2013 (http://www.hanspub.org/journal/csa.html)

Group Key Management Based on T-OFT*

Chao Xu, Hui Li, Di Liu

School of Computer Science, Beijing University of Posts and Telecommunications, Beijing
Email: jessia19891012@126.com, lihuill@bupt.edu.cn, ld_bupt@163.com

Received: Sep. 24th, 2013; revised: Oct. 19th, 2013; accepted: Oct. 27th, 2013

Abstract: A novel group key management protocol based on Ternary Tree and One-way Function (T-OFT) is proposed in this paper to avoid the problem about forward confidentiality, backward confidentiality and conspiracy attack. The ternary tree is used in the protocol which reduces the number of storing keys and lowers the cost of storage and communication. We also use TPM to generate and store keys to ensure no keys outside plainly, guaranteeing absolute security of keys. The group key will be renewed when group members join or quit in order to provide a safe key management module. The protocol overcomes the above defects and lowers the cost of storage and communication, and could guarantee the physical security of the key server.

Keywords: Multicast; Group Key Management; TPM; OFT; T-OFT

一种基于 T-OFT 的组密钥管理协议*

徐　超，李　晖，刘　迪

北京邮电大学计算机学院，北京
Email: jessia19891012@126.com, lihuill@bupt.edu.cn, ld_bupt@163.com

收稿日期：2013 年 9 月 24 日；修回日期：2013 年 10 月 19 日；录用日期：2013 年 10 月 27 日

摘　要：针对集中式组播密钥管理协议具有前向安全、后向安全、同谋破解等问题，本文提出了一种基于三叉树(Ternary Tree)的 OFT 组密钥管理协议(T-OFT)。使用三叉树的逻辑密钥结构，减少了密钥服务器存储密钥的数量，有效的降低了存储和通信开销。并借用可信安全模块(TPM)来产生和保存密钥信息，确保没有密钥信息显式的出现在 TPM 之外，保证了密钥的绝对安全。当组成员关系发生变化时，本协议通过更新组密钥保证前后向安全和防止同谋破解，提供了一种安全高效的组密钥管理服务。分析结果表明，该协议可以有效的降低存储和通信开销，并能保证密钥服务器的物理安全性。

关键词：组播；密钥管理；TPM；OFT；T-OFT

1. 引言

组播[1]是一种可以把一份报文同时发送给多个接收者的有效通信方式。与单播通信方式相比，组播通信有效的减少发送者的资源开销和节约网络带宽等优势。但是目前的组播协议缺乏可靠的安全机制，采用明文传输的组播报文方式在网络上很容易被篡改、监听和冒充。对组播报文加密传输是实现组播保密性和完整性的一种有效的方法。加解密所用的密钥只有组成员可以获得，这样确保了被加密的报文只有组成员才能解读。与单播的密钥管理方式相比，组播方式面临更多挑战，如前向安全(Forward confidentiality)、后向安全(Backward confidentiality)、同谋破解(Conspiracy attack)等是组播密钥管理特有的问题。

*国家自然科学基金资助项目(61070207)。

前向安全要求离开后的组成员无法继续参与组播，即无法利用它所知道的密钥解密后继组播报文。后向安全要求新加入的组成员无法破解它加入之前的组播报文。当组成员退出或者加入时，更新组密钥可以实现前向和后向安全，但是更新密钥时不仅要防止某个已离开的组成员破解新的组密钥，还要防止多个已离开的组成员联合起来破解，这就涉及到同谋破解问题。同谋破解是指几个组成员恶意联合，掌握足够多的信息，使得新的组密钥被破解，导致组播管理的前向安全和后向安全失败。为了解决这些问题，近年来提出的一些组密钥管理协议大多有更新密钥通信代价过大、安全性不高等问题，如 GKMP (Group key Management Protocol)集中控制式组密钥管理方案[2-5]，此方案的密钥更新方式无法做到前向安全，且密钥服务器遭到物理攻击后，容易泄露密钥等重要信息。基于树的 LKH (Logical Key Hierarchy)密钥管理方案[6-8]中，密钥更新的通信等代价较大，也没有解决密钥的可信产生、可信存储等问题。基于 OFT (One-way Function Tree)的组密钥管理协议[9-11]，有效的减少了的密钥更新产生的报文，但是 OFT 协议无法抵抗同谋破解，且没有解决密钥的安全存储问题，还有一定的改进空间。为此本文提出了一种安全高效的组播密钥管理协议 T-OFT (One-way Function Tree based Ternary Tree)。其安全性主要体现在借用可信安全模块(TPM)来安全的产生和保存密钥等信息，确保没有密钥信息显式的出现在 TPM 之外，保证了密钥的绝对安全；而其高效性体现在采用三叉树(Ternary Tree)的逻辑密钥结构，减少了密钥服务器存储密钥的数目，有效的降低了存储开销，且由于组成员的加入和退出导致的密钥更新代价为 $O(\log_3 n)$，降低了系统的通信代价。

