计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2020.20200421
57(10): 2177-2187, 2020

工业物联网中服务器辅助且可验证的属性基签名方案

张应辉[1,2] 贺江勇[1,2] 郭 瑞[1,2] 郑 东[1,2,3]
[1](西安邮电大学网络空间安全学院 西安 710121)
[2](无线网络安全技术国家工程实验室(西安邮电大学) 西安 710121)
[3](卫士通摩石实验室 北京 100070)
[1](yhzhaang@163.com)

Server-Aided and Verifiable Attribute-Based Signature for Industrial Internet of Things

Zhang Yinghui[1,2], He Jiangyong[1,2], Guo Rui[1,2], and Zheng Dong[1,2,3]
[1](*School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an* 710121)
[2](*National Engineering Laboratory for Wireless Security (Xi'an University of Posts and Telecommunications), Xi'an* 710121)
[3](*Westone Cryptologic Research Center, Beijing* 100070)

Abstract Industrial Internet of things (IIoT) devices encounter problems such as data authentication and privacy protection when collecting and storing data through the cloud. Attribute-based signature (ABS) can not only realize the data authentication, but also protect the identity privacy of the signer. In the existing server-aided ABS (SA-ABS) schemes, the computational overhead of the signer and the verifier is reduced with the help of the server, and the security of the server-aided verification phase is guaranteed by the defense of collusion attack of the signer and the server. However, none of the existing SV-ABS schemes can verify the validity of partial signature generated by the server, which will lead to a potential risk of partial signature forgery by the server. To overcome this challenge, a novel server-aided and verifiable ABS (SA-VABS) scheme is proposed in this paper, which not only reduces the computational overhead of the signer and the verifier, but also ensures the security of the server-aided verification phase by resisting the collusion attack of the signer and the server. The most important is that the scheme could verify the validity of partial signature generated by the server, so as to ensure the security of generation phase of the server-aided signature. Finally, our formal security analysis verifies the security of the SA-VABS scheme, and simulation experiments

收稿日期:2020-06-10;**修回日期**:2020-07-30
基金项目:国家重点研发计划项目(2017YFB0802000);国家自然科学基金项目(61772418,61671377,61802303);陕西省创新能力支撑计划项目(2020KJXX-052);陕西省特支计划青年拔尖人才支持计划项目;陕西省重点研发计划项目(2019KW-053,2020ZDLGY08-04);陕西省自然科学基础研究计划项目(2019JQ-866);四川省科技计划项目(2017GZDZX0002);青海省基础研究计划项目(2020-ZJ-701);西邮新星团队支持计划项目(2016-02)
This work was supported by the National Key Research and Development Program of China (2017YFB0802000), the National Natural Science Foundation of China (61772418, 61671377, 61802303), the Innovation Capability Support Program of Shaanxi (2020KJXX-052), the Shaanxi Special Support Program Youth Top-notch Talent Program, the Key Research and Development Program of Shaanxi (2019KW-053, 2020ZDLGY08-04), the Natural Science Basic Research Plan in Shaanxi Province of China (2019JQ-866), the Sichuan Science and Technology Program (2017GZDZX0002), the Basic Research Program of Qinghai Province (2020-ZJ-701), and the New Star Team Program of Xi'an University of Posts and Telecommunications (2016-02).

as well as comparative analysis indicate that the SA-VABS scheme improves security while ensuring efficiency.

Key words attribute-based signature (ABS); server-aided (SA); collusion attack; verifiable; privacy protection

摘 要 工业物联网(industrial Internet of things, IIoT)设备通过云端收集和存储数据时,会遇到数据认证和隐私保护等问题.属性基签名(attribute-based signature, ABS)不仅可以实现数据认证,而且可以保护签名者的身份隐私.目前存在的SA-ABS(server-aided ABS)方案中,借助服务器减小了签名者和验证者的计算开销,而且通过抵抗签名者和服务器的共谋攻击保证了服务器辅助验证阶段的安全性.但是,现有的SA-ABS方案都不能对服务器产生的部分签名进行有效性验证,所以存在服务器对部分签名伪造的安全隐患.为克服这一挑战,提出一种服务器辅助且可验证的属性基签名(server-aided and verifiable ABS, SA-VABS)方案,该方案不仅减小了签名者和验证者的计算开销,而且通过抵抗签名者和服务器的共谋攻击来保证服务器辅助验证阶段的安全性,最重要的是对服务器产生的部分签名进行了有效性验证,从而保证了服务器辅助签名产生阶段的安全性.形式化安全性分析表明SA-VABS方案是安全的.仿真实验和对比分析表明SA-VABS方案在保证效率的同时提高了安全性.

关键词 属性基签名;服务器辅助;共谋攻击;可验证;隐私保护

中图法分类号 TP309

工业物联网(industrial Internet of things, IIoT)作为新一代信息技术的重要组成部分,它按照约定的信息交换协议通过传感器设备连接各种网络,以实现智能识别、跟踪、监视、定位和管理[1-2].随着IIoT的普及,其安全问题也越来越受到研究者的关注[3].由于IIoT环境的开放性,数据经过公共信道传输时可能被恶意的敌手伪造或篡改.另外,涉及IIoT情况的数据包含用户身份的敏感信息,容易造成用户身份隐私的泄露.因此,如何确保IIoT系统的数据安全和用户身份隐私非常具有挑战性[4].属性基签名(attribute-based signature, ABS)既可以保护用户身份隐私,又可以实现数据认证.近年来,大多数国内外学者基于ABS,围绕其签名和验证阶段的计算开销、灵活的访问结构、服务器辅助(server-aided, SA)签名和验证阶段的安全性等问题进行了研究,形成了较为丰富的理论成果.在减小签名者和验证者的计算开销方面,一个可行的方法是采用SA技术[5]将繁重的计算委托给服务器.但是针对SA签名产生和验证阶段安全性的研究,大多数学者只是通过抵抗签名者和服务器的共谋攻击保证了SA验证阶段的安全性.如何抵抗服务器对部分签名的伪造,保证SA签名产生阶段的安全性具有重要的研究意义,需进一步研究.

本文的主要贡献包括3个方面:

1) 提出了一种服务器辅助且可验证的ABS(server-aided and verifiable ABS, SA-VABS)方案,通过对服务器产生的部分签名进行有效性验证,抵抗了服务器对部分签名的伪造;

2) 提出的SA-VABS方案可以抵抗签名者和服务器的共谋攻击,即签名者勾结服务器并指导服务器产生一个无效的中介签名去欺骗验证者;

3) 对提出的SA-VABS方案进行了严格的安全性分析,并在理论上和实验上评估了其性能,最后通过对比分析说明了SA-VABS方案是安全高效的.

1 相关工作

ABS的概念是从属性基加密(attribute-based encryption, ABE)演变而来的[6],正式定义首先由Maji等人[7]提出,该方案仅能在一般的群模型下给出安全性证明.为了提高方案的安全性,Li等人[8]提出两种支持门限访问结构的ABS方案,并且在随机预言模型和标准模型下分别证明了方案的不可伪造性;为了减小系统的存储负担,Ge等人[9]在标准模型下提出一种高效的ABS方案,该方案的签名长度是恒定的,不会随着属性的数量发生变化;为了实现更灵活的访问结构,Su等人[10]在2014年提出一种支持树形访问结构的ABS方案.然而,这5种ABS方案的一个共同问题是签名者和验证者的计算开销随着属性的数量呈线性增长.

外包计算基于云计算[11],最早是由 Hohenberger 等人[12]提出;由于现有 ABS 方案中签名生成算法需要大量的指数运算,Chen 等人[13]首先提出外包 ABS(outsourced ABS, OABS)方案,将签名产生算法的主要计算开销委托给服务器;Ren 等人[14]在 2018 年提出另一种可以验证外包签名有效性的 OABS 方案;最近,Mo 等人[15]也提出了一种应用于医疗系统中的 OABS 方案,该方案支持更灵活的访问结构;在 2019 年 Sun 等人[16]提出一种外包的分散式多属性机构 ABS(outsourced decentralized multi-authority,ODMA-ABS)方案,该方案相关的公私钥由多个属性机构交互来生成,提高了 OABS 方案的安全性.然而,这些 OABS 方案只减小了签名产生阶段的计算开销,并没有减小签名验证阶段的计算开销.

