计算机研究与发展 ISSN 1000-1239/CN 11-1777/TP
Journal of Computer Research and Development 50(10): 2070-2081, 2013

一种高效的具有灵活属性证书状态校验机制的 PBA 方案

周福才[1] 岳笑含[2] 白洪波[3] 徐 剑[1]
[1](东北大学软件学院 沈阳 110819)
[2](沈阳工业大学信息科学与工程学院 沈阳 110870)
[3](东北大学信息科学与工程学院 沈阳 110819)
(yxh21@yeah.net)

An Efficient Property-Based Attestation Scheme with Flexible Checking Mechanisms of Property Certificate Status

Zhou Fucai[1], Yue Xiaohan[2], Bai Hongbo[3], and Xu Jian[1]
[1](*College of Software, Northeastern University, Shenyang* 110819)
[2](*College of Information Science and Engineering, Shenyang University of Technology, Shenyang* 110870)
[3](*College of Information Science and Engineering, Northeastern University, Shenyang* 110819)

Abstract Remote binary attestation scheme of the trusted computing platform guarantees the integrity and hence the trustworthiness of the platform can be demonstrated to remote parties. However, as pointed out recently, the binary attestation has some shortcomings, particularly in applications. The major problem of the binary attestation is that it reveals the information about the configuration of a platform (hardware and software) or applications, which may lead to privacy issues, such as discrimination services, anonymity violations, etc. In order to solve the problems of platform configuration information leakage caused by the traditional binary attestation in the trusted computing environment, we propose a new privacy-preserving property-based attestation (PBA) scheme, which has flexible checking mechanisms of property certificate status, low computational cost and provable security in the random oracle model. By making use of the ideas of the verifier-local revocation and tracing signatures in the group signature, we present new certificate checking mechanisms, which include offline checking mechanism and online checking mechanism. Moreover, we design the model of the scheme, define the security model of the scheme, give concrete construction of the scheme in detail, and formally prove the security of this scheme in the random oracle model. We prove that this scheme satisfies the correctness, attestation unforgeability and configuration privacy. Finally, compared with other existing PBA schemes, the proposed PBA is more practical and efficient in both the computational cost and attestation length.

Key words trusted computing; property-based attestation (PBA); verifier-local revocation; attestation unforgeability; configuration privacy

摘 要 可信计算平台的远程二进制证明方案确保了该平台的完整性,通过这种方法平台可以向远程方证明其可信性.然而这种二进制证明方案却存在很多缺陷,其中一个主要问题就是泄露了关于平台的(软、硬件)配置信息,这导致很多隐私问题的出现,例如差别化服务及匿名性破坏等问题.因此针对在可

收稿日期:2011-10-12;**修回日期**:2012-04-26
基金项目:国家科技重大专项基金项目(2013ZX03002006);国家"八六三"高技术研究发展计划基金项目(2009AA01Z122);辽宁省百千万人才工程项目(2011921071);沈阳市科学技术计划项目(F12-277-1-41)
通信作者:岳笑含(yxh21@yeah.net)

信计算环境下传统二进制证明中所带来的平台配置信息泄露等问题,提出了一种新型的基于属性证明方案(property-based attestation, PBA).该方案具有属性证书状态校验机制灵活、方案整体计算代价小及随机预言模型下可证安全等特点.利用本地验证者撤销的技术,设计了方案的模型,定义了方案的安全性,给出了方案的具体构建,并在随机预言模型下对该方案进行了安全性证明,证明其满足正确性、证明不可伪造性及配置隐私性等安全性质.最后将提出的 PBA 方案与现有 PBA 方案在计算代价和证明值长度方面分别进行了比较,比较表明该方案同时具有实用、高效的特点.

