改进的无证书广义指定验证者聚合签名方案

胡小明[1] 马 闯[1] 斯桃枝[1] 蒋文蓉[1] 许华杰[2] 谭文安[1]
(上海第二工业大学计算机与信息工程学院 上海 201209)[1]
(广西大学计算机与电子信息学院 南宁 530004)[2]

摘 要 无证书广义指定验证者聚合签名(CTL-ASWUDV)能有效解决签名者的隐私保护问题。针对最近指出的张玉磊等学者的CTL-ASWUDV方案构造无效且不满足两类敌手攻击的问题,提出了一个改进的CTL-ASWUDV方案(CTL-ASWUDV-1)。该方案在保持了原方案中聚合签名长度和双线性配对数固定的优点的同时,有效克服了两类敌手的攻击。进一步提出了一个更加高效的CTL-ASWUDV方案(CTL-ASWUDV-2)。在随机预言机模型下,证明该方案的安全性可规约为CDH问题。同时,该方案与目前已有的同类方案相比具有如下优势:单个签名和聚合签名无需双线性配对运算,而且聚合签名验证所需的双线性配对数量与签名人数无关,与单个签名验证数量相当,都是1个配对运算;聚合签名长度和指定验证者签名长度与签名人数无关,与单个签名长度相当,都是固定的1个元素,大大节省了网络带宽。

关键词 网络安全,无证书签名,聚合签名,指定验证者签名,双线性配对

中图法分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.08.030

Improved Certificateless Aggregate Signature Scheme with Universal Designated Verifier

HU Xiao-ming[1] MA Chuang[1] SI Tao-zhi[1] JIANG Wen-rong[1] XU Hua-jie[2] TAN Wen-an[1]
(College of Computer and Information Engineering, Shanghai Second Polytechnic University, Shanghai 201209, China)[1]
(School of Computer and Electronic Information, Guangxi University, Nanning 530004, China)[2]

Abstract Certificateless aggregate signature scheme with universal designated verifier (CTL-ASWUDV) can effectively solve the problem of protecting the privacy of the signer. An improved CTL-ASWUDV scheme (CTL-ASWUDV-1) was proposed according to the problems existing in Zhang et al.'s CTL-ASWUDV scheme on the invalid construction and two types of adversary attacks. The improved scheme not only keeps the advantages of constant aggregate signature length and constant bilinear pairing operation number, but also overcomes the attacks from two types of adversaries. This paper further proposed a highly efficient CTL-ASWUDV scheme (CTL-ASWUDV-2). In the random oracle model, the security of the second improved scheme can be reduced to computational Diffie-Hellman problem. At the same time, compared with the existing similar schemes, the proposed second scheme has the following advantages. It has no bilinear pairing operation in both single signature and aggregate signature, and the number of bilinear pairing operation needed by the aggregate signature verification is independent on the number of signers and it is equivalent to the number of a single signature verification, i.e. one pairing operation. The length of an aggregate signature and the length of a designated verifier signature are both independent on the number of signers and they are equivalent to the length of a single signature verification, i.e. one element, which largely saves the network bandwidth.

Keywords Network security, Certificateless signature, Aggregate signature, Designated verifier signature, Bilinear pairing

---

到稿日期:2016-07-05 返修日期:2016-11-28 本文受上海市教育委员会科研创新基金重点项目(14ZZ167),国家自然科学基金资助项目(61103213,61272036,61672022),广西自然科学基金项目(2014GXNSFAA11838-2),上海第二工业大学校级重点学科计算机科学与技术(XXKZD1604)资助。

胡小明(1978—),女,博士,副教授,CCF会员,主要研究方向为密码学、信息安全、网络安全,E-mail:xmhu@sspu.edu.cn(通信作者);**马 闯**(1974—),男,博士,讲师,主要研究方向为物联网、软件定义网络、网络安全;**斯桃枝**(1964—),女,硕士,副教授,主要研究方向为网络工程、网络安全、路由交换;**蒋文蓉**(1968—),女,硕士,副教授,主要研究方向为大数据、大数据安全、可穿戴技术;**许华杰**(1974—),男,博士,副教授,主要研究方向为无线网络、网络安全;**谭文安**(1965—),男,博士,教授,主要研究方向为软件服务工程、可信软件、协同计算。

1 引言

传统的基于证书的公钥密码体制由于需要维护复杂的密钥管理，因此其应用受到了限制。Shamir[1]于 1984 年提出的基于身份的密码体制（IDC）直接使用用户身份信息作为公钥，有效地解决了传统密码体制中的密钥管理问题。但在 IDC 中，由于用户的私钥由密钥建立中心（KGC）生成，从而产生了密钥托管的问题。AlRiyami 和 Paterson[2]于 2003 年提出了无证书公钥密码体制（CTLC）的概念。在 CTLC 中，用户的完整私钥由 KGC 生成的部分私钥和用户自己生成的秘密值组成，从而有效地解决了传统密码体制和基于身份的密码体制这两者中存在的问题。CTLC 的良好特性吸引了众多学者的兴趣，如文献[3-5]等提出了很多优秀的 CTLC。

聚合签名（AS）最早由 Boneh 等学者[6]提出。在 AS 中，聚合者可以将多个签名者对不同消息的单个签名聚合成一个签名。当验证者验证这些签名时，只需要验证聚合后的一个签名即可确定所有单个签名的有效性。AS 由于能同时批量验证签名且缩短签名的长度，因此大大降低了系统的验证开销以及带宽的占有率，可广泛应用于各种资源受限的环境。鉴于 CTLC 和 AS 的优越性，近年来无证书聚合签名（CTLC-AS）成为了密码学的研究热点之一[7-16]。

Zhang 等学者于 2009 年基于计算 Diffie-Hellman（CDH）的困难性假设提出了一个 CTLC-AS 方案[7]，然而该方案的计算效率比较低。在单次无证书签名验证中需要 3 个最耗时的双线性配对运算，签名长度为 2 个元素。而且聚合签名的长度以及验证中双线性配对运算的数量会随着签名人数的增加而增加，这极大地降低了该方案的效率。杜红珍等学者于 2013 年在随机预言机模型下提出了一个高效的 CTLC-AS 方案[8]。该方案的聚合签名长度以及验证所需的双线性配对数分别为固定的 2 个元素和 4 个对运算，该方案被认为是当时拥有最短签名长度的 CTLC-AS 方案。然而，陈明[9]指出该方案存在安全缺陷：不满足不可伪造的安全性。同时陈明也提出了一个改进的 CTLC-AS 方案，但该改进方案也存在安全缺陷：恶意 KGC 能进行伪造攻击。

Zhou M 等[10]于 2014 年提出了一个基于状态信息的 CTLC-AS 方案。在该方案中虽然聚合签名的长度不会随签名人数的增加而增加（保持固定的 2 个元素），但是一次聚合签名验证中的双线性配对数会随着签名人数的增加而上升，这影响了其效率。陈虎等[11]于 2015 年基于公开状态信息提出了一个 CTLC-AS 方案，然而该方案也存在不足之处：聚合签名的长度会随签名人数的增加呈线性增长，这限制了其应用范围。同年，周彦伟等[12]提出了两个无需双线性配对运算的 CTLC-AS 方案。其中方案一由于无需双线性配对操作，因此计算效率略优于文献[11]的方案，但聚合签名长度存在的问题和文献[11]一样。方案二通过签名者之间共享信息解决了聚合签名长度随签名人数增加的问题，但由于产生聚合签名之前需要所有签名人相互之间发布共享的信息，并且所有签名人必须收到所有其他签名人的共享信息后才能生成自己的签名，这大大增加了通信量，同时也限制了签名人自由进入聚合签名的灵活性。