为了提高 ABS 方案签名验证的效率,Matsumoto 等人[17]首次提出 SA 的概念,可以将验证者的繁重计算委托给服务器.2014 年 Wang 等人[18]首先提出 SA 验证的 ABS(attribute-based server-aided verification signature, ABSAVS)方案,借助服务器减轻了验证者的计算开销.最近,Cui 等人[19]第 1 次提出可撤销的 SA-ABS(server-aided ABS with revocation, SA-ABSR)方案,该方案借助服务器同时减小了签名者和验证者的计算开销,而且支持用户撤销的功能,但是不能抵抗签名者和服务器的共谋攻击.基于此,Xiong 等人[20]提出另一种 SA-ABS 方案,不仅抵抗签名者和服务器的共谋攻击,而且实现了更灵活的 LSSS 访问结构.但是,他们提出的 SA-ABS 方案都不能验证部分签名(即服务器产生的签名)的有效性,因此不能抵抗服务器对部分签名的伪造.

综上,现有的 SA-ABS 方案中,对于计算开销的研究比较理想的技术是 SA 技术.而对于 SA 阶段的安全性问题,大多数学者主要是围绕 SA 验证阶段的安全性研究,通过抵抗签名者和服务器的共谋攻击来保证 SA 验证阶段的安全性.如何抵抗服务器对部分签名的伪造,确保 SA 签名产生阶段的安全性还存在许多问题,成为本文的主要研究工作之一.

2 基础理论

本节介绍了文中用到的主要符号和基础知识.

2.1 符号说明

文中所用到的主要符号及解释说明如表 1 所示:

Table 1 The Main Notations and Description

表 1 主要符号及说明

Notations	Description
$\Gamma_{k,S}$	threshold access structure
par	public parameter
msk	master private key
psk	partial signing key
sk	signing key
tk	transformed key
σ	signature
σ'	partial signature
$\bar{\sigma}$	transformed signature
$\hat{\sigma}_1$	intermediate signature

2.2 基础知识

定义 1. 双线性映射.给定 G,G_1 为 2 个阶为大素数 p 的乘法循环群,g 是 G 的生成元,Z_p 为有限域.一个映射 $e:G\times G\rightarrow G_1$ 如果满足 3 个特性,则称该映射为双线性映射:

1) 双线性.$\forall a,b\in Z_p$ 以及 $g_1,g_2\in G$,等式:$e(g_1^a,g_2^b)=e(g_1,g_2)^{ab}$ 成立.

2) 非退化性.$e(g,g)\neq 1$.

3) 可计算性.对任意 $g_1,g_2\in G$,存在可以计算 $e(g_1,g_2)$ 的高效算法.

定义 2. n-DHE 问题.对任意 $a\in Z_p,g\in G$,给定 $g,g^a,g^{a^2},\cdots,g^{a^n},g^{a^{n+2}},\cdots,g^{a^{2n}}$,计算 $g^{a^{n+1}}$ 的值.如果不存在能够以不可忽略的概率优势解决 n-DHE 问题的多项式时间算法,则称 n-DHE 问题是困难的.

定义 3. 拉格朗日插值.设 $p(x)$ 是有限域 Z_p 上 $n-1$ 阶多项式,定义 $\Omega\in\{1,2,\cdots,n\}$,计算 $p(x)$:

$$p(x)=\prod_{i=1}^{n}p(i)\Delta_i^{\Omega}(x),$$

其中,拉格朗日系数是 $\Delta_i^{\Omega}(x)=\prod_{j\in\Omega,j\neq i}\frac{x-j}{i-j}$.

定义 4. 门限访问结构.门限访问结构是一个单调的布尔函数,可以描述为

$$\Gamma_{k,S}(A)=\begin{cases}1, & |A\cap S|\geqslant k,\\ 0, & \text{otherwise},\end{cases}$$

其中,A 是用户的属性集,S 是访问结构的属性集,k 是访问结构中所指定的门限值.当 $\Gamma_{k,S}(A)=1$ 时,我们认为属性集 A 满足访问结构 $\Gamma_{k,S}(A)$.

3 系统模型及安全模型

本节介绍了 SA-VABS 方案的系统模型和安全

性模型,包括正确性、不可伪造性、抗共谋攻击以及匿名性.

3.1 系统模型

我们提出的 SA-VABS 方案包含 4 个实体:属性机构、签名者、验证者以及服务器.系统结构图如图 1 所示:

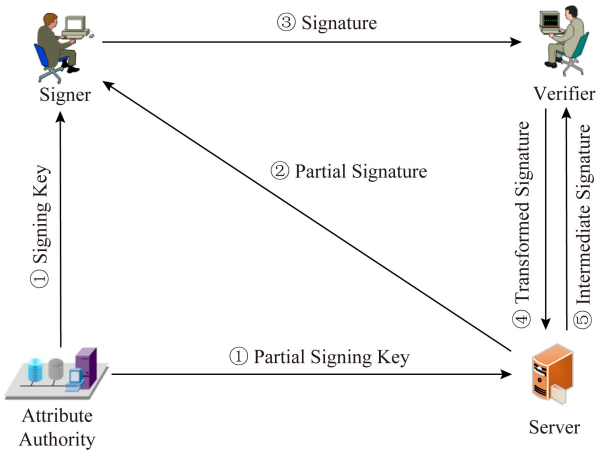

Fig. 1 The system architecture

图 1 系统结构图

在阶段①,属性机构首先产生公共参数 par 和主私钥 msk,然后根据用户的属性集合分别为服务器和签名者生成部分签名钥 psk 和签名钥 sk.在阶段②,服务器利用部分签名钥 psk 和访问结构 $\Gamma_{k,S}$ 对消息 m 进行签名,生成部分签名 σ' 并发送给签名者.在阶段③,签名者首先验证部分签名 σ' 的有效性,如果有效则签名者利用签名钥 sk 以及部分签名 σ' 生成完整的签名 σ 并发送给验证者,否则终止签名.在阶段④,验证者随机选取转换钥 tk,然后产生转换签名 $\bar{\sigma}$ 发送给服务器.在阶段⑤,服务器根据转换签名 $\bar{\sigma}$ 生成一个中介签名 $\hat{\sigma}_1$ 返回给验证者.最后,验证者基于转换签名 $\bar{\sigma}$ 和转换钥 tk 验证签名 σ 的有效性.我们的 SA-VABS 方案用 7 个算法来描述:Setup, KeyGen, SSign, USign, Transform, SVerify 以及 UVerify.

Setup.属性机构将安全参数 λ 作为输入,输出公共参数 par 和主私钥 msk.

KeyGen.属性机构将公共参数 par、主私钥 msk 以及用户属性集 A 作为输入,分别为服务器和签名者输出部分签名钥 psk 和签名钥 sk.

SSign.服务器将公共参数 par、部分签名钥 psk、消息 m 以及访问结构 $\Gamma_{k,S}$ 作为输入,输出部分签名 σ' 和相应的验证信息 W_1 及 W_2.

USign.签名者将公共参数 par、签名钥 sk、验证信息 W_1 及 W_2、部分签名 σ' 作为输入,如果部分签名 σ' 验证有效则输出完整的签名 σ.

Transform.验证者将公共参数 par、转换钥 tk 以及签名 σ 作为输入,输出转换签名 $\bar{\sigma}$.

SVerify.服务器将公共参数 par、转换签名 $\bar{\sigma}$ 作为输入,输出中介签名 $\hat{\sigma}_1$.

UVerify.验证者将公共参数 par、转换钥 tk 以及中介签名 $\hat{\sigma}_1$ 作为输入,输出 true 或者 false.

3.2 安全模型

提出的 SA-VABS 方案的安全性需要满足正确性、不可伪造性、抗共谋攻击以及匿名性.