关键词 可信计算;基于属性的证明;本地验证者撤销;证明不可伪造性;配置隐私性

中图法分类号 TP309.07

在当今 Internet 开放式的网络环境中,恶意代码的滋生蔓延造成了用户和服务提供商的巨大损失,因此需要建立一种分布式的可信计算环境来确保各方的行为是可预期的.为达到这个目的,计算机行业成立了可信计算组织 TCG[1],并开发了可信平台模块 TPM 来确保平台的真实性、机密性及认证性.在平台远程证明方面 TCG 为平台认证提供了相应的解决方案,称作二进制证明[2],即作为示证者的 TPM 将平台上软硬件的度量结果(通常是二进制的 Hash 值,亦称完整性报告),发送给验证者,验证者校验完整性报告并评估平台当前的状态及安全性.但是这种二进制证明的方法存在以下两方面的缺点:

1) 验证者校验完整性报告负担大.由于不同平台具有不同的软硬件配置,这需要验证方知道所有平台的所有"可信"的配置信息,以对示证者所提交的完整性报告进行验证.对于验证者来说,维护一个这样庞大的且需实时更新的平台配置信息数据库是非常困难的,这同样会导致验证者在验证过程中的执行性变差.

2) 会破坏示证者的平台隐私性.由于示证者报告了自身系统中所拥有软硬件的标识,敌手通过该标识可以得出该平台拥有的特性,这会导致以下几种情况的发生:①差别化的服务,远程服务提供方可能会因为对方所采用的商业模型而拒绝提供服务,比如某些聊天软件针对某种平台配置信息拒绝提供服务;②针对配置的攻击,若敌手通过收集平台配置信息知道某些硬件或软件的配置存在于大量平台上,那么敌手可以根据这些配置实施有针对性的攻击;③破坏了配置隐私性,敌手可以根据平台配置信息来唯一确定一个平台.

1 相关工作

为克服上述问题,国内外学者提出了不同的解决方法.文献[3]提出了一种语义远程认证(semantic remote attestation)方法,该方法利用基于 Java 可信虚拟机来远程证明程序代码的属性;文献[4]提出了一种基于 Linux 的软件结构体系的远程证明方案,在该体系中需要用认证中心 CA 来保证长生命期数据(例如 Linux 内核)某种配置的可信赖性;另外还有基于软件的证明[5]和基于属性的证明[6]等方法.在众多方法中,基于属性的远程证明具有明显的发展优势,它克服了 TCG 远程证明的证明复杂性、配置隐私泄漏和滥用证明结果等缺陷,并且易于实现远程证明扩展.

Poritz 等人[6]及 Sadeghi 等人[7]于 2004 年提出了基于属性证明(property-based attestation, PBA)的概念及模型.PBA 的基本思想是不同的平台会有不同的配置标准(configuration specification, cs),但是它们可能拥有相同的属性标准(property specification, ps)满足相同的验证需求.相对于二进制证明,基于属性的证明利用属性证书(property certificate)通过将二进制证明中对平台配置的证明转换为对平台属性的证明,一方面示证者可以根据验证者需要验证的目标策略属性给出满足属性的证明,提高了验证者的验证效率;另一方面确保了用户平台配置信息的隐私性.

在基于属性的证明研究方面,文献[8]通过对启动引导程序(boot loader)改进,提出了基于属性的引导程序(property-based boot loader, PBBL)体系结构;文献[9]在其基础之上作了进一步的扩展,给出了基于 PBBL 的 PBA 协议;文献[10]针对移动环境的特点,基于移动设备的硬件可信执行环境(trusted execution environments, TrEE)提出了利用可信证明服务(trusted attestation service),该服务是能将应用程序标志符与对应属性之间进行映射的 PBA 方案.但上述 3 种方案均需要对原有的 TCG 信任模型[1]进行修改且在文献[10]方案中对于属性

撤销的问题没有考虑.