张玉磊等[13]于 2015 年将广义指定验证者签名和无证书聚合签名相结合，提出了无证书广义指定验证者聚合签名（CTL-ASWUDV）。在 CTL-ASWUDV 中，只有指定的验证者才能验证聚合签名的有效性，同时指定验证者又不能向第三方证实该聚合签名的有效性。因此，CTL-ASWUDV 签名有效保护了签名者的隐私。然而，杜红珍[14]指出，张玉磊等人的 CTL-ASWUDV 存在较多的安全缺陷，方案不仅在构造上存在缺陷，而且无法抵御两类敌手的攻击，因此该方案不能满足不可伪造的安全特性。但是，杜红珍没有给出改进的方案。本文针对杜红珍提出的问题，对张玉磊等学者的 CTL-ASWUDV 方案进行改进，提出一个能有效克服各种安全缺陷的 CTL-ASWUDV 方案，称为 CTL-ASWUDV-1。同时，受文献[4,15]的启发，提出一个高效的 CTL-ASWUDV 方案，称为 CTL-ASWUDV-2。在 CTL-ASWUDV-2 中，无论有多少个签名人，聚合签名长度和指定验证者签名长度保持固定的 1 个元素，聚合签名和指定验证者签名所需的双线性配对数量始终为 1 个。

2 准备知识

2.1 线性配对

设 G_1 和 G_2 是两个群，它们的阶均为大素数 p。$e:G_1\times G_1\to G_2$ 称为双线性配对映射，如果它满足如下的 3 个性质：

（1）双线性性：对任意 $a,b\in Z_p^*$ 以及 $P,Q\in G_1$，满足 $e(P^a,Q^b)=e(P,Q)^{ab}$。

（2）非退化性：存在生成元 $g\in G_1$ 满足 $e(g,g)\neq 1$。

（3）可计算性：对任意 $P,Q\in G_1$，$e(P,Q)$ 可有效计算。

2.2 计算 Diffie-Hellman（CDH）问题

给定一个生成元 $P\in G_1$ 以及 $aP,bP\in G_1$，其中 $a,b\in Z_p^*$，那么 CDH 问题即为给定 (P,aP,bP)，在不知道 a,b 的情况下求 abP 的值。

3 无证书广义指定验证者聚合签名（CTL-ASWUDV）相关概念和安全性要求

3.1 CTL-ASWUDV 定义

CTL-ASWUDV 方案由以下 8 个阶段组成。

（1）系统初始化阶段（Initial Setup）：该阶段由密钥生成中心（KGC）执行。输入安全参数 k、KGC 输出系统主密钥 x 和系统公共参数 $params$。

（2）秘密值生成阶段（Secret Value Generate）：该阶段由用户 U_i 执行。U_i 随机地选择一个 $s_i\in Z_p^*$ 作为它的秘密值并计算和公开公钥 Y_i。

（3）部分私钥抽取阶段（Part Secret Key Generate）：该阶段由 KGC 执行。输入用户 U_i 的身份信息 ID_i、系统主密钥 x 和系统公共参数 $params$，输出部分私钥 S_i。

（4）单个签名产生阶段（Single Signature Generate）：该阶段由用户 U_i 执行。输入消息 m_i、用户 U_i 的身份信息 ID_i 和完整私钥 (s_i,S_i) 以及系统公共参数 $params$，输出消息 m_i 上的单个签名 σ_i。

（5）聚合签名产生阶段（Aggregate Signature Generate）：该阶段由聚合签名人 U_A 执行。U_A 收到 n 个用户的单个签

名($\sigma_i=(K_i,R_i),m_i$)($1\leqslant i\leqslant n$)后,首先验证这 n 个签名是否有效。如果有效,则输入 n 个用户的身份$\{ID_1,\cdots,ID_n\}$、公钥$\{Y_1,\cdots,Y_n\}$以及对应消息$\{m_1,\cdots,m_n\}$上的 n 个签名$\{\sigma_1,\cdots,\sigma_i\}$,输出聚合签名为 σ。

(6)聚合签名验证阶段(Aggregate Signature Verify):该阶段由聚合签名验证人执行。输入 n 个用户的身份$\{ID_1,\cdots,ID_n\}$、公钥$\{Y_1,\cdots,Y_n\}$以及对应消息$\{m_1,\cdots,m_n\}$上的聚合签名 σ,验证人验证签名 σ 是否有效,如果有效则输出"true",否则输出"false"。

(7)指定验证者签名产生阶段(Designated Verifier Signature Generate):该阶段由聚合签名持有人 U_P 执行。输入聚合签名 σ 和指定验证者的公钥 Y_V,输出指定验证者聚合签名 σ_V。

(8)指定验证者签名验证阶段(Designated Verifier Signature Verify):这个阶段由指定验证者 U_V 执行。输入 n 个用户的身份$\{ID_1,\cdots,ID_n\}$、公钥$\{Y_1,\cdots,Y_n\}$、消息$\{m_1,\cdots,m_n\}$以及指定验证者的私钥 s_V 和指定验证者聚合签名 σ_V,验证人验证 σ_V 是否有效,如果有效则输出"true",否则输出"false"。

3.2 CTL-ASWUDV 安全性要求

(1)不可伪造性

在无证书聚合签名(CTL-AS)中,存在两种类型的攻击者:A_{I} 类敌手,不能获得系统主密钥,但可以替换任意用户的公钥;A_{II}类敌手,能获得系统主密钥,但不能替换目标用户的公钥。下面结合文献[11,13]给出 CTL-ASWU DV 的不可伪造性定义。CTL-ASWUDV 的不可伪造性通过挑战者 C 和敌手 $A\in\{A_{\mathrm{I}},A_{\mathrm{II}}\}$之间的如下的一个交互游戏来定义。

系统设置:C 设置系统主密钥 x 和系统公共参数 $params$。如果 C 和 A_{I} 进行游戏,则把 $params$ 发给 A_{I};如果 C 和 A_{II}进行游戏,则把 $params$ 以及 x 一起发给 A_{II}。

询问:A 向 C 进行以下的预言机询问。

秘密值询问:当 A 向 C 询问身份 ID_i 的秘密值时,C 返回 ID_i 对应的秘密值 Y_i 给 A。

部分私钥询问:当 A 向 C 询问 ID_i 的部分私钥时,C 返回部分私钥 S_i 给 A_{I}。注意,当 $A=A_{\mathrm{II}}$时,A_{II}由于知道系统主密钥,因此无需进行该类询问。

单个签名询问:当 A 向 C 询问(m_i,ID_i,Y_i)的签名时,C 返回签名 σ_i 给 A。

公钥替换询问:当 A 向 C 要求替换 ID_i 的公钥为 Y_i'时,C 将原公钥替换为 Y_i'。

指定验证者签名询问:当 A 提交一个聚合签名 σ 给 C 并要求指定给用户 U_V 来验证时,C 返回一个验证者为 U_V 的聚合签名 σ_V 给 A_{I}。

指定验证者签名验证询问:当 A 向 C 提交一个指定验证者聚合签名 σ_V 时,如果该签名有效,则 C 返回"true",否则返回"false"。

如果存在哈希预言机,C 对于 A 提交的哈希询问返回哈希值给 A。由于 A 在获得了单个签名后能自行建立聚合签名,因此在此安全模型中无需聚合签名询问。

伪造:A 最终输出一个身份$\{ID_1^*,\cdots,ID_n^*\}$、公钥$\{Y_1^*,\cdots,Y_n^*\}$上的 n 个用户对消息$\{m_1^*,\cdots,m_n^*\}$的无证书指定验证者聚合签名 σ_V^*。

1)σ_V^* 是一个有效的无证书指定验证者聚合签名。

2)存在一个 $ID_k^*\in\{ID_1^*,\cdots,ID_n^*\}$,$A$ 没有向 C 做过关于 ID_k^* 的部分私钥询问,也没有做过(ID_k^*,Y_k^*,m_k^*)的单个签名询问。

3)存在一个 $ID_k^*\in\{ID_1^*,\cdots,ID_n^*\}$,$A$ 没有向 C 做过关于 ID_k^* 的秘密值询问和公钥替换询问,也没有做过(ID_k^*,Y_k^*,m_k^*)的单个签名询问。