1) 正确性.SA-VABS 方案的正确性是指对任意消息 m,任何满足访问结构 $\Gamma_{k,S}$ 的属性集 A,运行算法 Setup, KeyGen, SSign, USign, Transform 以及 SVerify,最后 UVerify 算法输出的结果为 true,则说明方案 SA-VABS 满足正确性.

2) 不可伪造性.SA-VABS 方案的不可伪造性表现为一个伪造者 $\mathcal{F}$ 和一个挑战者 $\mathcal{C}$ 之间的交互游戏.

Initialize.伪造者 $\mathcal{F}$ 向挑战者 $\mathcal{C}$ 宣布要攻击的访问结构 $\Gamma_{k,S}$.

Setup.挑战者 $\mathcal{C}$ 产生公共参数 par 和主私钥 msk,然后挑战者 $\mathcal{C}$ 将公共参数 par 发送给伪造者 $\mathcal{F}$.

Queries.伪造者 $\mathcal{F}$ 向挑战者 $\mathcal{C}$ 进行 4 方面询问,$\mathcal{C}$ 首先初始化一个空的列表 L.

① 部分签名钥询问(partial signing key oracle).具有属性集 A 的伪造者 $\mathcal{F}$ 进行部分签名钥询问,然后 $\mathcal{C}$ 在列表 L 中检查是否存在元组 (A, psk, sk).如果存在则返回 psk 给 $\mathcal{F}$,否则运行 KeyGen 算法并且将新元组 (A, psk, sk) 添加到 L 之后返回 psk 给 $\mathcal{F}$.

② 签名钥询问(signing key oracle).具有属性集 A 的伪造者 $\mathcal{F}$ 进行签名钥询问,然后 $\mathcal{C}$ 在列表 L 中检查是否存在元组 (A, psk, sk).如果存在则返回 sk 给 $\mathcal{F}$,否则运行 KeyGen 算法并且将新元组 (A, psk, sk) 添加到 L 之后返回 sk 给 $\mathcal{F}$.

③ 签名询问(signing oracle).伪造者 $\mathcal{F}$ 选择一个消息 m 和一个访问结构 $\Gamma_{k,S}$,然后 $\mathcal{C}$ 运行 SSign 以及 USign 算法去产生相应的签名 σ 并发送给 $\mathcal{F}$.

④ 签名验证询问(UVerify oracle).伪造者 $\mathcal{F}$ 用签名 σ 发起签名验证询问,挑战者 $\mathcal{C}$ 运行 Transform 算法并返回转换签名 $\bar{\sigma}$ 给 $\mathcal{F}$.然后 $\mathcal{F}$ 返回中介签名 $\hat{\sigma}_1$ 给 $\mathcal{C}$.最后,$\mathcal{C}$ 运行 UVerify 算法将验证结果返回给 $\mathcal{F}$.

Forgery.伪造者 $\mathcal{F}$ 对消息 m^* 在访问结构 Γ_{k^*,S^*} 下产生签名 σ^*,如果满足以下 2 个条件:m^* 和 Γ_{k^*,S^*} 从未被 $\mathcal{F}$ 询问;$\mathcal{C}$ 收到签名 σ^* 后,计算转换签名 $\bar{\sigma}^*$ 发送给 $\mathcal{F}$,然后 $\mathcal{F}$ 返回中介签名 σ_1^* 给 $\mathcal{C}$,最后运行 UVerify 算法返回的结果为 true,则认为 $\mathcal{F}$ 赢得该游戏.

3) 抗共谋攻击.签名者和服务器的共谋攻击是指签名者使用一个伪造的消息 m^* 产生签名,然后勾结服务器基于 m^* 执行 SVerify 算法去产生中介签名,但是服务器欺骗验证者中介签名是基于 m 产生的.这样就可能导致验证者将一个无效的签名通过 UVerify 算法.因此,包含消息 m 的验证部分不能由服务器来验证,如果这部分验证由验证者来完成,则可以有效抵抗签名者和服务器的共谋攻击.

4) 匿名性.对于任何消息 m,公共参数和主私钥 $Setup(1^\lambda)\rightarrow(par,msk)$,满足访问结构 $\Gamma_{k,S}$ 的 2 个属性集合 A_1 和 A_2,对应的签名钥 $KeyGen(par,msk,A_1)\rightarrow(psk_1,sk_1)$ 和 $KeyGen(par,sk,A_2)\rightarrow(psk_2,sk_2)$,如果生成的 2 个签名 $USign(par,sk_1,\sigma_1')\rightarrow\sigma_1^*$ 和 $USign(par,sk_2,\sigma_2')\rightarrow\sigma_2^*$ 是不可区分的,则称提出的方案 SA-VABS 具有匿名性.

4 服务器辅助且可验证的属性基签名方案

本节我们给出 SA-VABS 方案的具体构造,包括 7 个算法:Setup,KeyGen,SSign,USign,Transform,SVerify,UVerify.

Setup.该算法用于产生系统公共参数和主私钥,输入安全参数 λ,算法运行步骤为:

1) 定义 U 为系统中的属性集合,M 定义为长度最大为 m 的明文空间.Ω 为默认属性集,其中 $|\Omega|=n$.假设 $U\cup\Omega$ 中的每一个属性都是 Z_p 中的元素.

2) 设 G,G_1 为 p 阶乘法循环群,定义一个双线性映射 $e:G\times G\rightarrow G_1$,其中 g 是 G 的生成元.随机选取 $\alpha\in Z_p$,计算 $Z=e(g,g)^\alpha$.

3) 随机选取 $v_0\in Z_p$,$\boldsymbol{V}=(v_1,v_2,\cdots,v_N)^{\mathrm{T}}\in Z_p^N$,其中 $N=2n+1$,并计算 $h_0=g^{v_0}$,$h_i=g^{v_i}$,$i\in\{1,2,\cdots,N\}$.

4) 从 Z_p 中选取 n 个元素,令 $D=\{d_1,d_2,\cdots,d_n\}$ 作为虚拟属性集.

5) 另外,从 G 中随机选取 $u_0,u_1,\cdots,u_{n_m}$,定义 Hash 函数 $H(m)=u_0\prod_{j=1}^{n_m}u_j^{m_j}$,$m_i$ 是消息 m 的第 i 个字节.

因此,系统的主私钥为 $msk=\alpha$,公共参数为 $par=(g,G,G_1,e,p,Z,h_0,h_1,\cdots,h_N,u_0,u_1,\cdots,u_{n_m})$.

KeyGen.输入公共参数 par、主私钥 msk 以及用户属性集 A,随机选取 $\beta,a_1,a_2,\cdots,a_{n-1}\in Z_p$,定义一个多项式 $q(x)=\sum_{i=1}^{n-1}a_i^{x^i}+\beta$ 使得 $q(0)=\beta$.

1) 对虚拟属性 $d\in AUD$,随机选择 $r_d\in Z_p$,计算:

$$P_{d,1}=g^{q(x)}\times h_0^{r_d},$$

$$P_{d,2}=g^{r_d},$$

$$P_{d,i}=(h_1^{-d^i}\times h_{i+1})^{r_d},i\in\{1,2,\cdots,N-1\},$$

2) 对属性 $w\in A\cup\Omega$,随机选择 $r_w\in Z_p$,计算:

$$P_{w,1}=g^{\alpha-\beta}\times h_0^{r_w},$$

$$P_{w,2}=g^{r_w},$$

$$P_{w,i}=(h_1^{-w^i}\times h_{i+1})^{r_w},\ i\in\{1,2,\cdots,N-1\}.$$

因此,属性机构为服务器和用户产生的部分签名钥以及签名钥分别是

$$psk=\{P_{d,1},P_{d,2},\{P_{d,i}\}_{i\in\{1,2,\cdots,N-1\}}\},$$

$$sk=\{g^{\alpha-\beta},P_{w,1},P_{w,2},\{P_{w,i}\}_{i\in\{1,2,\cdots,N-1\}}\}.$$

SSign.该算法用于将签名过程的繁重计算委托给服务器.输入公共参数 par、部分签名钥 psk、消息 m 以及访问结构 $\Gamma_{k,S}$,算法运行为:

1) 随机选择任意属性集 $S'\subset A\cap S$,令 $|S'|=k$,进一步选择一个虚拟属性集 $D'\subset D$,令 $|D'|=n-k$.