可信安全模块(TPM)[12-14]是指在当前计算机架构上添加硬件模块及相应的软件，以构建一个操作系统体系之外的计算机安全平台，从而从根本上解决计算机的安全隐患。TPM 可以提供随机数生成、加解密算法封装、密封存储、哈希函数、证书操作等功能，为许多对物理安全性要求较高的领域提供安全可信的保证。将 TPM 和 T-OFT 密钥管理结合起来，有效的解决了对密钥信息的安全存储和计算密钥时保证没有密钥显式出现在 TPM 之外，做到了密钥不出卡，保证了密钥服务器的物理安全性。

2. 基于 T-OFT 的组密钥管理协议

由于缺乏有效的权限控制和身份认证机制，组播通信方式使得任何用户都可以接收和发送组播报文，有效的限制方式是加密组播敏感数据，使得只有组成员即拥有组播密钥的成员才能解密获得数据内容。组播密钥管理协议就是负责为组成员生成和下发密钥，并在有成员加入和离开时更新密钥，以保证系统的前向安全和后向安全。组密钥本身的安全性至关重要，对组密钥长度、随机性、也有一定的要求，这里我们用 TPM 生成随机数用于生成组密钥。

2.1. T-OFT 结构

为了解决组密钥管理中密钥更新代价大以及密钥的安全生成及存储问题，本文提出了一种基于 T-OFT 组密钥管理协议，下面介绍协议里需要用到了函数及其逻辑密钥树结构，逻辑密钥树结构我们采用三叉树(Ternary Tree)，与传统二叉树相比，可以减少密钥服务器(KS)存储的密钥数量，有效的降低了系统的存储开销。

- 单向函数 $h(x)$：x 通常为密钥，密钥 通过单向函数 $h(\ )$ 的隐藏了原始密钥的内容，这样 $h(x)$ 虽然携带密钥信息，但是无法计算得到 ，$h(x)$ 在组内共享对密钥 没有安全隐患，在本协议中我们用可信安全模块(730)中提供的哈希函数作为单向函数 $h(x)$。
- 普通混淆函数 $f(a,b,c)$：该函数输入为三个数 a、b、c，经过函数 $h(\ )$ 得到一个值 $f(a,b,c)$，无法从 $f(a,b,c)$ 的值退出 a、b 或 c 的值。
- 辅助密钥 k_H：用于生成组密钥，防止同谋破解
- 图 1 为高度 $h=2$ 的 T-OFT 逻辑密钥树，各成员可以通过下面的公式计算得到各个节点的密钥：

$$k_i = f\left(h\left(k_{l(i)}\right), h\left(k_{c(i)}\right), h\left(k_{r(i)}\right)\right) \qquad (1)$$

其中 $k_{l(i)}, k_{c(i)}$和$k_{r(i)}$ 是节点 k_i 的左中右孩子节点密钥，我们称 $h(k)$ 为 k 节点的单向密钥信息，公式中的密钥以及单向函数都是在 TPM 中计算完成的，保证了密钥不会在计算过程中暴露在内存里。