如果 1)和 2)满足,那么 A_{I} 赢得了上述游戏的胜利;如果 1)和 3)满足,那么 A_{II}赢得了上述游戏的胜利。

如果不存在多项式的敌手 A 能在上述游戏中以不可忽略的概率获胜,那么称该 CTL-ASWUDV 在适应性选择消息和身份攻击下满足不可伪造性。

(2)指定验证者特性

在聚合签名的验证过程中需要使用验证者的私钥,因此聚合签名的验证只能被指定的验证者验证。

(3) 不可传递性

指定的验证者能有效验证聚合签名的有效性,但不能向其他第三方证明。因为指定验证者能建立一个与签名者产生的聚合签名不可区分的模拟聚合签名。

4 改进的 CTL-ASWUDV 方案

文献[14]指出,张玉磊等[13]提出的 CTL-ASWUDV 方案不仅构造无效而且不能满足无证书签名方案中的两类敌手攻击,但文献[14]没有给出改进的方法。为了解决这些问题,本文对张玉磊等学者的方案进行简单修改并提出一个改进的 CTL-ASWUDV 方案,称为 CTL-ASWUDV-1。改进的方案是对 CTL-ASWUDV 方案的一个简单修改。具体如下:

Initial Setup:KGC 随机地选择两个 G_1 的生成元 P 和 Q,同时随机选择 $x\in Z_p^*$ 并计算 $P_{pub}=xP$。然后,KGC 定义 6 个单向哈希函数 H_1-H_6,其中 $H_1-H_2:\{0,1\}^*\rightarrow Z_p^*$;$H_3:\{0,1\}^*\times G_1\rightarrow G_1$;$H_4-H_6:\{0,1\}^*\rightarrow G_1$。最后,KGC 把$\{G_1,G_2,e,p,P,Q,P_{pub},H_1-H_6\}$公开发布并将 x 作为系统主密钥秘密保存。

Secret Value Generate:U_i 随机选择一个 $s_i(s_i\in Z_p^*)$作为他的秘密值,并将 $Y_i=s_iP$ 作为他的公钥且将其公开。

Part Secret Key Generate:设用户 U_i 的身份信息为 ID_i,那么 U_i 的部分私钥为 $S_i=xQ_i$,并将 S_i 发给 U_i,其中 $Q_i=H_3(ID_i,Y_i)$。那么 U_i 的完整私钥为(s_i,S_i)。

Single Signature Generate:为了生成消息 m_i 上的签名,U_i 随机地选取 $k_i\in Z_p^*$,然后计算 $K_i=k_iP$,$c_i=H_1(ID_i,m_i,Y_i,K_i)$,$l_i=H_2(ID_i,m_i,Y_i)$,$P_1=H_4(P_{pub})$,$P_2=H_5(P)$,$R_i=S_i+c_ik_iP_1+l_is_iP_2$。最后,$U_i$ 把消息 m_i 上的单个签名 $\sigma_i(\sigma_i=(K_i,R_i))$发给聚合签名人 U_A。

Aggregate Signature Generate:U_A 收到 n 个用户的单个签名$(\sigma_i=(K_i,R_i),m_i)(1\leqslant i\leqslant n)$后,首先用如下方法验证这 n 个签名是否有效。U_A 计算 $c_i=H_1(ID_i,m_i,Y_i,K_i)$,$l_i=H_2(ID_i,m_i,Y_i)$,$P_1=H_4(P_{pub})$,$P_2=H_5(P)$,验证等式 $e(R_i,P)=e(Q_i,P_{pub})e(P_1,c_iK_i)e(P_2,l_iY_i)$是否成立,其中 $1\leqslant i\leqslant n$。如果这 n 个等式都成立,则按如下方式产生聚合签名:

$c_i=H_1(ID_i,m_i,Y_i,K_i),1\leqslant i\leqslant n$

$K=\sum_{i=1}^{n}c_iK_i$

$R=\sum_{i=1}^{n}R_i$

那么 U_A 最终得到的聚合签名为 $\sigma=(K,R)$。

Aggregate Signature Verify：为了验证消息 $m_i(1\leqslant i\leqslant n)$ 上的聚合签名 $\sigma=(K,R)$ 的有效性，验证人计算 $P_1=H_4(P_{pub})$，$P_2=H_5(P)$，$l_i=H_2(ID_i,m_i,Y_i)$，然后验证 $e(R,P)=e(\sum_{i=1}^{n}Q_i,P_{pub})e(P_1,K)e(P_2,\sum_{i=1}^{n}l_iY_i)$ 是否成立。如果成立，则接受这个聚合签名；否则拒绝。

Designated Verifier Signature Generate：假设 U_P 要将该签名指定给用户 U_V 来验证，设 U_V 的身份信息为 ID_V，公钥为 Y_V，然后 U_P 计算 σ_V：

$$\sigma_V=e(R,Y_V)=[e(\sum_{i=1}^{n}Q_i,P_{pub})e(P_1,K)e(P_2,\sum_{i=1}^{n}l_iY_i)]^{s_V}$$

那么，最后建立的指定验证者聚合签名为 (σ_V,K)。

Designated Verifier Signature Verify：U_V 收到 (σ_V,K) 后，验证等式 $[e(\sum_{i=1}^{n}Q_i,P_{pub})e(P_1,K)e(P_2,\sum_{i=1}^{n}l_iY_i)]^{s_V}=\sigma_V$ 是否成立。如果成立，则接受这个聚合签名；否则拒绝。

用以上方式产生的聚合签名是正确的，因为：

$$\begin{aligned}e(R_i,P)&=e(S_i+c_ik_iP_1+l_is_iP_2,P)\\&=e(S_i,P)e(c_ik_iP_1,P)e(l_is_iP_2,P)\\&=e(Q_i,P_{pub})e(P_1,c_iK_i)e(P_2,l_iY_i)\end{aligned}$$

$$\begin{aligned}e(R,P)&=e(\sum_{i=1}^{n}(S_i+c_ik_iP_1+l_is_iP_2),P)\\&=e(\sum_{i=1}^{n}S_i,P)e(\sum_{i=1}^{n}c_ik_iP_1,P)e(\sum_{i=1}^{n}l_is_iP_2,P)\\&=e(\sum_{i=1}^{n}Q_i,P_{pub})e(P_1,K)e(P_2,\sum_{i=1}^{n}l_iY_i)\end{aligned}$$

4.1 效率和安全性分析

效率分析：本文提出的改进方案 CTL-ASWUDV-1 与张玉磊等学者的方案相比主要有两处不同：单个签名生成阶段和指定验证者签名阶段。在单个签名阶段，本文增加了 $l_i=H_2(ID_i,m_i,Y_i)$，并将签名的主要参数 R_i 的构成从原来的 $R_i=S_i+c_ik_iP_1+s_iQ$ 变成了 $R_i=S_i+c_ik_iP_1+l_is_iP_2$，其中 $P_2=H_5(P)$。对应地，后面的验证、聚合和指定验证者签名等阶段相应地改变。从这个变化可以看出，本文的改进在单个签名阶段仅仅增加了一次 Z_p^* 上的乘法操作即 l_is_i 以及两次哈希操作即映射到 Z_p^* 的 $H_2(ID_i,m_i,Y_i)$ 和 $P_2=H_5(P)$，没有增加最耗时的配对操作，其中 $P_2=H_5(P)$ 可以提前预计算。相比于最耗时的配对操作，Z_p^* 上的乘法和哈希映射可以忽略不计。对应地，在单个签名验证阶段和聚合签名等阶段也没有增加最耗时的配对操作，仍然保持 4 个配对操作；在指定验证者签名阶段，本方案将张玉磊等学者的原方案中的 3 个计算步骤即 $h_V=H_6(s_PY_V)$，$\sigma_V=e(R,h_VY_V)$ 和 $K_V=h_VK$ 变成了一个计算步骤，即 $\sigma_V=e(R,Y_V)$。对应地，在指定验证者签名验证过程中，省去了 $h_V=H_6(s_VY_P)$ 的计算，最耗时的配对操作个数不变，仍然为 4。因此，总的来说，本文的改进方案保持了张玉磊等学者的原方案的计算高效率。此外，在签名长度方面，本文的改进方案同张玉磊等学者的方案一样，单个签名和聚合签名长度相当，且都是固定的 2 个元素。