2) 定义一个向量 $\boldsymbol{b}=(b_1,b_2,\cdots,b_N)$,计算多项式:

$$\varphi(z)=\prod_{d\in S\cup D'}(z-d)=\sum_{i=1}^{N}b_iz^{i-1},$$

当 $|S\cup D'|+2\leqslant i\leqslant N$ 时,设置 $b_i=0$.

3) 对每一个虚拟属性 $d\in S'\cup D'$,计算:

$$P'_{d,1}=P_{d,1}\times\prod_{i=1}^{N-1}P_{d,i}^{b_i+1}=g^{q(d)}\times\left(h_0\times\prod_{i=1}^{N}h_i^{b_i}\right)^{r_d},$$

$$P'_1=\prod_{d\in S'\cup D'}P'^{\Delta_w^{S'\cup D'}(0)}_{d,1}=g^\beta\times\left(h_0\times\prod_{i=1}^{N}h_i^{b_i}\right)^r,$$

$$P'_2=\prod_{d\in S'\cup D'}P_{d,2}^{\Delta_w^{S'\cup D'}(0)}=g^r,$$

其中,$r=\sum_{d\in S'\cup D'}\Delta_w^{S'\cup D'}(0)\times r_d$.

4) 随机选择 $s_0,s_1\in Z_p$,计算:

$$\sigma'_0=P'_1\times\left(h_0\times\prod_{i=1}^{N}h_i^{b_i}\right)^{s_0}\times H(m)^{s_1},$$

$$\sigma'_1=P'_2\times g^{s_0},$$

$$\sigma'_2=g^{s_1}.$$

5) 另外,计算相应的验证信息 W_1 和 W_2:

$$W_1=e(g,\sigma'_0),$$

$$W_2=\sum_{d\in S'\cup D'}\left[Z\times e\left(\sigma'_1,h_0\times\prod_{i=1}^{N}h_i^{b_i}\right)\times e(\sigma'_2,H(m))\right].$$

最终,消息 m 的部分签名为 $\sigma'=\{m,\Gamma_{k,S},\sigma'_0,\sigma'_1,\sigma'_2,W_1,W_2\}$.

USign.输入公共参数 par、签名钥 sk 以及部分签名 σ',算法运行为:

1) 计算:$Z_1=e(g,g^{\alpha-\beta})$.

2) 验证下列等式是否成立:

$$Z_1\times W_1=W_2.$$

3) 如果等式成立,计算:

$$P'_{w,1}=P_{w,1}\times\prod_{i=1}^{N-1}P_{w,i}^{b_i+1}=g^{\alpha-\beta}\times\left(h_0\times\prod_{i=1}^{N}h_i^{b_i}\right)^{r_w}.$$

4) 随机选择 $s\in Z_p$,计算:

$$\sigma_0=\sigma'_0\times P'_{w,1}\times H(m)^s,$$

$$\sigma_1=\sigma'_1\times P_{w,2},$$

$$\sigma_2=\sigma'_2\times g^s.$$

最终,消息 m 的签名为 $\sigma=\{m,\Gamma_{k,S},\sigma_0,\sigma_1,\sigma_2\}$.

Transform.输入公共参数 par 和签名 σ,算法运行如下:

1) 随机选择 $t\in Z_p$.

2) 计算 $\tilde{\sigma}_0=\sigma_0^t,\tilde{\sigma}_1=\sigma_1^t,\tilde{\sigma}_2=\sigma_2^t$.

最后输出转换签名为 $\tilde{\sigma}=\{m,\Gamma_{k,S},\tilde{\sigma}_0,\tilde{\sigma}_1,\tilde{\sigma}_2\}$,转换钥为 $tk=t$.

SVerify.该算法用于将验证过程的繁重计算委托给服务器.输入公共参数 par 和转换签名 $\tilde{\sigma}$,算法运行产生中介签名 $\hat{\sigma}_1$:

$$\frac{e(g,\tilde{\sigma}_0)}{e\left(h_0\prod_{i=1}^{N}h_i^{b_i},\tilde{\sigma}_1\right)}=\hat{\sigma}_1.$$

收到 $\hat{\sigma}_1$ 之后,服务器将 $\hat{\sigma}_1$ 发送给验证者.

UVerify.从服务器收到中介签名 $\hat{\sigma}_1$ 之后,算法运行为:

1) 验证者首先计算 $\hat{\sigma}_2=e(H(m),\tilde{\sigma}_2)\times Z^t$.

2) 验证 $\hat{\sigma}_1=\hat{\sigma}_2$ 是否成立.

如果 $\hat{\sigma}_1=\hat{\sigma}_2$ 成立,则输出 true,表明签名是有效的;否则输出 false.

5 安全性分析

本节主要参考文献[20-21],对提出的 SA-VABS 的正确性、不可伪造性、抗共谋攻击以及匿名性进行了详细的安全性分析.

5.1 正确性

SA-VABS 方案的正确性证明:

$$\hat{\sigma}_1=\frac{e(g,\tilde{\sigma}_0)}{e\left(h_0\prod_{i=1}^{N}h_i^{b_i},\tilde{\sigma}_1\right)}=\frac{e(g,(\sigma'_0P'_{w,1}H(m)^s)^t)}{e\left(h_0\prod_{i=1}^{N}h_i^{b_i},(\sigma'_1P_{w,2})^t\right)}=$$

$$\frac{e\left(g,\left(P'_1\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{s_0+r_w}H(m)^{s_1}g^{\alpha-\beta}H(m)^s\right)^t\right)}{e\left(h_0\prod_{i=1}^{N}h_i^{b_i},(P'_2g^{s_0}g^{r_w})^t\right)}=$$

$$\frac{e\left(g,\left(g^{\alpha}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}H(m)^{s_1+s}\right)^t\right)}{e\left(\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w},g^t\right)}=$$

$$\frac{e\left(g,\left(\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}\right)^t\right)e(g,(g^{\alpha}H(m)^{s_1+s})^t)}{e\left(\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w},g^t\right)}=$$

$$e(g,(g^{\alpha}H(m)^{s_1+s})^t),$$

$$\hat{\sigma}_2=e(H(m),\tilde{\sigma}_2)\times Z^t=$$

$$e(H(m),\sigma_2^t)\times(e(g,g)^{\alpha})^t=$$

$$e(H(m),(g^{s_1+s})^t)\times(e(g,g)^{\alpha})^t=$$

$$e(H(m)^{s_1+s},g^t)\times(e(g,g)^{\alpha})^t=$$

$$e(g,(H(m)^{s_1+s})^t)\times e(g,(g^{\alpha})^t)=$$

$$e(g,(g^{\alpha}\times H(m)^{s_1+s})^t).$$

由于等式 $\hat{\sigma}_1=\hat{\sigma}_2$ 验证成立,所以我们提出的 SA-VABS 方案满足正确性.

5.2 不可伪造性

假设伪造者 $\mathcal{F}$ 允许挑战者构造一个算法 C,该算法根据 $(g,g^{\alpha},g^{\alpha^2},\cdots,g^{\alpha^N},g^{\alpha^{N+2}},\cdots,g^{\alpha^{2N}})$ 计算 $g^{\alpha^{N+1}}$,其中,$N=2n+1$.

Initialize.定义向量 $\boldsymbol{a}=(a^1,a^2,\cdots,a^N)$、$\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_N)$ 以及与多项式 $\varphi(z)$ 相关的向量 $\boldsymbol{b}=(b_1,b_2,\cdots,b_N)$,$g_i=g^{a^i},i\in\{1,2,\cdots,2N+1\}\backslash\{N+1\}$.伪造者 $\mathcal{F}$ 指定目标访问结构 $\Gamma^*_{k^*,S^*}$ 发送给 C.

