第 33 卷 第 8 期 计 算 机 工 程 2007 年 4 月
Vol.33 No.8 Computer Engineering April 2007
·安全技术· 文章编号：1000—3428(2007)08—0148—02 文献标识码：A 中图分类号：TP309

一个成员撤消算法的安全缺陷

王凤和[1,2]，胡予濮[2]，王春晓[3]
(1. 泰山学院数学系，泰安 271000；2. 西安电子科技大学计算机网络与信息安全教育部重点实验室，西安 710071；
3. 山东建筑工程学院数理系，济南 250014)

摘 要：2003 年陈等提出了一个高效的群成员撤消算法，与已有方案比较该方案无疑是高效的，但该文的研究表明陈的方案还存在较大的安全缺陷，针对这一缺陷给出了一种攻击方法，使得一个已经被撤销的群成员仍然可以利用这种攻击继续生成合法签名。从而说明该撤消算法是不安全的。

关键词：攻击；撤消；群签名；ACJT 方案

Security Flaw in Revocation Algorithm

WANG Fenghe[1,2], HU Yupu[2], WANG Chunxiao[3]
(1. Dept. of Mathematics, Taishan College, Taian 271000; 2. Key Lab of Computer Networks and Information Security, Ministry of Education, Xidian University, Xi'an 710071; 3. Dept. of Math. and Phys., Shandong Institute of Architecture and Engineering, Jinan 250014)

【Abstract】The paper analyzes the security of an efficient revocation algorithm which was proposed by Chen in 2003. Security flaw is identified in their scheme. The paper proposes an attack to this flaw, it shows a deleted member can forge valid group signature. So their scheme is insecure.

【Key words】Attack; Revocation; Group signature; ACJT scheme

群签名的概念由D.Chaum 和Van.Heyst提出[1]。群签名允许群成员代表群体生成匿名的签名。在有争议时可以由一个群管理员打开签名。群签名在许多场合有着广泛的应用，如电子现金系统和电子投票协议等。ACJT群签名[2]的提出是群签名领域的重要进展，它是迄今为止最好的群签名之一。

如何有效地实现群成员的撤消是群签名的一个重要问题。因为在实际中群组成员总是动态的：不仅会有新成员加入同时还有群成员的撤消问题。群组成员可能因为人事变动、岗位调整退出群组，或是由于不光彩的事被逐出群组。因此必须提供一个安全高效的群成员撤消算法使被撤消者不能再代表群体生成合法的签名。而已有的群撤消方案签名或验证算法多线性依赖群成员的个数或被撤消成员的个数[3~5]。当群组较大时效率较低。

陈泽文等基于ACJT群签名提出了一个新的群成员撤消方案[6]。与已有撤消方案[3~5]比较，文献[6]中的方案无疑是高效的：每次成员撤消群管理员仅需要作一次乘法运算，且签名和验证算法均独立于成员个数和撤消成员个数。但是本文的研究表明文献[6]中存在安全缺陷。它事实上不能防止已撤消成员继续生成合法的签名。从而他们的尝试是不成功的。

1 Chen 方案的回顾

本节简要介绍加入Chen-撤消功能后的ACJT群签名[6]。

1.1 互素性证明

设 n=pq，其中 $p=2p'+1, q=2q'+1$为素数，g, h 是 QR（n）中的元证明者利用以下方法在不泄露 e 的条件下证明 e 和 E 互素。

证明

(1)利用扩展的 GCD 算法得到 a, b 满足 $ae_i+bE=1$。

(2) $u,v,\omega\in[0,1]^{2l_p}$，计算：

$T_1=g^{\omega}(\bmod n), T_2=g^{e_i}h^{\omega}(\bmod n), T_3=g^{v}(\bmod n)$,

$T_4=T_3^{a}h^{v}(\bmod n), T_5=g^{u}(\bmod n)$,

$T_6=g^{Eb}h^{u}(\bmod n), T_7=h^{-(a\omega+u+v)}$,

随机地选择：$r_1,r_2,r_3,r_4,r_5,r_6,r_7\in\pm\{0,1\}^{\varepsilon(2l_p+k)}$

计算：

$R_1=g^{r_1}, R_2=g^{r_2}, R_3=g^{r_3}, R_4=h^{r_4}, R_5=g^{e}h^{\omega}, R_6=T_2^{r_6}, R_7=g^{Er_7}h^{r_3}$。

$c=h(g\|h\|E\|T_1\|T_2\|T_3\|T_4\|T_5\|T_6\|T_7\|R_1\|R_2\|\|R_3\|R_4\|R_5\|R_6\|R_7)$。