安全性分析：下面主要分析本文改进的 CTL-ASWUDV-1 方案是否能够有效克服文献[14]指出的针对张玉磊等学者的方案的攻击。本文提出的方案是对张玉磊等学者的 CTL-ASWUDV 方案的改进，因此其余的安全性证明方法可以采用张玉磊等学者的安全证明方法，证明过程和文献[13]类似。

针对第二类敌手 A_{II} 的攻击[14]：恶意的 KGC 不替换目标用户的公钥，但是能通过修改系统参数伪造聚合签名。文献[14]指出，在张玉磊等学者的 CTL-ASWUDV 方案中，A_{II} 敌手通过将系统参数之一的 Q 设置成 tP（即 $Q=tP$）并计算 $R_i=S_i+c_ik_iP_1+tY_i$（其中 $t\in Z_p^*$）可以伪造一个有效的签名（具体攻击过程参考文献[14]）。在本文改进的 CTL-ASWUDV-1 方案中，$R_i=S_i+c_ik_iP_1+l_is_iP_2$，将原来的 Q 用哈希值 $P_2=H_5(P)$ 代替，这样有效抵制了敌手 A_{II} 在系统参数发布阶段将 Q 设置成 tP 为后续的攻击留下后门，从而即使敌手 A_{II} 知道系统主密钥 x，也无法计算 $R_i=S_i+c_ik_iP_1+l_is_iP_2$，因为要计算 s_iP_2 必须先解决离散对数问题，即从用户的公钥 $Y_i=s_iP$ 中获得 s_i。根据离散对数问题假设，这是不可能的。因此，本文的改进 CTL-ASWUDV-1 方案有效地抵御了第二类敌手 A_{II} 的攻击[14]。

针对第一类敌手 A_{I} 的攻击[14]：首先 A_{I} 替换聚合签名持有人的公钥，然后建立有效的指定验证者聚合签名。文献[14]指出，在张玉磊等学者的 CTL-ASWUDV 方案中，敌手 A_{I} 通过随机选择 $s_P'\in Z_p^*$ 将聚合签名持有人 U_P 的公钥替换成 $Y_P'=s_P'P$，最终伪造一个指定验证者签名 $h_V'=H_6(s_P'Y_V)$，$\sigma_V'=e(R,h_V'Y_V)$，$K_V'=h_V'K$。在本文的改进方案中，由于无需使用聚合签名持有人 U_P 的公钥即可计算指定验证者签名 $\sigma_V=e(R,Y_V)$，因此自然地抵制了第一类敌手 A_{I} 的替换公钥攻击[14]。

针对无证书聚合签名方案构造无效的解决，文献[14]指出，在张玉磊等学者的 CTL-ASWUDV 方案中，由于验证等式 $e(R,P)=e(\sum_{i=1}^{n}Q_i,P_{pub})e(P_1,K)e(Q,\sum_{i=1}^{n}Y_i)$ 中没有用到签名的消息 $m_i(1\leqslant i\leqslant n)$，因此即使验证等式成立也无法判断该签名是否为消息 $m_i(1\leqslant i\leqslant n)$ 上的聚合签名。本文改进的 CTL-ASWUDV-1 方案在验证等式中增加了哈希值 $l_i=H_2(ID_i,m_i,Y_i)$，从而签名验证者在验证签名时必须首先计算 $l_i=H_2(ID_i,m_i,Y_i)$，然后验证 $e(R,P)=e(\sum_{i=1}^{n}Q_i,P_{pub})e(P_1,K)e(P_2,\sum_{i=1}^{n}l_iY_i)$ 是否成立，这样就解决了验证者无法判定聚合签名是否为消息 $m_i(1\leqslant i\leqslant n)$ 上的签名的问题。

5 高效的无证书广义指定验证者聚合签名（CTL-ASWUDV）方案

上述的改进方案 CTL-ASWUDV-1 虽然解决了文献[14]指出的安全问题，但聚合签名的验证仍然需要 4 个配对操作。受文献[4,15]的启发，接下来本文提出一个更加有效的无证书广义指定验证者聚合签名方案 CTL-ASWUDV-2 。CTL-ASWUDV-2 的具体构造如下。

Initial Setup：KGC 随机选择一个 G_1 的生成元 P，同时随机选择 $x\in Z_p^*$ 并计算 $P_{pub}=xP$。然后，KGC 定义 4 个单向

哈希函数 H_1-H_4，其中 $H_1,H_2,H_4:\{0,1\}^*\rightarrow G_1$；$H_3:\{0,1\}^*\rightarrow Z_p^*$。最后，KG 将把$\{G_1,G_2,e,p,P,P_{pub},H_1-H_4\}$公开发布并将 x 作为系统主密钥秘密保存。

Secret Value Generate：身份信息为 ID_i 的用户 U_i 随机地选择一个 $s_i\in Z_p^*$ 作为他的秘密值，并将 $Y_i=s_iP$ 作为他的公钥公开。

Part Secret Key Generate：设用户 U_i 的身份信息为 ID_i，那么 U_i 的部分私钥为 $S_i=xQ_i$，并将 S_i 发给 U_i，其中 $Q_i=H_1(ID_i,Y_i)$。那么 U_i 的完整私钥为(s_i,S_i)。

Single Signature Generate：为了生成消息 m_i 上的签名，U_i 计算 $c_i=H_3(ID_i,m_i,Y_i),W_i=H_4(ID_i),\sigma_i=s_iH_2(ID_i)-c_i(S_i+s_iW_i)$。最后，$U_i$ 把消息 m_i 上的单个签名 σ_i 发给聚合签名人 U_A。

Aggregate Signature Generate：U_A 收到 n 个用户的单个签名$(\sigma_i,m_i)(1\leqslant i\leqslant n)$后，首先用如下方法验证这 n 个签名是否有效。U_A 计算 $c_i=H_3(ID_i,m_i,Y_i)$和 $W_i=H_4(ID_i)$，验证等式 $e(\sigma_i,P)=(e(P_{pub},Q_i)e(W_i,Y_i))^{-c_i}\ e(H_2(ID_i),Y_i)$是否成立，其中 $1\leqslant i\leqslant n$，如果这 n 个等式都成立，则计算 $\sigma=\sum_{i=1}^{n}\sigma_i$。从而 U_A 最终得到的聚合签名为 σ。

Aggregate Signature Verify：为了验证消息 $m_i(1\leqslant i\leqslant n)$上的聚合签名 σ 的有效性，验证人计算 $c_i=H_3(ID_i,m_i,Y_i)$和 $W_i=H_4(ID_i)$ $(1\leqslant i\leqslant n)$，然后验证 $e(\sigma,P)=\prod_{i=1}^{n}(e(P_{pub},Q_i)e(W_i,Y_i))^{-c_i}\prod_{i=1}^{n}e(H_2(ID_i),Y_i)$是否成立。如果成立，则接受这个聚合签名，否则拒绝。

Designated Verifier Signature Generate：假设 U_P 为聚合签名的持有人，U_P 要将一个聚合签名 σ 指定给用户 U_V 来验证，设 U_V 的身份信息为 ID_V，公私钥为 Y_V 和(s_V,S_V)，U_P 计算 $\sigma_V=e(\sigma,Y_V)$。那么，最后建立的指定验证者聚合签名为 σ_V。