Setup.首先选择一个包含 n 个元素的虚拟属性集 D 以及一个子集 $D'\subset D$,其中 $|D'|=n-k^*$.

1) 挑战者 $\mathcal{C}$ 随机选择 $\theta_0,\delta_0\in Z_p$,向量 $\boldsymbol{\theta}\in Z_p^N$,计算 $h_0=g^{\theta_0}\times g^{-\langle\boldsymbol{a},\boldsymbol{b}\rangle},h_i=g^{\boldsymbol{a}}\times g^{\boldsymbol{\theta}},e(g,g)^{\alpha}=e(g_1,g_N)\times e(g,g)^{\delta_0}$.所以相应的主私钥为 $g^{\alpha}=g^{\alpha^{N+1}}\times g^{\delta_0}$.

2) 挑战者 $\mathcal{C}$ 定义向量 $\boldsymbol{U}=(u_0,u_1,\cdots,u_{n_m})$,随机选择变量 $\nu\in[1,n_m]$ 以及 $\eta_0,\lambda_0,\{\eta_j\},\{\lambda_j\}\in Z_p$,其中 $j\in[1,n_m]$.定义 $u_0=g^{a(\eta_0-2\nu q)}\times g^{\lambda_0}$($q$ 为签名询问的最大次数),$u_j=g^{a\eta_j}\times g^{\lambda_j}$.

3) 挑战者 $\mathcal{C}$ 选择一个 Hash 函数 H 并设置公共参数为 $par=(g,e,h_0,h_1,\cdots,h_N,u_0,u_1,\cdots,u_{n_m},\boldsymbol{U},D,H)$.

Queries.伪造者 $\mathcal{F}$ 向挑战者 $\mathcal{C}$ 进行询问:

1) 部分签名钥询问(partial signing key oracle).从 $\mathcal{F}$ 收到属性集 A 之后,C 在列表 L 中检查是否存在元组(A,psk,sk),如果存在则返回 psk 给 $\mathcal{F}$.否则随机选择 $\beta\in Z_p$,运行 KeyGen 算法并将新元组(A,psk,sk)添加到 L 之后返回 psk 给 $\mathcal{F}$.

2) 签名钥询问(signing key oracle).从 $\mathcal{F}$ 收到属性集 A 之后,C 在列表 L 中检查是否存在元组(A,psk,sk),如果存在则返回 sk 给 $\mathcal{F}$.否则随机选择 $\beta\in Z_p$,计算如下:

对任意 $w\in(A\cap S^*)\cup D'$,定义 $\boldsymbol{M}_w^n=(1,w,w^2,\cdots,w^{n-1})^{\mathrm{T}}$.对于任意满足 $|A\cap S^*|<k^*$ 的属性集 A,$\mathcal{F}$ 可以获得私钥.由于$(A\cap S^*)\cup D'$的基数严格小于 $n(|D'|=n-k^*)$,向量 $\boldsymbol{M}_0^n=(1,0,\cdots,0)^{\mathrm{T}}$ 不在 $\boldsymbol{M}_w^n$ 的范围内.C 选择一个向量 $\boldsymbol{\tau}$ 使得$\langle\boldsymbol{M}_w^n,\boldsymbol{\tau}\rangle=0$ 但$\langle\boldsymbol{M}_0^n,\boldsymbol{\tau}\rangle\neq0$,记 $\phi=\langle\boldsymbol{M}_0^n,\boldsymbol{\tau}\rangle$.$C$ 随机选择一个向量 $\boldsymbol{p}=(p_1,p_2,\cdots p_n)^{\mathrm{T}}$ 和 $\psi=\dfrac{\alpha-\beta-p_1}{\phi}$,定义 $\boldsymbol{u}=\boldsymbol{p}+\psi\boldsymbol{\tau}$ 使得:

$$\langle\boldsymbol{M}_0^n,\boldsymbol{u}\rangle=\langle\boldsymbol{M}_0^n,\boldsymbol{\tau}\rangle+\psi\langle\boldsymbol{M}_0^n,\boldsymbol{\tau}\rangle=p_1+\psi\tau_1=p_1+\frac{\alpha-\beta-p_1}{\phi}\times\tau_1=p_1+\frac{\alpha-\beta-p_1}{\tau_1}\times\tau_1=\alpha-\beta.$$

① 如果 $w\in(A\cup D\cup\Omega)\cap((A\cap S^*)\cup D')$,则$\langle\boldsymbol{M}_w^n,\boldsymbol{\tau}\rangle=0$,$q(w)=\langle\boldsymbol{M}_w^n,\boldsymbol{u}\rangle=\langle\boldsymbol{M}_w^n,\boldsymbol{p}\rangle+\psi\langle\boldsymbol{M}_w^n,\boldsymbol{\tau}\rangle=\langle\boldsymbol{M}_w^n,\boldsymbol{p}\rangle$.所以 C 可以简单选取 $r_w\in Z_p^*$ 并计算:

$$P_{w,1}=g^{q(w)}\times h_0^{r_w}=g^{\langle\boldsymbol{M}_0^n,\boldsymbol{p}\rangle}\times h_0^{r_w},$$

$$P_{w,2}=g^{r_w},$$

$$\{P_{w,i}\}_i^{N-1}=(h_1^{-w^i}\times h_{i+1})^{r_w}.$$

② 如果 $w\notin(A\cup D\cup\Omega)\cap((A\cap S^*)\cup D')$,则$\langle\boldsymbol{M}_w^n,\boldsymbol{\tau}\rangle\neq0$,$\boldsymbol{M}_0^n$ 不在 $\boldsymbol{M}_w^n$ 的范围,$\langle\boldsymbol{M}_0^n,\boldsymbol{u}\rangle=\alpha-\beta$,$C$ 可以通过两步来构建 sk_A.

第 1 步,对于属性 $w\notin(A\cap S^*)\cup D'$,$C$ 首先构造 $P_{w,1}^*$,$P_{w,2}^*$,$\{P_{w,i}^*\}_{i=1}^{N-1}=g^{\alpha-\beta}\times h_0^{\tilde{r}_w}$,$g^{\tilde{r}_w}$,$(h_1^{-w^i}\times h_{i+1})^{\tilde{r}_w}$并且定义一个 $N\times(N-1)$的矩阵 $\boldsymbol{Q}_w$:

$$\boldsymbol{Q}_w=\begin{pmatrix}-w & -w^2 & \cdots & -w^{N-1}\\ 1 & 0 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1\end{pmatrix},$$

然后随机选择 $\varepsilon_1\in Z_p^*$,$r\in Z_p$.定义 $\boldsymbol{\varepsilon}=\varepsilon_1\boldsymbol{M}_w^N$,使得 $\boldsymbol{\varepsilon}$ 满足 $\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{Q}_w=\boldsymbol{0}$ 但$\langle\boldsymbol{b},\boldsymbol{\varepsilon}\rangle=\varepsilon_1\varphi(w)\neq0$,另外定义 $\tilde{r}_w$ 为

$$\tilde{r}_w=\frac{r+\langle(a^N,a^{N-1},\cdots,a)^{\mathrm{T}},\boldsymbol{\varepsilon}\rangle}{\langle\boldsymbol{b},\boldsymbol{\varepsilon}\rangle}.$$