$s_1=r_1-c\omega, s_2=r_2-cv, s_3=r_3-cu, s_4=r_4+c(a\omega+u+v), s_5=r_5-ce, s_6=r_6-ca, s_7=r_7-cb$。

公开数组：

$(c,s_1,s_2,s_3,s_4,\ldots,s_7,T_1,T_2,T_3,T_4,\ldots,T_7)$

验证

(1)计算：

$c'=h(g\|h\|E\|T_1\|T_2\|T_3\|T_4\|T_5\|T_6\|T_7\|g^{s_1}T_1^{c}\|g^{s_2}T_3^{c}\|g^{s_3}T_5^{c}\|h^{s_4}T_7^{c}\|g^{s_5}h^{s_2}T_2^{c}\|T_2^{s_6}h^{s_2}T_4^{c}\|g^{s_7E}h^{s_3}T_6^{c})$

(2)验证 $c'=c$？若不相等则验证失败。

(3)若 $g=T_4T_6T_7$，则 e 和 E 互素。

1.2 ACJT 群签名中成员撤消实现

以下是利用上述协议设计的具有成员撤消功能的 ACJT 群签名方案。签名建立、加入、打开过程参考文献[2]，本文

基金项目：国家自然科学基金资助项目(60273084)
作者简介：王凤和(1979－)，男，硕士，主研方向：群签名及其应用；胡予濮，博导、教授；王春晓，硕士、讲师
收稿日期：2006-04-25 **E-mail：**fenghe2166@tom.com

只给出群签名的撤消、签名和验证过程。

撤消：

群管理员计算 $E=\prod_{e_i\in A}e_i$，其中A是所有撤消成员对应群证书中的素数。群管理员公开E和当前的时间T以及所有撤消者的e_i。当又有成员撤消时，群管理员只要在E上再乘以新撤消的群证书中的素数来更新E，同时更新时　间T。

签名：

群成员的签名过程包括两部分：首先利用ACJT群签名证明自己拥有群证书(A_i,e_i,x_i)，然后利用上述互素性证明来证明自己的e_i和当前最新的E互素，从而证明自己不是一个被撤消的群成员。

(1)利用扩展的 GCD 算法得到 a, b 满足：$ae_i+bE=1$。

(2)$u,v,\omega\in[0,1]^{2l_p}$，计算：

$T_1=A_i y^{\omega}(\bmod n),T_2=g^{\omega}(\bmod n),T_3=g^{e_i}h^{\omega}(\bmod n),$

$T_4=g^{v}(\bmod n),T_5=T_3^{a}h^{v}(\bmod n),T_6=g^{u}(\bmod n),$

$T_7=g^{Eb}h^{u}(\bmod n),T_8=h^{-(a\omega+u+v)}$

生成 ACJT 的群签名：

$PK1\{(\alpha,\beta,\delta,\varepsilon):a_0=T_1^{\alpha}a^{-\beta}y^{-\delta}\wedge T_2=g^{\varepsilon}\wedge 1$

$=T_2^{\alpha}g^{-\delta}\wedge T_3=g^{\alpha}h^{\varepsilon}\wedge\alpha\in\Gamma\wedge\beta\in\Delta\}$

利用 1.1 节给出成员未撤消的证明：

$PK2\{(\alpha,\eta,\varepsilon,\tau,\xi,\zeta):T_2=g^{\varepsilon}\wedge T_6=g^{\xi}\wedge T_8=h^{\zeta}\wedge T_3$

$=g^{\alpha}h^{\varepsilon}\wedge T_5=T_3^{\eta}h^{\tau}\wedge T_7=g^{E\gamma}h^{\xi}\wedge g=T_5T_7T_8\wedge\alpha\in\Gamma\}$

数组$(t,T_1,T_2,T_3,T_4,T_5,T_6,T_7,T_8,PK1,PK2)$作为群成员的签名。

验证：

验证者通过时间 T 得到相应的 E 然后检查 PK1，PK2 的正确性。

2 撤消方案的安全缺陷

文献[6]利用互素性证明实现群成员的撤消并指出被撤消成员不能通过这种身份验证。被撤消者如果严格按照协议的确不能通过互素性证明。但事实上一个被撤消者有可能通过改变某些量的形式以通过身份验证。又由于互素性证明协议没有很好地嵌入群签名方案中，从而使得在改变签名某些量后 ACJT 群签名系统不能发现作弊。结果一个撤消成员顺利地完成了群签名。下面给出详细讨论。