Designated Verifier Signature Verify：U_V 收到 σ_V 后，验证等式 $\sigma_V=\prod_{i=1}^{n}(e(P_{pub},Q_i)e(W_i,Y_i))^{-c_is_V}\prod_{i=1}^{n}e(H_2(ID_i),Y_i)^{s_V}$是否成立。如果成立，则接受这个聚合签名；否则拒绝。以上建立的签名是正确的，因为：

$$
\begin{aligned}
e(\sigma_i,P)&=e(s_iH_2(ID_i)-c_i(S_i+s_iW_i),P)\\
&=e(s_iH_2(ID_i),P)e(-c_iS_i,P)e(-c_is_iW_i,P)\\
&=e(H_2(ID_i),s_iP)(e(xQ_i,P)e(W_i,s_iP))^{-c_i}\\
&=e(H_2(ID_i),Y_i)(e(Q_i,xP)e(W_i,Y_i))^{-c_i}\\
&=(e(P_{pub},Q_i)e(W_i,Y_i))^{-c_i}e(H_2(ID_i),Y_i)
\end{aligned}
$$

$$
\begin{aligned}
e(\sigma,P)&=e(\sum_{i=1}^{n}\sigma_i,P)\\
&=e(\sum_{i=1}^{n}(s_iH_2(ID_i)-c_i(S_i+s_iW_i)),P)\\
&=e(\sum_{i=1}^{n}s_iH_2(ID_i),P)e(\sum_{i=1}^{n}-c_iS_i,P)e(\sum_{i=1}^{n}-c_is_iW_i,P)\\
&=\prod_{i=1}^{n}e(H_2(ID_i),s_iP)\prod_{i=1}^{n}e(Q_i,xP)^{-c_i}\prod_{i=1}^{n}e(W_i,s_iP)^{-c_i}\\
&=\prod_{i=1}^{n}(e(P_{pub},Q_i)e(W_i,Y_i))^{-c_i}\prod_{i=1}^{n}e(H_2(ID_i),Y_i)
\end{aligned}
$$

$$
\begin{aligned}
\sigma_V&=e(\sigma,Y_V)\\
&=e(\sum_{i=1}^{n}\sigma_i,s_VP)\\
&=(e(\sum_{i=1}^{n}(s_iH_2(ID_i)-c_i(S_i+s_iW_i)),P))^{s_V}\\
&=(\prod_{i=1}^{n}(e(P_{pub},Q_i)e(W_i,Y_i))^{-c_i}\prod_{i=1}^{n}e(H_2(ID_i),Y_i))^{s_V}\\
&=\prod_{i=1}^{n}(e(P_{pub},Q_i)e(W_i,Y_i))^{-c_is_V}\prod_{i=1}^{n}e(H_2(ID_i),Y_i)^{s_V}
\end{aligned}
$$

5.1 效率和安全性分析

5.1.1 效率

为了验证本文提出的 CTL-ASWUDV-2 方案的高效率，下面将本方案和目前已有的一些类似的无证书聚合签名方案[7-11,13]从计算效率和签名长度两方面进行比较(见表 1 和表 2)。在比较过程中，考虑一些计算量比较大的运算，如配对操作(用 TP 表示一次运算)、群 G_1 上的点乘操作(用 TM_1 表示一次点乘运算)、群 G_1 上的点加操作(用 TA 表示一次点加运算)、群 G_2 上的乘法操作(用 TM_2 表示一次乘法运算)、群 G_2 上的指数操作(用 TE_2 表示一次点乘运算)和映射到 G_1 的哈希操作(用 TH 表示一次该映射操作运算)。用$|G_1|$表示 G_1 上一个元素的比特长度，$|Z_p^*|$表示 Z_p^* 上一个元素的比特长度。

表 1 各无证书聚合签名方案的计算量比较

方案	单个签名	单个签名验证	n 个消息的聚合签名	聚合签名验证	指定验证者签名和验证
文献[7]	$2TA+3TM_1+2TH$	$3TP+2TM_2+2TH$	$(n-1)TA$	$(n+1)TP+2nTM_2+(n+1)TH=T_1$	T_1+1TE_2
文献[8]	$2TA+4TM_1$	$2TP+2TM_2+2TE_2$	$(2n-2)\ TA$	$2TP+2nTM_2+2nTE_2=T_2$	T_2+1TE_2
文献[10]	$2TA+3TM_1+2TH$	$3TP+2TM_2+2TH$	$(2n-2)\ TA$	$(n+1)TP+(n+1)TM_2+(n+1)TH=T_3$	T_3+1TE_2
文献[13]	$2TA+3TM_1$	$2TP+2TM_2+1TE_2$	$nTM_1+(2n-2)TA$	$2TP+2nTM_2=T_4$	T_4+1TE_2
文献[9]	$2TA+6TM_1+2TH$	$2TP+2TM_2+2TE_2+2TH$	$(2n-2)TA$	$2TP+(2n-1)\ TM_2+2nTE_2+2TH=T_5$	T_5+1TE_2
文献[11]	$2TA+4TM_1+2TH$	$2TP+2TM_2+2TE_2+2TH$	$(n-1)TA$	$2TP+2nTM_2+2nTE_2+(n-1)TA+2TH=T_6$	T_6+1TE_2
本文算法 CTL-ASWUDV-1	$2TA+3TM_1$	$2TP+2TM_2+2TE_2$	$nTM_1+(2n-2)TA$	$2TP+(n+1)TM_2+nTE_2=T_7$	T_7+1TE_2
本文算法 CTL-ASWUDV-2	$2TA+3TM_1$	$1TP+2TM_2+1TE_2$	$(n-1)TA$	$1TP+nTE_2+nTM_2=T_8$	T_8+1TE_2

表 2　各无证书聚合签名方案的签名长度比较

方案	单个签名长度	聚合签名长度	指定验证者签名长度
文献[7]	$2\|G_1\|$	$(n+1)\|G_1\|$	$(n+1)\|G_1\|$
文献[8]	$2\|G_1\|$	$2\|G_1\|$	$2\|G_1\|$
文献[10]	$2\|G_1\|$	$2\|G_1\|$	$2\|G_1\|$
文献[13]	$2\|G_1\|$	$2\|G_1\|$	$2\|G_1\|$
文献[9]	$2\|G_1\|$	$2\|G_1\|$	$2\|G_1\|$
文献[11]	$2\|G_1\|$	$(n+1)\|G_1\|$	$(n+1)\|G_1\|$
本文算法 CTL-ASWUDV-1	$2\|G_1\|$	$2\|G_1\|$	$2\|G_1\|$
本文算法 CTL-ASWUDV-2	$1\|G_1\|$	$1\|G_1\|$	$1\|G_1\|$

从表 1 可以看出，本文提出的 CTL-ASWUDV-1 方案的计算量小于文献[7,9-11]的方案，与文献[8,13]的方案基本相当，但文献[8,13]的方案已被证实存在严重的安全缺陷。在签名长度方面，从表 2 可以看到，本文的 CTL-ASWUDV-1 方案优于文献[7,11]的方案，与文献[8-10,13]的方案一致，都是两个 G_1 元素。因此，综合来看，本文的 CTL-ASWUDV-1 方案略优于其他几个 CTL-ASWUDV 方案[7-11,13]。

对于本文提出的 CTL-ASWUDV-2，从表 1 的统计数据可以看到，CTL-ASWUDV-2 方案无论是对单个签名还是 n 个消息的聚合签名和指定验证者签名，最耗时的配对操作都只需要固定的 1 个，因此在计算量上本方案明显小于文献[7,9-11]的方案，而略小于文献[8,13]的方案。对于签名长度，从表 2 可以看出，本文的 CTL-ASWUDV-2 方案的签名长度不会随签名人数量的增加而增加，单个签名和 n 个消息的聚合签名及指定验证者签名的签名长度都是恒定的 1 个 G_1 元素，就我们所知，其比目前所存在的 CTLC-AS 方案和 CTL-ASWU DV 方案的都要小。因此，本文改进的 CTL-ASWUDV-2 方案目前在 CTLC-AS 方案和 CTL-ASWUDV 方案中效率是最好的。