在不改变原有 TCG 信任模型的可证明安全 PBA 方案中,文献[11]第 1 次提出了可证明安全的基于属性证明方案,但是该方案的属性证书撤销算法需要示证者与验证者之间的在线交互来完成且计算代价较大;文献[12]利用环签名等技术提出一种无需可信第三方的 PBA 方案,但是该方案存在签名长度和计算代价与备选 cs 配置信息个数呈线性关系的问题;文献[13]提出了一种组件级细粒度的 PBA 方案,但在该方案中各参与者的计算代价较大,不能满足实际需求;文献[14]提出了较高效的基于双线性配对的 PBA 方案,虽然该方案的撤销检查 Check 协议计算代价为常数级,但该协议需要验证者与可信第三方(trusted third party, TTP)进行在线交互来实现,即该方案每一次运行都需要 TTP 的参与,若 TTP 处于离线状态,那么验证者无法对示证者的属性证明进行校验,另外方案中存在 TPM 计算代价较高的问题.

基于以上的问题,本文利用本地验证者可撤销性技术[15-18],结合基于属性证明的思想[7]提出了高效的具有灵活撤销机制的 PBA 方案,该方案较以上 PBA 方案[11-14]具有以下的优点:

1) 为验证者灵活地提供了 2 种属性证书状态查询的方法:一是离线证书状态校验.在本地通过下载属性证书撤销列表 PCRL 对示证者的属性证书状态自行进行验证,无需 TTP 的参与并避免了可信第三方离线的情况;二是在线证书状态查询.验证者可以通过属性证书状态查询服务器对属性证书的状态进行查询.这 2 种方法为用户校验查询提供了可靠的保障.

2) 由于 TPM 自身的计算能力及存储能力有限,因此针对 TPM 资源有限的性质,本文所设计的 PBA 方案在计算代价方面及证明值长度方面要优于现有的 PBA 方案[11-14].

2 相关知识

2.1 双线性配对

令 G_1,G_2 是 2 个 q 阶循环群,其中 q 为素数,g_1,g_2 分别是 G_1,G_2 的生成元.定义 2 个群上的双线性映射为 $e:G_1\times G_2\rightarrow G_T$,且满足下面的性质:

1) 双线性. $e(g_1^a,g_2^b)=e(g_1,g_2)^{ab}$,对所有的 a, $b\in\mathbb{Z}_q^*$ 均成立.

2) 非退化性. $e(g_1,g_2)\neq 1_{G_T}$,其中 1_{G_T} 是 G_T 的单位元.

3) 可计算性.存在有效算法来计算 $t=e(g_1,g_2)$.

2.2 密码学假设

定义 1. 判定 Diffie-Hellman(decisional Diffie-Hellman, DDH)假设.该 DDH 难题在群 $G_1=\langle g_1\rangle$ 中的定义如下,以四元组 $(g_1,g_1^a,g_1^b,g_1^c)\in G_1^4$ 作为输入,不存在一个概率多项式时间算法能够以不可忽略的概率 ε 判断 c 是否等于 $a\times b$.

定义 2. 外 Diffie-Hellman(external Diffie-Hellman, XDH)假设[19].对于一个非对称双线性配对 $e:G_1\times G_2\rightarrow G_T$,若 DDH 难题在群 G_1 中是困难的,那么该 XDH 假设成立.

定义 3. l-强 Diffie-Hellman(l-strong Diffie-Hellman, l-SDH)假设[20].该 l-SDH 假设在群($G_1=\langle g_1\rangle$, $G_2=\langle g_2\rangle$)中定义如下:以$(l+2)$个元组 $(g_1,g_2,g_2^x,g_2^{x^2},\cdots,g_2^{x^l})\in G_1\times G_2^{l+1}$ 作为输入,不存在一个概率多项式时间算法能够以不可忽略的概率 ε 输出一个二元组(A,c),使得 $A^{(x+c)}=g_1$ 成立,其中 $c\in\mathbb{Z}_p^*$.