我们认为撤消算法并没有很好地嵌入群签名方案中：由于 1.2 节中T_4,T_5,T_6,T_7,T_8大量的信息并没有出现在 PK1 中，因此被撤消者可能通过改变T_4,T_5,T_6,T_7,T_8的值顺利通过 PK2，而利用自己已经作废的群证书(A_i,e_i,x_i)可以通过 PK1 的验证。从而实现签名。

设 eve 是一个不诚实的被撤销者，则 eve 可以通过如下过程通过撤销检验伪造签名：

(1)因为$e_i\mid E$，所以$(e_i,E-1)=1$，eve 利用扩展的 GCD 算法得到：$a,b:ae_i+b(E-1)=1$。又因为 E 是一些素数的乘积，所以 E 在模 n 下存在逆元。记为：E^{-1}。

(2)在 1.1 节素性证明中，eve 只改变T_7,s_7的形式使得

$T_7=g^{(E-1)b}h^{u}$, $s_7=r_7-(E-1)E^{-1}bc$

其他项的形式不变，则 eve 通过这种改变也可以通过 PK2。

证明

因为 eve 只改变了两个值，所以其他不含T_6,s_7的等式一定成立。即

$R_1=g^{s_1}T_1^{c}$, $R_2=g^{s_2}T_3^{c}$, $R_3=g^{s_3}T_5^{c}$, $R_4=h^{s_3}T_7^{c}$, $R_5=g^{s_5}h^{s_2}T_2^{c}$, $R_6=T_2^{s_6}h^{s_2}T_4^{c}$。

仅验证包含T_6,s_7的项：

$g^{Es_7}h^{s_3}T_6^{\,c}=g^{E(r_7-(E-1)E^{-1}bc)}h^{r_3-cu}g^{(E-1)bc}h^{cu}=g^{Er_7}h^{r_3}=R_7$

于是：$c=c'$成立。

另一方面：

$T_4T_6T_7=T_3^{a}h^{v}g^{b(E-1)}h^{u}h^{-(a\omega+u+v)}$

$=g^{ae}h^{a\omega}h^{v}g^{(E-1)b}h^{u}h^{-(a\omega+u+v)}=g^{ae+b(E-1)}=g$

于是 eve 通过改变T_6,s_7的形式，使得自己顺利地通过身份验证：PK2。

(3)由于 eve 仅改变了T_6,s_7的值，而T_6,s_7在 PK1 中均没有出现。即 PK1 是 eve 完全利用自己的群证书作的 ACJT 群签名。因此 eve 一定也可以通过 PK1 的验证。这样 eve 就同时通过了 PK1 和 PK2。生成了合法签名。而 eve 是一个被撤消成员。因此文献[6]存在安全缺陷。证毕。

3 结论

本文对文献[6]提出的一个高效成员撤销算法进行了安全性分析，指出它不能防止被撤销的成员继续生成合法签名。因此该撤销算法是不安全的。

文献[6]不安全的原因在于原撤消算法没有真正嵌入群签名中。从而使得被撤消者 eve 在改变T_6,s_7的形式，事实上是向验证者证明$(e_i,E-1)=1$时，验证者不能发现。要利用文献[6]给出一个安全高效的撤消算法，就必须真正地将撤消算法嵌入群签名中，使得撤消者做类似的变形后无法通过 PK1。如何实现这一设想有待进一步的研究。

参考文献

1 Chaum D, Heyst V E. Group Signatures[C]//Proc. of EUROCRYPT Lecture Notes in Computer Science,1991: 257-265.

2 Ateniese G, Camenish J, Joye M, et al. A Practical and Provably Secure Coalition-resistant Group Signature Scheme[C]//Proc. of Crypt'00. Springer-Verlag, 2000: 255-270.

3 Song D. Practical Forward Secure Group Signature[C]//Proc. of the 8th ACM Conf. on Computer and Communication Security. 2001: 225-234.

4 Ateniese G, Tsudik G. Quasi-efficient Revocation of Group Signature[Z]. 2001. http://eprint.iacr.org/2001/101/.

5 Bresson E, Stern J. Efficient Revocation in Group Signature[C]//Proc. of PKC'01. Springer-Verlag, 2001: 190-206.

6 Chen Z W, Wang J L, Huang J W, et al. An Efficient Revocation Algorithm in Group Signature[C]//Proc. of ICISC'03. Springer-Verlag, 2004: 339-351.