在 T-OFT 结构中，各个节点的密钥都是相互依赖的。每个组成员拥有一定的单向密钥信息。根据以上公式可以计算出各个节点的密钥分别为：

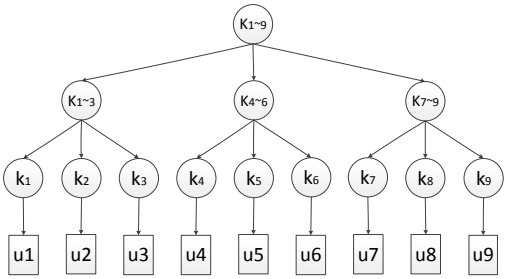

Figure 1. T-OFT logical key tree
图 1. T-OFT 逻辑密钥树

$$k_{1\sim3} = f\left(h(k_1), h(k_2), h(k_3)\right)$$

$$k_{4\sim6} = f\left(h(k_4), h(k_5), h(k_6)\right)$$

$$k_{7\sim9} = f\left(h(k_7)h(k_8), h(k_9)\right)$$

从而得到密钥为：

$$k_{1\sim9} = f\left(h(k_{1\sim3}), h(k_{4\sim6}), h(k_{7\sim9})\right)$$

最后求出组密钥：

$$K = h(k_{1\sim9}, k_H) \tag{2}$$

如组成员u1拥有其兄弟节点的单向密钥信息以及其祖先的兄弟的单向密钥信息，即 $h(k_2)$、$h(k_3)$、$h(k_{4\sim6})$、$h(k_{7\sim9})$ 以及自己的节点密钥 k_1，就可以求得组密钥 $K_{1\sim9}$

$$k_{1\sim3} = f\left(h(k_1), h(k_2), h(k_3)\right)$$

$$k_{1\sim9} = f\left(h(k_{1\sim3}), h(k_{4\sim6}), h(k_{7\sim9})\right)$$

$$K = h(k_{1\sim9}, k_H)$$

这里，密钥服务器和组成员都是通过计算获得组密钥 K。

最后，对协议中用到的一些符号做一个定义和说明：

$-\!>$：单播

$=\!>$：组播

$<\!=\!>$：通过安全信道

$-\gg$：请求TPM生成随机数

U_i：第 i 个组成员

$\{M\}_k$：用密钥 k 给 M 加密

2.2. 密钥生成与下发

密钥管理包括安全的生成和下发密钥、密钥信息的安全存储以及安全的更新密钥等工作。密钥生成与分配主要是指密钥服务器如何安全的生成密钥并将密钥分配给各个用户，以组成员数 $n=9$ 为例，主要流程如下：

1) KS：$\text{TPM}-\gg k_i$，$i=1,2,\cdots,9$

2) $\text{KS}<\!=\!>U_i$：$\{k_i\}$，$i=1,2,\cdots,9$

3) KS 在 TPM 中计算单向密钥信息 $h(k_i)$，$i=1,2,\cdots,9$，再根据 $h(k_i)$ 计算出 $h(k_{1\sim3})$、$h(k_{4\sim6})$、$h(k_{7\sim9})k_{1\sim9}$。

4) KS 给各个组成员下发单向密钥信息用于各自计算出密钥 $k_{1\sim9}$，如组成员 U_1 下发消息结构如下：$\text{KS}-\!>U_1$：$\{h(k_2)|h(k_3)|h(k_{4\sim6})|h(k_{7\sim9})\}_{k_1}$ 同理给其他组成员下发密钥信息，并用其 k_i 加密保护。

5) 各组成员收到单向密钥信息后，根据公式(1)求得密钥 $k_{1\sim9}$ 以及其对应的叶节点到根节点路径上的各个密钥。

6)KS：$\text{TPM}-\gg k_H$

7) $\text{KS}=\!>U_i$：$\{k_H\}_{k_{1\sim9}}$ $i=1,2,\cdots,9$

各成员根据公式(2)求得组密钥 K。

8) 最后 KS 请求其 TPM 封存 k_i，$h(k_i)$，$h(k_{1\sim3})$，$h(k_{4\sim6})$，$h(k_{7\sim9})$，k_H，K，$i=1,2,\cdots,9$，以保证系统的物理安全性。