注意：在文献[7-11,13]的所有方案的计算量统计中，都没有把可以预计算的操作统计在内，比如在文献[7-11,13]的方案以及本文的 CTL-ASWUDV-1 和 CTL-ASWUDV-2 方案中，验证者事先可以预计算 $e(P_{pub},Q_i)$，$e(Q_0,Y_i)$，$e(H_2(ID_i),Y_i)$和 $H_4(P_{pub})$和$H_4(ID_i)$等。另外，除了文献[13]的方案直接给出了指定验证者签名和验证的构造外，文献[7-11]的方案都没有给出如何从无证书聚合签名转化成指定验证者签名以及验证的方法。但文献[7-11]的方案都可采用与本文类似的方法进行转化，因此我们对这些方案采用和本文一样的方法转化成指定验证者聚合签名后再进行统计。由于文献[12]的方案无法用本文的方法直接转变成 CTL-ASWUDV 方案(可能存在其他转化方法，但代价非常大)，因此本文未作比较统计。表 1 和表 2 的统计结果显示，增加指定验证者后本文的 CTL-ASWUDV-2 方案仍然是最高效的。

5.1.2　安全性

定理 1　如果提出的 CTL-ASWUDV-2 方案能被 A_{I} 敌手在时间 t 内以概率 ε 攻破，那么挑战者 C_I 能利用 A_{I} 在 $t'\leqslant 2t+2\sum_{i=1}^{10}t_iq_i+2t_A$，$\varepsilon'\geqslant\varepsilon^2(1-q_2^{-1})^2(C_{q_4}^{n})^{-1}(nq_2^{-1})^2$ 内解决一个 CDH 问题。其中，$q_1,\cdots,q_{10}$ 分别表示敌手能进行的秘密值询问、H_1 哈希询问、H_2 哈希询问、H_3 哈希询问、H_4 哈希询问、部分私钥询问、单个签名询问、公钥替换询问、指定验证者签名询问、指定验证者签名验证询问的次数，$t_1,\cdots,t_{10}$ 表示上述每种询问所需要的时间，t_A 表示敌手进行一次伪造所需要的时间。

证明：给定一个随机的 CDH 问题实例$(P,P_1=aP,P_2=bP)$，挑战者 C_{I} 的目标是利用 A_I 求得的 abP 值。为了求得该值，C_{I} 和 A_{I} 执行 3.2 节定义的安全游戏。

系统设置：C_{I} 设置 $P_{pub}=P_1$，然后将系统参数$\{G_1,G_2,e,p,P,P_{pub},H_1-H_4\}$发给 A_{I}。

询问：A_{I} 向 C_{I} 进行如下预言机询问。假设 T_{Mi}，T_{H1i}，T_{H2i}，T_{H3i}，T_{H4i}为 5 个列表，用来保存数据，并将其初始化为空。

秘密值询问：当 A_{I} 向 C_{I} 询问 ID_i 的秘密值时，C_{I} 检查列表 T_{Mi}，如果 ID_i 没在 T_{Mi} 列表中，那么 C_{I} 随机选择 $s_i\in Z_p^*$ 并计算 $Y_i=s_iP$，然后将(ID_i,s_i,Y_i)保存到列表 T_{Mi} 中，返回 s_i 给 A_{I}。如果 ID_i 存在于 T_{Mi} 列表中，C_I 直接返回 s_i 给 A_{I}。

H_1 哈希询问：C_I 随机选择 $k\in\{1,\cdots,q_2\}$。当 A_{I} 向 C_{I} 询问 $H_1(ID_i,Y_i)$的值时，如果 $i\neq k$，那么 C_I 随机选择 $\lambda_i\in Z_p^*$，设置 $Q_i=\lambda_iP$，$S_i=\lambda_iP_1$；如果 $i=k$，那么设置 $Q_i=\lambda_iP_2$，$S_i=$null。保存$(ID_i,Y_i,\lambda_i,Q_i,S_i)$到 T_{H1i}，返回 Q_i 给 A_{I}。

H_2 哈希询问：当 A_{I} 向 C_{I} 询问 $H_2(ID_i)$的值时，C_{I} 随机选择 $\beta_i\in Z_p^*$，并计算 $T_i=\beta_iP$，将(ID_i,β_i)保存到 T_{H2i} 中，返回 T_i 给 A_{I}。

H_3 哈希询问：当 A_{I} 向 C_{I} 询问 $H_3(ID_i,m_i,Y_i)$的值时，C_{I} 随机选择 $c_i\in Z_p^*$，将(ID_i,Y_i,m_i,c_i)保存到 T_{H3i} 中，返回 c_i 给 A_{I}。

H_4 哈希询问：当 A_{I} 向 C_{I} 询问 $H_4(ID_i)$的值时，C_{I} 随机选择 $\alpha_i\in Z_p^*$，并计算 $W_i=\alpha_iP$，将(ID_i,α_i)保存到 T_{H4i} 中，返回 W_i 给 A_{I}。

部分私钥询问：当 A_{I} 向 C_{I} 询问 ID_i 的部分私钥时，如果 $i=k$，游戏终止；如果 $i\neq k$，C_I 在 T_{H1i} 中找到 ID_i 所在的元组$(ID_i,Y_i,\lambda_i,Q_i,S_i)$，并将 S_i 返回给 A_{I}。

单个签名询问：当 A_{I} 向 C_{I} 询问(m_i,ID_i,Y_i)的签名时，C_I 从 H_1-H_4 中分别获得 λ_i,β_i,c_i 和 α_i，如果在 H_1-H_4 中未找到相关项，那么 C_I 分别询问 H_1-H_4 以获得相关的元素。然后，C_I 计算 $\sigma_i=\beta_iY_i-c_i\alpha_iY_i-c_i\lambda_iP_1$，返回 σ_i 给 A_I。那么 C_I 产生的签名 σ_i 是有效的，因为：

$$\begin{aligned}e(\sigma_i,P)&=e(\beta_iY_i-c_i\alpha_iY_i-c_i\lambda_iP_1,P)\\&=e(\beta_iY_i,P)e(-c_i\alpha_iY_i,P)e(-c_i\lambda_iP_1,P)\\&=e(Y_i,\beta_iP)(e(Y_i,\alpha_iP)e(P_1,\lambda_iP))^{-c_i}\\&=(e(P_{pub},Q_i)e(W_i,Y_i))^{-c_i}e(H_2(ID_i),Y_i)\end{aligned}$$

公钥替换询问：当 A_{I} 向 C_{I} 要求替换 ID_i 的公钥为 Y_i' 时，C_{I} 将 T_{Mi} 中的(ID_i,s_i,Y_i)元组替换为$(ID_i,\mathrm{null},Y_i')$。

指定验证者签名询问：当 A_{I} 提交一个聚合签名 σ 给 C_{I} 且要求将其指定给用户 U_V 来验证时，设 U_V 的身份信息为

ID_V，公钥为 Y_V，那么 C_I 计算 $\sigma_V=e(\sigma,Y_V)$并将 σ_V 返回给 A_I。

指定验证者签名验证询问：对于 A_I 提交的指定验证者聚合签名 σ_V，C_I 使用上面的 Designated Verifier Signature Verify 算法进行验证，如果有效，则返回“true”，否则返回“false”。