因为对于任意向量 $\boldsymbol{f}$,在乘积 $\tilde{r}_w\langle\boldsymbol{f},\boldsymbol{a}\rangle$中 a^{N+1} 的系数为$\dfrac{\langle\boldsymbol{f},\boldsymbol{\varepsilon}\rangle}{\langle\boldsymbol{b},\boldsymbol{\varepsilon}\rangle}$.$C$ 可以计算:

$$P_{w,1}^*=g^{\alpha-\beta}h_0^{\tilde{r}_w}=\frac{g^{a^{N+1}}\times g^{\delta_0}}{g^{\beta}}(g^{\theta_0}g^{-\langle\boldsymbol{a},\boldsymbol{b}\rangle})^{\tilde{r}_w},$$

$$P_{w,2}^*=g^{\tilde{r}_w},$$

$$P_{w,i}^*=g^{\boldsymbol{Q}_w^{\mathrm{T}}\boldsymbol{a}\tilde{r}_w}=(g^{-w\alpha_1}\times g^{\alpha_2})^{\tilde{r}_w}+\cdots+(g^{-w^{N-1}\alpha_1}\times g^{\alpha_N})^{r_w}=((g^{\alpha_1})^{-w^i}\times g^{\alpha_{i+1}})^{r_w}=((g^{a_1}\times g^{\theta_1})^{-w^i}\times(g^{a_{i+1}}\times g^{\theta_1}))^{\tilde{r}_w}=(h_1^{-w^i}\times h_{i+1})^{\tilde{r}_w}.$$

由于 a^{N+1} 在 $-\tilde{r}_w\langle\boldsymbol{a},\boldsymbol{b}\rangle$中的系数为 -1,所以 a^{N+1} 在 $P_{w,1}^*$ 中的系数为 0.因此 $P_{w,1}^*$,$P_{w,2}^*$,$\{P_{w,i}^*\}_i^{N-1}=1$ 可以被 C 高效地计算.

第 2 步,因为:

$$\boldsymbol{M}_w^n=\langle\boldsymbol{M}_w^n,\boldsymbol{p}\rangle+\psi\langle\boldsymbol{M}_w^n,\boldsymbol{\tau}\rangle=\sum_{j=1}^{n}w^{j-1}\left(p_j+\frac{\alpha-\beta-p-1}{\phi}\times\tau_j\right)=k_1(\alpha-\beta)+k_2,$$

其中 k_1,k_2 都是可计算的,分别为

$$k_1=\frac{1}{\phi}\sum_{j=1}^{n}w^{j-1}\times\tau_j,$$

$$k_2=\frac{1}{\phi}\sum_{j=1}^{n}w^{j-1}\times(\phi p_j-p_1\tau_j).$$

所以,C 可以选择 r'_w 并计算:

$$P_{w,1}=P_{w,1}^{*k_1}\times g^{k_2}\times h_0^{r'_w},$$

$$P_{w,2}=P_{w,2}^{*k_1}\times g^{r'_w},$$

$$\{P_{w,i}\}_i^{N-1}=P_{w,i}^{*k_i}\times(h_1^{-w_i}\times h_{i+1})^{r'_w}.$$

最后,C 将新元组(A,psk,sk)添加到列表 L 之后返回签名钥 sk 给 $\mathcal{F}$.

3) 签名询问(signing oracle).对每个消息 m,C 定义函数:

$$J(M)=\eta_0+\sum_{j=1}^{n_m}\eta_j\times u_j-2\nu q,$$

$$K(M)=\lambda_0+\sum_{j=1}^{n_m}\lambda_j\times u_j,$$

通过以上函数,对于一个消息 m,存在:

$$H(m)=u_0\prod_{j=1}^{n_m}u_j^{m_j}=g_N^{J(M)}\times g^{K(M)},$$

① 如果 $J(M)=0$,C 中止游戏.

② 如果 $J(M)\neq 0$,C 随机选择 r,r_w,s_0,s_1,s 以及 $s_2\in Z_p$,使得:

$$s+s_1=s_2-\frac{a}{J(M)},$$

签名可以模拟为:

$$\delta_0=g^{\delta_0}(h_0\prod_{i=1}^{N}h_i^{b_i})^r\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{s_0}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r_w}\times$$
$$(g_N^{J(M)}g^{K(M)})^{s_2}g_1^{-\frac{K(M)}{J(M)}}=$$
$$g^{\delta_0}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}g^{a^{N+1}}g^{-a^{N+1}}\times$$
$$(g_N^{J(M)}g^{K(M)})^{s_2}g_1^{-\frac{K(M)}{J(M)}}=$$
$$g^{\delta_0}g^{a^{N+1}}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}(g_N^{J(M)}g^{K(M)})^{s_2}\times$$
$$g^{-a^{N+1}}g_1^{-\frac{K(M)}{J(M)}}=$$
$$g^{a}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}(g_N^{J(M)}g^{K(M)})^{s_2}\times$$
$$(g^{-a^{N+1}\times\frac{J(M)}{J(M)}}g_1^{-\frac{K(M)}{J(M)}})=$$
$$g^{a}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}(g_N^{J(M)}g^{K(M)})^{s_2}\times$$
$$(g_N^{J(M)}g^{K(M)})^{-\frac{a}{J(M)}}=$$
$$g^{a}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}(g_N^{J(M)}g^{K(M)})^{s+s_1},$$
$$\delta_1=g^r\times g^{s_0}\times g^{r_w}=g^{r+s_0+r_w},$$
$$\delta_2=g^{s_2}\times g_1^{-\frac{1}{J(M)}}=g^{s+s_1},$$

因此 C 返回 m 的签名 $\sigma=\{m,\sigma_0,\sigma_1,\sigma_2\}$ 给 $\mathcal{F}$.

4) 签名验证询问(UVerify oracle).C 运行 Transform 算法即 C 随机选择转换钥 $t\in Z_p$ 并计算 $\tilde{\sigma}_0=\sigma_0^t,\tilde{\sigma}_1=\sigma_1^t$,然后返回转换签名 $\tilde{\sigma}$ 给 $\mathcal{F}$.收到 $\tilde{\sigma}$ 之后,F 运行 SVerify 算法去产生中介签名:

$$\frac{e(g,\tilde{\sigma}_0)}{e(h_0\prod_{i=1}^{N}h_i^{b_i},\tilde{\sigma}_1)}=\hat{\sigma}_1.$$

最后,$\mathcal{F}$ 返回中介签名 $\hat{\sigma}_1$ 给 C.C 计算 $\hat{\sigma}_2=e(H(m),\tilde{\sigma}_2)\times Z^t$ 并检查 $\hat{\sigma}_1=\hat{\sigma}_2$ 是否成立,输出 true 或者 false.

Forgery.如果 $\mathcal{F}$ 从未对 m^* 在访问结构 Γ_{k^*,S^*}^* 下进行签名询问,输出伪造签名 $\sigma^*=\{m^*,\Gamma_{k^*,S^*}^*,\sigma_0^*,\sigma_1^*,\sigma_2^*\}$.如果 $J(M)\neq 0$ 则 C 中止,否则 $J(M)=0$,存在:

$$\delta_0^*=g^{\delta_0}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{s_0}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r_w}\times$$
$$(g_N^{J(M^*)}g^{K(M^*)})^{s_2}g_1^{-\frac{K(M^*)}{J(M^*)}}=$$
$$g^{\delta_0}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}(g_N^{J(M^*)}g^{K(M^*)})^{s_2}g_1^{-\frac{K(M^*)}{J(M^*)}}=$$
$$g^{\delta_0}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}(g_N^{J(M^*)}g^{K(M^*)})^{s+s_1}\times$$
$$(g_N^{J(M^*)}g^{K(M^*)})^{\frac{a}{J(M^*)}}g_1^{-\frac{K(M^*)}{J(M^*)}}=$$
$$g^{\delta_0}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}(g_N^{J(M^*)}g^{K(M^*)})^{s+s_1}\times$$
$$(g_N^{J(M^*)})^{\frac{a}{J(M^*)}}(g^{K(M^*)})^{\frac{a}{J(M^*)}}g_1^{-\frac{K(M^*)}{J(M^*)}}=$$
$$g^{\delta_0}(g^{a^NJ(M^*)})^{\frac{a}{J(M^*)}}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}\times$$
$$(g_N^{J(M^*)}g^{K(M^*)})^{s+s_1}(g^{K(M^*)})^{\frac{a}{J(M^*)}}g^{-\frac{aK(M^*)}{J(M^*)}}=$$
$$g^{\delta_0}g^{a^{N+1}}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}(g_N^{J(M^*)}g^{K(M^*)})^{s+s_1}=$$
$$g^{\delta_0}g^{a^{N+1}}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}(g_N^{J(M^*)})^{s+s_1}\times$$
$$(g^{K(M^*)})^{s+s_1}=g^{\delta_0}g^{a^{N+1}}\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}\times$$
$$(g^{K(M^*)})^{s+s_1}=g^{\delta_0}g^{a^{N+1}}(g^{\theta_0+\langle\boldsymbol{\theta},\boldsymbol{b}\rangle})^{r+s_0+r_w}\times$$
$$(g^{K(M^*)})^{s+s_1}=g^{\delta_0}g^{a^{N+1}}(\delta_1^*)^{\theta_0+\langle\boldsymbol{\theta},\boldsymbol{b}\rangle}(\delta_2^*)^{K(M^*)},$$