2.3 本地验证者撤销

本地验证者撤销(verifier-local revocation, VLR)的概念由 Boneh 等人[15]提出并作为一种群成员撤销机制应用于群签名[21-22]中. VLR 的基本思想是群管理者维护一个撤销列表(revocation list, RL)用于存储群成员撤销令牌 $RT[i]$,当验证者接收到来自于该群的群签名 σ 及消息 m 时,不同于传统群签名,一方面需要利用群公钥 gpk 通过校验算法 Verify 判定该签名的有效性,另一方面通过下载的 RL 来校验是否该群签名由已被撤销的群成员生成.验证过程的形式化表示如下:

$$Verify(gpk,\sigma,m,RL)\rightarrow\begin{cases}invalid(RT[i]\in RL)\\ valid(RT[i]\notin RL)\end{cases}.$$

并且当有非法用户存在的情况时,验证者也可以通过撤销令牌关联该用户之前产生的所有群签名并能够追踪到该名用户;然而在不需要对非法用户追踪的情况下时,这种撤销方法存在隐私性的问题,即向后关联的问题[16],即成员在被撤销后,验证者可以通过对应的撤销令牌来关联该名成员之前生成的所有群签名,破坏了在这种情况下的用户隐私性.因此文献[16]提出了向后不可关联性(back unlinkability, BU)的概念解决了上述的问题.综合以上 2 种情况,

本文在构建方案时首先根据第 1 种情况构建了不具备 BU 性质的 PBA 方案，然后在第 6 节中根据第 2 种情况作了进一步的讨论，使方案具备 BU 性质.

3 本文的 PBA 模型

3.1 PBA 模型描述

在本文提出的 PBA 方案中，方案的参与者有 4 个实体：属性证书发布中心（property certificate Issuer，*PCI*）、属性证书验证中心（property certificate verification center，*PCVC*）、示证者（prover，*P*）及验证者（Verifier，*V*）. 其中：

1）*PCI* 作为属性证书的发布者，提供对配置信息 *cs* 的属性评估及颁发关于该配置信息属性证书的服务；另外要维护本地证书数据库 $dbase_{\mathrm{Cert}}$ 以及属性证书撤销列表 *PCRL* 的更新；

2）*PCVC* 作为属性证书验证者，以在线的方式提供对属性证书有效性的校验；

3）示证者 *P* 由可信计算芯片 TPM *M* 及主机 *H* 构成，提供远程属性证明服务. 值得说明的是，对于可信环境下的安全协议（例如直接匿名证明、PBA 等）需要主机 *H* 具有可信组件（trusted component），通常这个可信组件属于可信计算基（trusted computing base，TCB），否则主机 *H* 会泄露平台的配置信息或者相关安全协议私密信息；

4）验证者 *V* 通常为服务提供者（server provider），校验由示证者 *P* 提供的属性证明，判定是否该属性证明能满足其所需属性，并且能够判断该证明是由一个属性证书未被撤销的示证者 *P* 产生.

PBA 模型主要由 *Setup*，*Issue*，*Revoke*，*Attest*，*Verify*，*Check* 六种算法组成：

1）$Setup(1^k)$. 以安全参数 1^k 作为输入，*PCI* 使用随机化的算法生成（ipk，isk，ivk），其中 ipk 是 *PCI* 的公钥，isk，ivk 分别是 *PCI* 私钥及 *PCVC* 的验证密钥.

2）$Issue(ipk, isk, cs_i)$. *PCI* 对示证者 *P* 提交的配置信息 cs_i 进行评估，得到对应的属性 ps_i，并用 ipk，isk 生成关于该配置信息的属性证书 $PCert_i$，将（$PCert_i$，cs_i，ps_i）颁发给示证者 *P*，并将（$PCert_i$，cs_i，ps_i，$RevFlag=1$）保存到本地证书数据库 $dbase_{\mathrm{Cert}}$ 中，其中 *RevFlag* 代表属性证书撤销标识符，为 1 是有效、为 0 是无效.

3）$Revoke(PCert_i)$. 当一个属性证书 *PCert* 过期或者配置信息不满足对应属性时需要对该证书进行撤销，首先 *PCI* 将证书数据库 $dbase_{\mathrm{Cert}}$ 对应该证书的 *RevFlag* 值设置为 0（无效），然后，将关于该证书的撤销令牌 $RT[i]$ 更新到 *PCRL* 列表中.