最终伪造：经过上述询问后，A_I 最终输出一个身份 $\{ID_1^*,\cdots,ID_n^*\}$、公钥 $\{Y_1^*,\cdots,Y_n^*\}$ 上 n 个用户对消息 $\{m_1^*,\cdots,m_n^*\}$的有效的无证书指定验证者聚合签名 σ_V^*。如果存在一个 $ID_k^*\in\{ID_1^*,\cdots,ID_n^*\}$没有向 C_I 做过关于 ID_k^* 的部分私钥询问，也没有做过 (ID_k^*,Y_k^*,m_k^*)的单个签名询问，那么根据 Pointeheval and Stern 的 Forking lemma[17]重复上述交互过程，可以以不可忽略的概率得到 $c_k^*=H_3(ID_k^*,m_k^*,Y_k^*)\neq\overline{c_k^*}=H_3(ID_k^*,m_k^*,Y_k^*)$。由此，可以得到：

$$\sigma_V^*=(e(P_{pub},Q_k^*)e(W_k,Y_k^*))^{-c_k^*s_V}\cdot\prod_{i=1,i\neq k}^{n}(e(P_{pub},Q_i^*)e(W_i,Y_i^*))^{-c_i^*s_V}\prod_{i=1}^{n}e(H_2(ID_i^*),Y_i^*)^{s_V}$$

$$\overline{\sigma_V^*}=(e(P_{pub},Q_k^*)e(W_k,Y_k^*))^{-\overline{c_k^*}s_V}\cdot\prod_{i=1,i\neq k}^{n}(e(P_{pub},Q_i^*)e(W_i,Y_i^*))^{-c_i^*s_V}\prod_{i=1}^{n}e(H_2(ID_i^*),Y_i^*)^{s_V}$$

上述两式相除后得到：

$$e(\sigma^*,Y_V)/e(\overline{\sigma^*},Y_V)=(e(P_{pub},Q_k^*)e(W_k,Y_k^*))^{-c_k^*s_V}/(e(P_{pub},Q_k^*)e(W_k,Y_k^*))^{-\overline{c_k^*}s_V}$$

由 $W_k=\alpha_kP,Q_k^*=\lambda_kbP$ 可以得到：

$$abP=((\overline{\sigma_V^*}-\sigma_V^*)(\overline{c_k^*}-c_k^*)^{-1}-\alpha_kY_k^*)\lambda_k^{-1}$$

由此，C_I 成功解决了一个给定的 CDH 问题。采用与文献[11]类似的分析方法，可以得到 C_I 在上述交互中的成功概率为：

$$t'\leqslant 2t+2\sum_{i=1}^{10}t_iq_i+2t_A,\varepsilon'\geqslant\varepsilon^2(1-q_2^{-1})^2(C_{q_4}^n)^{-1}(nq_2^{-1})^2$$

定理 2 如果提出的 CTL-ASWUDV-2 方案能被 A_{II} 敌手在时间 t 内以概率 ε 攻破，那么挑战者 C_{II} 能在 $t'\leqslant 2t+2\sum_{i=1,i\neq4}^{10}t_iq_i+2t_A,\varepsilon'\geqslant\varepsilon^2(1-q_2^{-1})^{2(q_1+q_8)}(C_{q_4}^n)^{-1}(nq_2^{-1})^2$ 内解决一个 CDH 问题。

证明：给定一个随机的 CDH 问题实例$(P,P_1=aP,P_2=bP)$，挑战者 C_{II} 的目标是利用 A_{II} 求得 abP 值。为了求得该值，C_{II} 和 A_{II} 执行如 3.2 节定义的安全游戏。

首先，在系统设置阶段：C_{II} 设置 $P_{pub}=tP$，其中 t 为系统主密钥，将系统参数$\{G_1,G_2,e,p,P,P_{pub},H_1-H_4\}$以及 t 发给 A_{II}。

接下来，在询问阶段：由于 A_{II} 知道系统主密钥，因此 A_{II} 不需要进行部分私钥询问。H_2 哈希询问、H_3 哈希询问、公钥替换询问、指定验证者签名询问和指定验证者签名验证询问与定理 1 类似。其余询问如下。

秘密值询问：C_{II} 随机选择 $k\in\{1,\cdots,q_1\}$。当 A_{II} 向 C_{II} 询问 ID_i 的秘密值时，如果 $i\neq k$，那么 C_{II} 随机选择 $s_i\in Z_p^*$ 并计算 $Y_i=s_iP$；如果 $i=k$，那么 C_{II} 计算 $Y_i=s_iP_2$。将(ID_i,s_i,Y_i)保存到列表 T_{Ms} 中，返回 s_i 给 A_{II}。

H_1 哈希询问：当 A_{II} 向 C_{II} 询问 $H_1(ID_i,Y_i)$的值时，C_{II} 随机选择 $\lambda_i\in Z_p^*$，设置 $Q_i=\lambda_iP$。将(ID_i,Y_i,λ_i,Q_i)保存到 T_{H1i}，返回 Q_i 给 A_{II}。

H_4 哈希询问：当 A_{II} 向 C_{II} 询问 $H_4(ID_i)$的值时，如果 $i\neq k$，C_{II} 随机选择 $\alpha_i\in Z_p^*$，计算 $W_i=\alpha_iP$；如果 $i=k$，那么 C_{II} 计算 $W_i=\alpha_iP_2$。将(ID_i,α_i)保存到 T_{H4i} 中，返回 W_i 给 A_{II}。

单个签名询问：当 A_{II} 向 C_{II} 询问(m_i,ID_i,Y_i)的签名时，如果 $i=k$，失败退出。否则，C_{II} 从 H_1-H_4 中分别获得 λ_i,β_i，c_i 和 α_i，如果在 H_1-H_4 中未找到相关项，那么 C_{II} 通过分别询问 H_1-H_4 获得相关的元素。然后，C_{II} 计算 $\sigma_i=\beta_iY_i-c_it\lambda_iP-c_i\alpha_iY_i$，返回 σ_i 给 A_{II}。那么 C_{II} 产生的签名 σ_i 是有效的，因为：

$$\begin{aligned}e(\sigma_i,P)&=e(\beta_iY_i-c_it\lambda_iP-c_i\alpha_iY_i,P)\\&=e(\beta_iY_i,P)e(-c_it\lambda_iP,P)e(-c_i\alpha_iY_i,P)\\&=e(Y_i,\beta_iP)(e(\lambda_iP,tP)e(Y_i,\alpha_iP))^{-c_i}\\&=(e(P_{pub},Q_i)e(W_i,Y_i))^{-c_i}e(H_2(ID_i),Y_i)\end{aligned}$$

最终伪造阶段：经过上述询问后，A_{II} 最终输出一个身份$\{ID_1^*,\cdots,ID_n^*\}$、公钥$\{Y_1^*,\cdots,Y_n^*\}$上的 n 个用户对消息$\{m_1^*,\cdots,m_n^*\}$的有效无证书指定验证者聚合签名 σ_V^*。如果存在一个 $ID_k^*\in\{ID_1^*,\cdots,ID_n^*\}$，$A_{II}$ 没有向 C_{II} 做过关于 ID_k^* 的秘密值询问和公钥替换询问，也没有做过(ID_k^*,Y_k^*,m_k^*)的单个签名询问，那么根据 Pointeheval and Stern 的 Forking lemma[17] 重复上述的交互过程，可以得到：$c_k^*=H_3(ID_k^*,m_k^*,Y_k^*)\neq\overline{c_k^*}=H_3(ID_k^*,m_k^*,Y_k^*)$。由此，可以得到：

$$\sigma_V^*=(e(P_{pub},Q_k^*)e(W_k,Y_k^*))^{-c_k^*s_V}\cdot\prod_{i=1,i\neq k}^{n}(e(P_{pub},Q_i^*)e(W_i,Y_i^*))^{-c_i^*s_V}\prod_{i=1}^{n}e(H_2(ID_i^*),Y_i^*)^{s_V}$$

$$\overline{\sigma_V^*}=(e(P_{pub},Q_k^*)e(W_k,Y_k^*))^{-\overline{c_k^*}s_V}\cdot\prod_{i=1,i\neq k}^{n}(e(P_{pub},Q_i^*)e(W_i,Y_i^*))^{-c_i^*s_V}\prod_{i=1}^{n}e(H_2(ID_i^*),Y_i^*)^{s_V}$$

上述两式相除后得到：

$$e(\sigma^*,Y_V)/e(\overline{\sigma^*},Y_V)=(e(P_{pub},Q_k^*)e(W_k,Y_k^*))^{-c_k^*s_V}/(e(P_{pub},Q_k^*)e(W_k,Y_k^*))^{-\overline{c_k^*}s_V}$$