这样，经过 1~8 步，就完成了各个组成员 k_i 密钥的生成和下发以及组密钥 K 的生成和下发。

2.3. 成员退出

当组成员 U_i 退出该组时，为了保证之后的组播信息不能被已退出的成员 U_i 通过它拥有的密钥解密获得信息即前向安全，必须对其拥有的密钥更新，以组成员 U_9 为例，U_9 离开后的逻辑密钥树如图 2 所示，此时假设存在节点 uv，其密钥为 k_v，由逻辑密钥树可以看出，需要更新的密钥是 $k_{7\sim9}$ 和组密钥 $k_{1\sim9}$，主要过程如下：

1) $U_9-\!>\text{KS}$：{quit request}

2) KS：$\text{TPM}-\gg k_v$ 并在 TPM 中计算出 $h(k_v)$ 和新的 $h(k_{7\sim9})$ 以替换旧的值。

3) 密钥服务器分别向组成员 u1~u3、u4~u6 以及 u7~u8 组播密钥更新消息，消息体如下：

$$KS=\!>\begin{cases} u1\sim u4:\{h(k_{7\sim9})\}_{k_{1\sim3}} \\ u5\sim u6:\{h(k_{7\sim9})\}_{k_{4\sim6}} \\ u7:\{h(k_v)\}_{k_7} \\ u8:\{h(k_v)\}_{k_8} \end{cases}$$

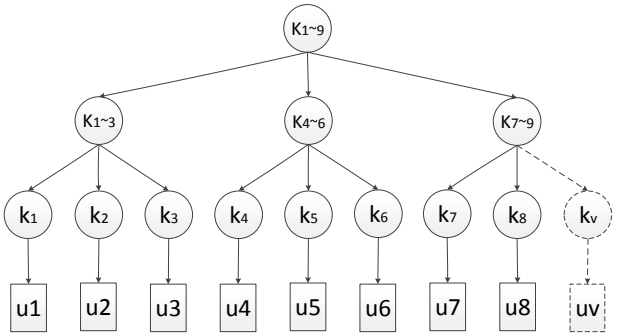

Figure 2. The group after new member u9 leaves
图 2. u9 离开后的 T-OFT 逻辑密钥树

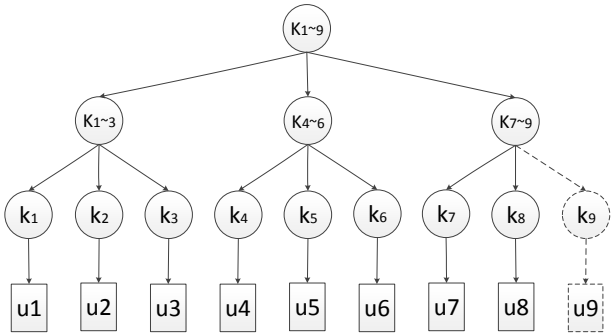

Figure 3. The group after new member u9 joins
图 3. u9 加入后的 T-OFT 逻辑密钥树

4) 各组成员收到单向密钥信息后，根据公式(1)求得新的密钥 $k_{1\sim9}$ 以及其对应的叶节点到根节点路径上的各个密钥。

5) 为抵抗同谋破解，KS：$\mathrm{TPM}-\gg k_H$

6) $\mathrm{KS}=>U_i$：$\{k_H\}_{k_{1\sim9}}\quad i=1,2,\cdots,8$

各成员根据公式(2)求得组密钥 K。

7) 最后 KS 请求其 TPM 封存 k_i，k_v，$h(k_i)$，$h(k_{1\sim3})$，$h(k_{4\sim6})$，$h(k_{7\sim9})$，k_H，K，$i=1,2,\cdots,8$，以保证系统的物理安全性。

通过以上过程，完成了前向安全的密钥更新，从该过程可以看出，密钥服务器需要组播的通信代价为 $2\log_3 n+1$，而前面提到的GKMP协议、基于树的LKH密钥管理方案中，都会产生大量的密钥更新报文，与之相比较，本协议大大降低了系统的资源开销和节约网络带宽，具有显著的优势。