4）$Attest(ipk, cs_i, ps_i, PCert_i, N_t, N_v)$. 示证者 *P* 对于验证者 *V* 的挑战值 N_v 及 TPM *M* 的随机值 N_t，结合（cs_i，ps_i，$PCert_i$）利用随机算法生成属性证明 σ_{PBA}.

5）$Verify(ipk, \sigma_{\mathrm{PBA}})$. 验证者 *V* 利用 *PCI* 的公钥 ipk 对属性证明 σ_{PBA} 进行校验，如果校验通过则返回 1（接受），否则返回 0（拒绝）.

6）$Check(ipk, \sigma_{\mathrm{PBA}}, PCRL)$. 该算法由以下 2 种验证者 *V* 可选择的算法组成：

① 离线校验算法（Check. Offline）. 结合 *PCRL* 列表中的撤销令牌，验证者 *V* 校验属性证明中的证书是否在 *PCRL* 列表中，若在则返回 1（未撤销），否则返回 0（已撤销）；

② 在线校验算法（Check. Online）. 验证者 *V* 提交相关属性证明信息，*PCVC* 利用校验密钥 ivk 得到相应的属性证书，查询并返回数据库 $dbase_{\mathrm{Cert}}$ 中证书的状态：1 为未撤销，0 为已撤销.

上述的实体及算法之间的关系如图 1 所示：

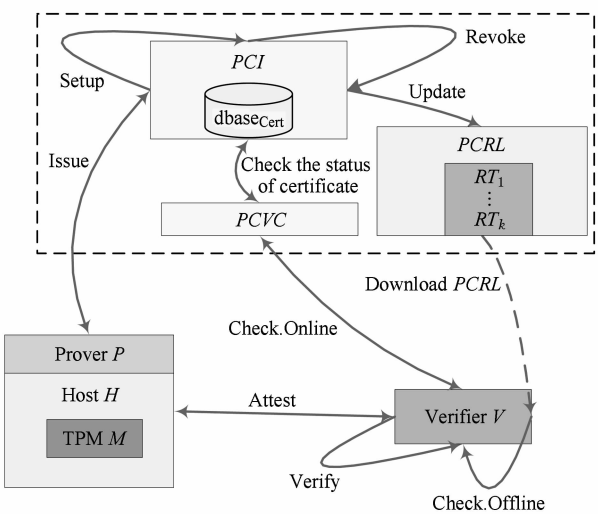

Fig. 1 The model of proposed PBA in this paper.

图 1 本文提出的 PBA 模型

3.2 PBA 安全模型

为满足本文提出的 PBA 方案的安全性，需要具备以下几个性质：

定义 4. 正确性. 对于诚实的验证者，由有效未被撤销示证者产生的任何证明都可以验证通过. 因此由 3.1 节提出的算法必须满足以下一致性需求，即

$$1 \leftarrow \mathrm{Verify}(ipk, N_t, N_v, \sigma_{\mathrm{PBA}}) \wedge \mathrm{Check}(ipk, \sigma_{\mathrm{PBA}}, PCRL).$$

定义 5. 配置隐私性.如果在下面的游戏中不存在概率多项式时间的敌手 A 拥有不可忽略的优势赢得该游戏,那么 PBA 方案具有配置隐私性.

1) 初始化.挑战者 C 运行 $Setup(1^k)$ 生成私钥 isk、公钥 ipk,将公钥 ipk 发送给敌手 A.

2) 质询.敌手 A 自适应地对挑战者 C 进行以下质询,质询方式如下:

① Issue oracle. A 向 C 发出第 i 次 Issue 请求,C 选择 $cs_i \in CS=\{cs_1,cs_2,\cdots,cs_n\}$,以及属性 $ps_i \in PS=\{ps_1,ps_2,\cdots,ps_n\}$,并运行 Issue 算法得到 cs_i 关于属性 ps_i 的证书 $PCert_i$,将 $PCert_i$ 及 ps_i 发送给 A.