由 $W_k=\alpha_kP_1,Y_k^*=s_kP_2,Q_k^*=\lambda_kP$，可以得到：$abP=((\overline{\sigma_V^*}-\sigma_V^*)(\overline{c_k^*}-c_k^*)^{-1}-\lambda_ktP)(\alpha_ks_k)^{-1}$。由此，$C_{II}$ 成功解决了一个给定的 CDH 问题。类似地，可得：

$$t'\leqslant 2t+2\sum_{i=1,i\neq4}^{10}t_iq_i+2t_A,$$

$$\varepsilon'\geqslant\varepsilon^2(1-q_2^{-1})^{2(q_1+q_8)}(C_{q_4}^n)^{-1}(nq_2^{-1})^2$$

定理 3 CTL-ASWUDV-2 方案满足不可传递性。

证明：对于$(m_i,ID_i,Y_i)(1\leqslant i\leqslant n)$，指定验证者可以使用自己的私钥 s_V 模拟产生与签名者不可区分的 CTL-ASWUDV。首先，指定验证者计算 $c_i=H_3(ID_i,m_i,Y_i)$和 $W_i=H_4(ID_i)(1\leqslant i\leqslant n)$，然后计算：$\sigma_V=\prod_{i=1}^{n}(e(P_{pub},Q_i)e(W_i,Y_i))^{-c_is_V}\prod_{i=1}^{n}e(H_2(ID_i),Y_i)^{s_V}$。显然，指定验证者产生的签名 σ_V 和签名者产生的签名 $\sigma_V'=e(\sigma,Y_V)$是不可区分的。因此，提出的 CTL-ASWUDV-2 方案满足不可传递性。

定理 4　CTL-ASWUDV-2 方案具有指定验证者特性。

证明：在本文提出的 CTL-ASWUDV-2 方案中，最后的签名验证通过 $\sigma_V=\prod_{i=1}^{n}(e(P_{pub},Q_i)e(W_i,Y_i))^{-c_i s_V}\prod_{i=1}^{n}e(H_2(ID_i),Y_i)^{s_V}$ 来进行。这个等式的验证需要使用验证者的私钥 s_V，因此其他任何人都不能验证签名 σ_V 的有效性。所以 CTL-ASWUDV-2 方案满足指定验证者验证的特性。

结束语　本文对最近指出的一个高效无证书广义指定验证者聚合签名存在的 3 个安全问题进行了研究，提出了一个能有效解决这 3 个问题的改进方案 CTL-ASWUDV-1。同时研究了无证书广义指定验证者聚合签名的安全模型，提出了一个更加高效的无证书广义指定验证者聚合签名方案 CTL-ASWUDV-2，并证明了该方案满足不可伪造性、不可传递性和指定验证者特性。本文也将提出的方案 CTL-ASWUDV-1 和 CTL-ASWUDV-2 与目前已有的同类方案在计算量和签名长度两方面进行了比较。比较结果显示，本文提出的 CTL-ASWUDV-1 和 CTL-ASWUDV-2 方案均优于其他同类方案。尤其是 CTL-ASWUDV-2 方案，除了预计算外，其聚合签名长度和指定验证者签名长度都是固定的 1 个元素，同时两个阶段的验证均只需 1 个双线性配对操作，因此可以高效地运用到资源受限的各种网络环境中，从而大大提高系统的运行效率并降低资源占有率。

参 考 文 献

[1] SHAMIR A. Identity-Based cryptosystems and signature schemes[J]. Workshop on the Theory & Application of Cryptographic Tecnhniques, 1984, 21(2): 47-53.

[2] AL-RIYAMI S S, PATERSON K G. Certificateless public key cryptography[J]. Lecture Notes in Computer Science, 2003, 133(2): 452-473.

[3] ZHANG L, ZHANG F T. A Method to Construct a Class of Certificateless Signature Schemes[J]. Chinese Journal of Computers, 2009, 32(5): 940-945. (in Chinese)
张磊，张福泰. 一类无证书签名方案的构造方法[J]. 计算机学报，2009，32(5)：940-945.

[4] CHEN H, ZHU C J, SONG R S. Efficient Certificateless Signature and Group Signature Schemes[J]. Journal of Computer Research and Development, 2010, 47(2): 231-237. (in Chinese)
陈虎，朱昌杰，宋如顺. 高效的无证书签名和群签名方案[J]. 计算机研究与发展，2010，47(2)：231-237.

[5] DU H Z, WEN Q Y. Certificateless proxy multi-signature[J]. Information Sciences, 2014, 276(c): 21-30.

[6] BONEH D, GENTRY C, LYNN B, SHACHAM H. Aggregate and verifiably encrypted signatures from bilinear maps[J]. Lecture Notes in Computer Science, 2003, 2656(1): 416-432.

[7] ZHANG L, ZHANG F T. A new certificateless aggregate signature scheme[J]. Computer Communications, 2009, 32(6): 1079-1085.

[8] DU H Z, HUANG M J, WEN Q Y. Efficient and provably-secure certificateless aggregate signature scheme[J]. Acta Electronica Sinica, 2013, 41(1): 72-76. (in Chinese)
杜红珍，黄梅娟，温巧燕. 高效的可证明安全的无证书聚合签名方案[J]. 电子学报，2013，41(1)：72-76.

[9] CHEN M. Improved certificateless aggregate signature with constant length[J]. Application Research of Computers, 2016(1): 271-275. (in Chinese)
陈明. 改进的签名长度固定的无证书聚合签名方案[J]. 计算机应用研究，2016(1)：271-275.

[10] ZHOU M, ZHANG M W, WAN C Z, et al. CCLAS: A Practical and Compact Certificateless Aggregate Signature with Share Extraction[J]. International Journal of Network Security, 2014, 16(3): 174-181.

[11] CHEN H, WEI S M, ZHU C J, et al. Secure Certificateless Aggregate Signature Scheme[J]. Journal of Software, 2015, 26(5): 1173-1180. (in Chinese)
陈虎，魏仕民，朱昌杰，等. 安全的无证书聚合签名方案[J]. 软件学报，2015，26(5)：1173-1180.

[12] ZHOU Y W, YANG B, ZHANG W Z. Efficient and Provide Security Certificateless Aggregate Signature Scheme[J]. Journal of Software, 2015, 26(12): 3204-3214. (in Chinese)
周彦伟，杨波，张文政. 高效可证安全的无证书聚合签名方案[J]. 软件学报，2015，26(12)：3204-3214.

[13] ZHANG Y L, ZHOU D R, LI C Y, et al. Certificateless-based efficient aggregate signature scheme with universal designated verifier[J]. Journal on Communications, 2015, 36(2): 1-8. (in Chinese)
张玉磊，周冬瑞，李臣意，等. 高效的无证书广义指定验证者聚合签名方案[J]. 通信学报，2015，36(2)：1-8.

[14] DU H Z. Attacks on a Certificateless Aggregate Signature Scheme with Universal Designated Verifier[J]. Henan Science, 2015, 33(7): 1087-1090. (in Chinese)
杜红珍. 无证书广义指定验证者聚合签名方案的攻击[J]. 河南科学，2015，33(7)：1087-1090.

[15] QIN Y L, WU X P. Efficient certificateless sequential multi-signature scheme[J]. Journal on Communications, 2013, 34(7): 105-110. (in Chinese)
秦艳琳，吴晓平. 高效的无证书有序多重签名方案[J]. 通信学报，2013，34(7)：105-110.

[16] LIU E G, WANG X, ZHOU H J, et al. Improved Certificateless Proxy Blind Signature Scheme[J]. Computer Science, 2016, 43(8): 92-94. (in Chinese)
刘二根，王霞，周华静，等. 改进的无证书代理盲签名方案[J]. 计算机科学，2016，43(8)：92-94.

[17] POINTEHEVAL D, STERN J. Security arguments for digital signatures and blind signatures[J]. Journal of Cryptology, 2000, 13(3): 361-396.