2.4. 成员加入

当组成员 U_i 加入该组时，为了保证之前的组播信息不能被刚加入的成员 U_i 通过它拥有的密钥解密获得信息即后向安全，必须对旧的密钥进行密钥更新，即组密钥 K 以及其对应的叶节点到根节点路径上各个密钥。以组成员 U_9 为例，U_9 加入后的逻辑密钥树如图 3 所示，由逻辑密钥树可以看出，需要更新的密钥是 $k_{7\sim9}$ 和组密钥 K，主要过程如下：

1) $U_9->\mathrm{KS}$：{join request}

2) KS：$\mathrm{TPM}-\gg k_9$

3) $\mathrm{KS}<=>U_9$：$\{k_9\}$ 并在 TPM 中计算出新的 $h(k_{7\sim9})$、$h(k_9)$ 以替换旧的值。

4)密钥服务器分别向组成员 u1~u3、u4~u6 以及 u7~u8 组播密钥更新消息，消息体如下：

$$KS=>\begin{cases}u1\sim u4:\{h(k_{7\sim9})\}_{k_{1\sim3}}\\ u5\sim u6:\{h(k_{7\sim9})\}_{k_{4\sim6}}\\ u7:\{h(k_9)\}_{k_7}\\ u8:\{h(k_9)\}_{k_8}\end{cases}$$

5) 各组成员收到单向密钥信息后，根据公式(1)求得密钥 $k_{1\sim9}$ 以及其对应的叶节点到根节点路径上的各个密钥。

6) 为抵抗同谋破解，KS：$\mathrm{TPM}-\gg k_H$

7) $\mathrm{KS}=>U_i$：$\{k_H\}_{k_{1\sim9}}\quad i=1,2,\cdots,9$ 各用户根据公式(2)得到组密钥 K。

8) 最后 KS 请求其 TPM 封存 k_i，$h(k_i)$，$h(k_{1\sim3})$，$h(k_{4\sim6})$，$h(k_{7\sim9})$，k_H，K，$i=1,2,\cdots,9$。

通过以上过程，完成了后向安全的密钥更新，从该过程可以看出，密钥服务器需要组播的单向密钥信息为 $2\log_3 n+1$，降低了系统的资源开销和节约网络带宽，具有明显的优势。#

3. 协议分析

下面比较了几种组密钥管理协议在组成员动态变化，即成员加入或离开时，为保证系统的前后向安全进行密钥更新造成的通信代价，密钥服务器及用户存储密钥信息的存储开销，以及各个协议做到的安全指标，具体内容如下表 1 安全性能比较表和表 2 通信及存储开销比较表，其中 n 为组成员数，K 为密钥比特数。

由表 1 和表 2 可以看出，LKH 协议的在组成员动态变化时引起的通信代价和存储开销较大，且没有解决密钥的安全存储问题，而 GKMP 协议中离开的组成员仍然能够解密报文，无法做到前向安全，且密钥服

Table 1. Comparisons of security performance
表 1. 安全性能对比表

协议	安全			抵抗同谋破解
	前向	后向	存储安全	
LKH	Y	Y	N	Y
GKMP	N	Y	N	Y
OFT	Y	Y	N	N
T-OFT	Y	Y	Y	Y

Table 2. Comparisons of communication and storage cost
表 2. 通信及存储安全开销对比表

协议	通信开销		存储开销	
	成员加入	成员离开	KS	用户
LKH	$O(\log_2 n)$	$O(\log_2 n)$	$2(n-1)K$	$2(n-1)K$
GKMP	$4K$	-	$2k$	$2k$
OFT	$O(\log_2 n)$	$O(\log_2 n)$	$2(n-1)K$	$(\log_2 n+1)K$
T-OFT	$O(\log_3 n)$	$O(\log_3 n)$	$(3n+1)K/2$	$(2\log_3 n+1)K$

务器遭到物理攻击后，容易泄露密钥等重要信息，没有解决密钥的安全存储问题，存在多种问题。OFT 协议有效的减少了的密钥更新产生的报文，但是 OFT 协议无法抵抗同谋破解，且没有解决密钥的安全存储问题。而本文提出的 T-OFT 组密钥管理协议既能做到前后向安全、抵御同谋破解，同时做到了系统中密钥的可信生成和存储，计算密钥时保证没有密钥显式出现在 TPM 之外，做到了密钥不出卡，保证了密钥服务器的物理安全性。并且其密钥更新所需要的通信代价为$O(\log_3 n)$，存储开销在四种协议中最小。最后使用三叉树(Ternary Tree)的逻辑密钥结构，有效的降低了密钥服务器的存储代价，在组成员数量较大的情况下优势更加明显。