所以 C 可以计算出

$$g^{a^{N+1}}=\frac{\delta_0^*}{g^{\delta_0}\times(\delta_1^*)^{\theta_0+\langle\boldsymbol{\theta},\boldsymbol{b}\rangle}\times(\delta_2^*)^{K(M^*)}}.$$

如果用 *abort* 表示 C 在模拟过程中终止,定义事件 $E_i:J(M_i)\neq 0,i\in[q]$,$E^*:J(M^*)=0$,则 C 成功的概率为

$$Pr[\overline{abort}]=Pr[\bigcap_{i=1}^{q}E_i\cap E^*]\geqslant\frac{1}{4q(n+1)}.$$

5.3 抗共谋攻击

由于我们所提出的 SA-VABS 方案在执行签名验证算法时包含消息 m 的部分是由验证者来执行,m 不参与中介签名的产生,所以签名者就无法勾结服务器基于消息 m^* 执行 SVerify 算法去产生中介签名,指导服务器欺骗验证者慌称中介签名是基于 m 产生的.因此,我们所提出的 SA-VABS 方案有效地抵抗了签名者和服务器的共谋攻击,保证了 SA 验证阶段的安全性.

5.4 匿名性

在 SA-VABS 方案中,消息 m 的签名为

$$\sigma_0=\sigma'_0\times P'_{w,1}=g^a\times\left(h_0\prod_{i=1}^{N}h_i^{b_i}\right)^{r+s_0+r_w}\times$$
$$H(m)^{s+s_1},$$

$$\sigma_1=\sigma_1'\times P_{w,2}=g^{r+s_0+r_w},$$
$$\sigma_2=\sigma_2'\times g^{s}=g^{s+s_1}.$$

从式中可以看出签名的产生只是通过选择随机数 r,r_w,s,s_0,s_1 并没有泄露用户的属性及访问结构的任何信息.所以我们提出的 SA-VABS 方案实现了匿名性.

6 性能评估

本节主要将提出的 SA-VABS 方案与其他的 3 种方案从功能和计算开销方面进行对比,最后对方案进行了性能分析.

6.1 功能对比

表 2 将提出的 SA-VABS 方案与方案 OABS-II[13],SA-ABSR[19],SA-ABS[20]进行了功能对比.在表 2 中,SAS Secure 和 SAV Secure 分别表示在 SA 签名产生阶段和 SA 验证阶段是否满足安全性.SA-Sign 和 SA-Verify 分别表示方案在签名和验证阶段是否运用 SA 技术.空白代表方案没有涉及.方案 OABS-II[13]只是在签名产生阶段运用 SA 技术,虽然方案 SA-ABSR[19]和 SA-ABS[20]在签名产生和验证阶段都用到了 SA 技术,但是它们不能对部分签名的有效性进行验证,因此不能抵抗服务器对部分签名的伪造,而且方案 SA-ABSR[19]不能抵抗签名者和服务器的共谋攻击.从表 2 可以看出 SA-VABS 方案在签名产生和验证阶段都运用了 SA 技术,而且可以抵抗签名者和服务器的共谋攻击,最重要的是可以对部分签名的有效性进行验证,从而抵抗了服务器对部分签名的伪造.因此我们的方案保证了 SA 签名产生和验证阶段的安全性,所以我们的方案有更好的安全性.

Table 2 The Functional Comparison of Four Schemes

表 2 4 种方案的功能对比

Scheme	SAS Secure	SAV Secure	SA-Sign	SA-Verify
OABS-II	×		√	×
SA-ABSR	×	×	√	√
SA-ABS	×	√	√	√
SA-VABS	√	√	√	√

Note: "√" means that the requirement is met; "×" means that the requirement is not met.

6.2 计算开销对比

表 3 将提出的 SA-VABS 方案与 OABS-II[13],SA-ABSR[19],SA-ABS[20]进行了计算开销的对比.Key.Gen,Sig.Gen,Verify 分别表示签名钥产生、签名产生以及用户验证的计算开销.n 和 d 分别表示默认的属性集合和用户的属性集合.E,P,H 分别表示指数运算、双线性对运算以及 Hash 运算的时间消耗.对于计算开销,从表 3 可以看出我们的方案在签名钥产生方面优于其他方案;在签名验证方面,我们的方案与最新的方案 SA-ABS[20]持平.

Table 3 The Computational Overheads Comparison

表 3 计算开销对比

Scheme	Key.Gen	Sig.Gen	Verify
OABS-II	$3(n+d+1)E$	$(2n+2)E$	$(2n+2)E+3P+H$
SA-ABSR	$(6n+3)E$	$(2n+3)E$	$4E$
SA-ABS	$(d^2+3d+2)E$	$5E$	$4E+P$
SA-VABS	$(4n+4)E$	$(2n+3)E$	$4E+P$

Note: E means the time consumption of an exponentiation;
P means the time consumption of a pairing operation;
H means the time consumption of a Hash computation.

6.3 性能分析

在实验中,我们使用 JPBC(Java pairing based cryptography)库[22]在装有 Intel Core i5-7440HQ 2.8 GHz 处理器和 8 GB 内存的 WINDOWS 系统上进行仿真实验.使用 type A 类型的双线性对在域 F_p 上构建椭圆曲线 $y^2=x^3+x$.

在图 2 中,将我们提出的 SA-VABS 方案和方案 OABS-II[13],SA-ABSR[19],SA-ABS[20]对签名钥产生所消耗的时间进行了对比.默认的属性集合大小设置为 $n=10$,横坐标表示用户的属性数量,纵坐标表示签名钥产生的时间.可以看出 SA-VABS 方案与方案 SA-ABSR[19]的签名钥产生时间是恒定的,

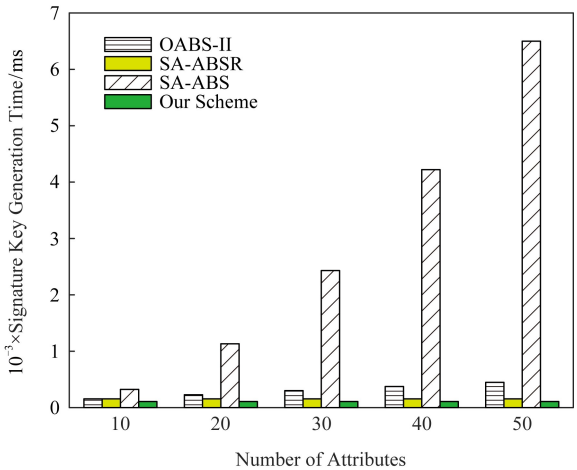

Fig. 2 Time comparison of signature key generation

图 2 签名钥产生的时间对比

不会随着用户属性的数量发生变化,而方案 OABS-II[13]和 SA-ABS[20]签名钥产生的时间随着属性的数量呈线性增长.因此,SA-VABS 方案的签名钥产生效率有着明显的优势.图 3 表示签名产生所消耗时间的对比,这 4 个方案签名产生的时间都保持不变,由于我们的 SA-VABS 方案在部分签名生成时进行了有效性验证,因此消耗的时间略高于方案 SA-ABS[20].同样,图 4 表示签名验证所消耗时间的对比,不可否认的是 SA-VABS 的签名验证时间与最新方案 SA-ABS[20]持平,略高于方案 SA-ABSR[19],但是我们的 SA-VABS 方案有更好的安全性.总之,在我们的方案中,签名钥生成、签名生成以及签名验证的时间消耗都是是恒定的,不会随着用户属性的数量而增加.因此,我们的方案适用于资源受限的 IIoT 场景中.