② Attest oracle. A 提交 $N_t \| N_v$ 向 C 发出第 i 次 Attest 请求,C 运行 Attest 算法生成证明 $\sigma_{\mathrm{PBA}}^{(i)}$,并将其作为对 A 的应答.

③ Corrupt oracle. A 提交它所选择的索引 i,C 将其对应的 cs_i 作为应答.

④ Revocation oracle. A 提交它所选择的索引 i,C 将其对应的 RT_{cs_i} 作为应答.

3) 挑战.在这个阶段,A 提交它所选择的 2 个索引 $\{i_0,i_1\}$ 以及消息 $N_t \| N_v$,索引 $\{i_0,i_1\}$ 是 A 未向 Corrupt 和 Revocation 质询过的.挑战者 C 随机选择 $b \in_R \{0,1\}$,并生成关于索引 i_b 对应的证明 $\sigma_{\mathrm{PBA}}^{(b)}$ 作为挑战.

4) 输出. A 需要根据属性证明 $\sigma_{\mathrm{PBA}}^{(b)}$ 输出对应索引值 i^*,如果 $i^*=i_b$,那么表示其挑战成功,能够判断该证明是由 i^* 对应的配置信息产生,否则挑战失败,表明敌手无法通过证明值来判定平台配置信息.

定义 6. 证明不可伪造性.如果在下面的游戏中不存在概率多项式时间的敌手 A 拥有不可忽略的优势赢得该游戏,那么 PBA 方案具有证明不可伪造性.

1) 初始化.挑战者 C 运行 $Setup(1^k)$ 生成校验密钥 ivk 以及公钥 ipk,将公钥 ipk、校验密钥 ivk 发送给敌手 A.

2) 质询.敌手 A 自适应地对挑战者 C 进行以下质询,质询方式如下:

① Issue oracle. A 仅提交一个它所选择的索引 i 给 C,C 运行 Issue 协议为该索引创建对应的 $(PCert_i,ps_i)$,将其作为返回值.

② Attest oracle. A 提交 $N_t \| N_v$ 向 C 发出第 i 次 Attest 请求,C 运行 Attest 算法生成证明 $\sigma_{\mathrm{PBA}}^{(i)}$,并将其作为对 A 的应答.

③ Corrupt oracle. A 提交它所选择的索引 i,C 将其对应的 cs_i 作为应答,并将 cs_i 以及对应的证书信息以撤销令牌的形式保存到质询列表 L_q 中.

3) 输出.在经过了多次质询后,这个阶段 A 输出 $(i^*,\sigma_{\mathrm{PBA}}^*,N_t^* \| N_v^*)$,其中 $\sigma_{\mathrm{PBA}}^* \notin \sigma_{\mathrm{PBA}}^{(i)}$,$i \in \{1,\cdots,q\}$,其中 q 为 A 质询次数.那么 A 赢得该游戏条件是:

① $\mathrm{Verify}(ipk,\sigma_{\mathrm{PBA}}^*,N_t^* \| N_v^*) \wedge \mathrm{Check}(\sigma_{\mathrm{PBA}}^*,PCRL) \rightarrow 1$;

② σ_{PBA}^* 由 Trace 算法打开后对应的结果不在 C 的质询列表 L_q 里.

4 PBA 方案的设计

4.1 初始化(setup)算法

1) 发布者 PCI 选择一个非对称双线性群对 (G_1,G_2),2 个群的阶是长度为 l_q 的素数 q,g_1 和 g_2 分别是群 G_1 和 G_2 的生成元,并有配对函数 $e:G_1 \times G_2 \rightarrow G_T$;

2) 发布者 PCI 选择一个 Hash 函数 $H(m) \rightarrow \{0,1\}^{l_H}$ 以及如下参数:$h_1 \in_R G_1$,$\gamma,\eta \in_R \mathbb{Z}_q^*$,并计算 $\omega:=g_2^{\gamma}$,$u:=g_1^{\eta}$;