4. 总结

本文中，我们提出了一种基于 T-OFT 组密钥管理协议，借助于可信平台模块(TPM)，保证物理安全性和高效性，其安全性主要体现在借用 TPM 来安全的产生和保存密钥等信息，并确保没有密钥信息显式的出现在 TPM 之外，保证了密钥的绝对安全；而其高效性体现在由于组成员的加入和退出导致的密钥更新代价为$O(\log_3 n)$，与前人提出的密钥管理协议相比，降低了系统的通信代价。在存储开销方面，协议采用三叉树(Ternary Tree)的逻辑密钥结构，有效降低了存储开销，且在组成员较多的情况下优势更为明显。#

本文的特色主要可以归结为四点，首先该协议实现系统中密钥的可信生成和存储，计算密钥时保证没有密钥显式出现在 TPM 之外，做到了密钥不出卡，保证了密钥服务器的物理安全性。第二采用 OFT 模式计算出组密钥，组密钥无需在密钥服务器生成后下发，而是由各个组成员自行计算得到，避免了组密钥在传输过程被窃取。第三采用 Ternary Tree 的逻辑密钥结构，降低了密钥服务器存储密钥的数量，有效的降低了系统的存储开销。最后，该协议的密钥更新的通信代价为$O(\log_3 n)$，代价很小，为密钥管理的实施提供便利。

参考文献 (References)

[1] M. W. Xu, X. H. Dong and K. Xu. A survey of research on key management for multicast. Journal of Software, 2004, 15(1): 141-150.

[2] B. Jiang, X. Hu. A survey of group key management. International Conference of Science and Software Engineering, 2008, 3: 994-1002.

[3] S. A. Mortazavi, A. N. Pour and T. Kato. An efficient distributed group key management using hierarchical approach with Diffie-Hellman and Symmetric Algorithm: DHSA. International Symposium on Computer Networks and Distributed Systems (CNDS), 2011: 49-54.

[4] B. R. Purushothama, B. B. Amberker. Group key management scheme for simultaneous multiple groups with overlapped membership. IEEE 2011 Third International Conference on Communication Systems and Networks, 1-10.

[5] M. Hajyvahabzadeh, E. Eidkhani, S. A. Mortazavi and A. N. Pour. A new group key management protocol using code for key calculation: CKC. 2010 International Conference on Information Science and Applications (ICISA), 2010: 1-6.

[6] W. H. D. Ng, M. Howarth, Z. Sun and H. Cruickshank. Dynamic balanced key tree management on computers. 2007, 56(5): 590-605.

[7] Y.-R. Chen, W.-G. Tzeng. Efficient and provably-secure group key management scheme using key derivation. IEEE 11th International Conference on Trust, Security and Computing and Communications, 2012.

[8] R. Velumadhava Rao, K. Selvamani and R. Elakkiya. A secure key transfer protocol for group communication. Advanced Computing: An International Journal (ACIJ), 2012, 3(6).

[9] 王巍. 群组密钥管理的理论与关键技术研究[D]. 2008.

[10] M. Yasir. Efficient group key management schemes for multicast dynamic. Communication Systems, 2012.

[11] D. A. Mcgrew, A. T. Sherman. Key establishment in large dynamic groups using one-way function trees. Tech Rep No. 0755, TIS Labs at Network Associates, Inc., Glenwood.

[12] X. Chang, H. G. Zhang, D. G. Feng, Z. F. Cao and J. W. Huang. Survey of information security. Science in China Series F-Information Science, 2008, 50(3): 273-298.

[13] TCG Group. TCG architecture overview specification. 2004. https://www.trustedcomputeringgroup.org/home

[14] W. Z. Yang, Z. Y. Zhang and X. H. Wu. Internet technology and applications. International Conference on Digital Object Identifier, 2010: 1-4.