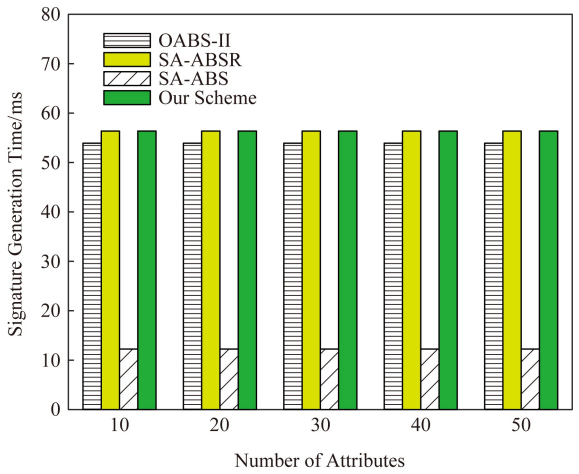

Fig. 3 Time comparison of signature generation

图 3 签名产生的时间对比

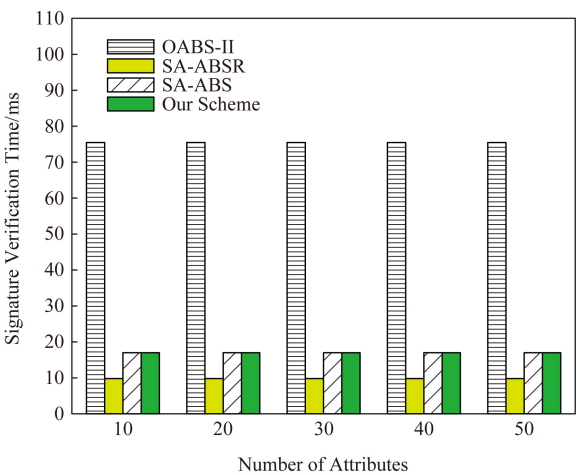

Fig. 4 Time comparison of signature verification

图 4 签名验证的时间对比

7 总 结

对于物联网中资源受限的设备,普通 ABS 方案存在的挑战是计算开销过高.方案 SA-ABSR[19]和 SA-ABS[20]通过将签名和验证阶段的主要计算委托给服务器来克服这种挑战.但是这些方案都不能对服务器产生的部分签名进行有效性验证,可能造成服务器对部分签名的伪造.基于此,我们提出一种 SA-VABS 方案,该方案不仅可以减小签名和验证阶段的计算开销,而且可以抵抗签名者和服务器的共谋攻击,最重要的是可以验证部分签名的有效性,防止服务器对部分签名的伪造.最后,通过具体的安全性分析表明所提出的 SA-VABS 方案是安全的,并且通过仿真实验和对比分析表明该方案是高效的.

参 考 文 献

[1] Gubbi J, Buyya R, Marusic S, et al. Internet of things (IoT): A vision, architectural elements, and future directions[J]. Future Generation Computer Systems, 2013, 29(7): 1645-1660

[2] Zhang Yinghui, Deng R H, Zheng Dong, et al. Efficient and robust certificateless signature for data crowdsensing in cloud-assisted industrial IoT [J]. IEEE Transactions on Industrial Informatics, 2019, 15(9): 5099-5108

[3] Que Mengfei, Zhang Junwei, Yang Chao, et al. Position based digital signature scheme in IoTs [J]. Journal of Computer Research and Development, 2018, 55(7): 1421-1431 (in Chinese)
(阙梦菲, 张俊伟, 杨超, 等. 物联网中基于位置的数字签名方案[J]. 计算机研究与发展, 2018, 55(7): 1421-1431)

[4] Zhang Jiansong, Wang Zeyu, Yang Zhice, et al. Proximity based IoT device authentication [C] //Proc of IEEE Conf on Computer Communications (INFOCOM'2017). Piscataway, NJ: IEEE, 2017: 1-9

[5] Wu Wei, Mu Yi, Susilo W, et al. Provably secure server-aided verification signatures [J]. Computers & Mathematics with Applications, 2011, 61(7): 1705-1723

[6] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption [C] //Proc of IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2007: 321-334

[7] Maji H K, Prabhakaran M, Rosulek M. Attribute-based signatures [C] //Proc of the Cryptographer's Track at RSA Conf. Berlin: Springer, 2011: 376-392

[8] Li Jin, Au M H, Susilo W, et al. Attribute-based signature and its applications [C] //Proc of the 5th ACM Symp on Information, Computer and Communications Security. New York: ACM, 2010: 60-69

[9] Ge A J, Ma C G, Zhang Z. Attribute-based signature scheme with constant size signature in the standard model [J]. IET Information Security, 2012, 6(2): 47-54

[10] Su Jinshu, Cao Dan, Zhao Baokang, et al. ePASS: An expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the Internet of things [J]. Future Generation Computer Systems, 2014, 33(4): 11-18

[11] Zhang Qi, Cheng Lu, Boutaba R. Cloud computing: State-of-the-art and research challenges [J]. Journal of Internet Services and Applications, 2010, 1(1): 7-18

[12] Hohenberger S, Lysyanskaya A. How to securely outsource cryptographic computations [C] //Proc of Theory of Cryptography Conf. Berlin: Springer, 2005: 264-282

[13] Chen Xiaofeng, Li Jin, Huang Xinyi, et al. Secure outsourced attribute-based signatures [J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(12): 3285-3294

[14] Ren Yanli, Jiang Tiejin. Verifiable outsourced attribute-based signature scheme [J]. Multimedia Tools and Applications, 2018, 77(14): 18105-18115

[15] Mo Ruo, Ma Jianfeng, Liu Ximeng, et al. EOABS: Expressive outsourced attribute-based signature [J]. Peer-to-Peer Networking and Applications, 2018, 11(5): 979-988

[16] Sun Jiameng, Su Ye, Qin Jing, et al. Outsourced decentralized multi-authority attribute based signature and its application in IoT [J]. IEEE Transactions on Cloud Computing, 2019. DOI:10.1109/TCC.2019.2902380

[17] Matsumoto T, Kato K, Imai H. Speeding up secret computations with insecure auxiliary devices [C] //Proc of Conf on the Theory and Application of Cryptography. Berlin: Springer, 1988: 497-506

[18] Wang Zhiwei, Xie Ruirui, Wang Shaohui. Attribute-based server-aided verification signature [J]. Applied Mathematics & Information Sciences, 2014, 8(6): 3183-3190

[19] Cui Hui, Deng R H, Liu J K, et al. Server-aided attribute-based signature with revocation for resource-constrained industrial-internet-of-things devices [J]. IEEE Transactions on Industrial Informatics, 2018, 14(8): 3724-3732

[20] Xiong Hu, Bao Yangyang, Nie Xuyun, et al. Server-aided attribute-based signature supporting expressive access structures for industrial Internet of things [J]. IEEE Transactions on Industrial Informatics, 2020, 16(2): 1013-1023

[21] Herranz J, Laguillaumie F, Libert B, et al. Short attribute-based signatures for threshold predicates [C] //Proc of Cryptographers' Track at the RSA Conf. Berlin: Springer, 2012: 51-67

[22] De Caro A, Iovino V. jPBC: Java pairing based cryptography [C] //Proc of IEEE Symp on Computers and Communications. Piscataway, NJ: IEEE, 2011: 850-855

Zhang Yinghui, born in 1985. PhD, professor. His main research interests include public key cryptography, cloud security and wireless network security.

He Jiangyong, born in 1994. Master candidate. His main research interests include cloud security and IoT security.

Guo Rui, born in 1984. PhD, associate professor. His main research interests include public key cryptography, cloud security and IoT security.

Zheng Dong, born in 1964. PhD, professor. His main research interests include code-based cryptography and IoT security.