3) 发布者 PCI 的公钥、群私钥以及证书校验者 $PCVC$ 的校验密钥分别如下:

$$\{ipk,isk,ivk\}:=\{(G_1,G_2,G_T,e,q,g_1,h_1,g_2,u,\omega,H),\gamma,\eta\}.$$

4.2 颁发(issue)算法

1) 示证者 P 收集相关平台配置信息 $cs_i \in \{0,1\}^{l_{cs}}$,为申请关于该配置的属性证书 $PCert_i$,P 将该配置信息 cs_i 经由安全通道提交给发布者 PCI(关于如何产生平台配置信息 cs_i 不在本文讨论范围之内);

2) 发布者 PCI 依据该配置信息 cs_i,评估其对应的属性 $ps_i \in \{0,1\}^{l_{ps}}$,并作如下计算生成属性证书:

$$PCert_i:=(g_1^{ps_i}h_1)^{1/(\gamma+cs_i)};$$

3) 发布者 PCI 将属性证书和 $\{cs_i,ps_i,RevFlag=1\}$ 储存到本地属性证书数据库 $dbase_{\mathrm{Cert}}$ 中;

4) 发布者 PCI 将该证书及属性 ps_i 发送给示证者 P;

5) 示证者 P 接收到该证书后,对证书进行如下计算及校验:

$$e(PCert_i,\omega \times g_2^{cs_i}) \stackrel{?}{=} e(g_1^{ps_i} \times h_1,g_2).$$

若通过校验则保存该证书作为关于配置信息 cs_i 的属性证书.

4.3 撤销(revoke)算法

如果有证书 $PCert_i$ 被撤销，发布者 PCI 作如下操作：

1）发布者 PCI 将在证书数据库中查找关于该证书的 $RevFlag$ 项，并将该值置 0；

2）发布者 PCI 发布以下参数作为撤销令牌：

$$RT_i := \{RT_i^{(1)}, RT_i^{(2)}\} = \{PCert_i, g_2^{cs_i}\},$$

并将该撤销令牌更新到 $PCRL$ 列表中.

4.4 证明(attest)算法

1）验证者 V 向示证者 P 发起属性证明挑战，挑战值为 $N_v \in_R \{0,1\}^{l_H}$ 用于确保新鲜性，示证者 P 接收到该值后发送给 TPM M；

2）TPM M 随机选取一个值 $\zeta \in_R \mathbb{Z}_q^*$，并计算关于 cs_i 的承诺值 $C_{cs_i} := g_1^{cs_i \times \xi}$；

3）TPM M 选取一个随机数 $N_t \in_R \{0,1\}^{l_\phi}$，并用 AIK(身份认证密钥)私钥(或 DAA 签名算法[23])对承诺值 C_{cs_i} 以及 N_v, N_t 作签名得到 TPM 的签名值：

$$\delta_m := Sig_m(C_{cs_i}, N_v \| N_t),$$

TPM M 将 $(\delta_m, C_{cs_i}, \zeta, N_t)$ 这 4 个值发送给主机 H；

4）主机 H 计算 $T_1 := PCert_i \times g_1^{\zeta}$，令 $T_2 := C_{cs_i}$ 以及 $T_3 := u^{\zeta}$，主机 H 执行接下来的步骤以完成如下的知识签名证明：

$$SPK\{(cs_i, \zeta, PCert_i):$$
$$T_1 = PCert \times g_1^{\zeta}, T_2 = g_1^{cs_i \times \zeta},$$
$$T_3 = u^{\zeta}, e(PCert_i, \omega g_2^{cs_i}) =$$
$$e(g_1^{ps_i} h_1, g_2)\}(N_v, N_t);$$

5）主机 H 计算 $\alpha := cs_i \times \zeta$，并随机选取 3 个盲化因子 $r_\zeta, r_{cs_i}, r_\alpha \in_R \mathbb{Z}_q^*$，并作如下计算：

$$R_1 := g_1^{r_\alpha},$$
$$R_2 := e(g_1, g_2)^{ps_i} e(T_1, g_2)^{-r_{cs_i}} e(g_1, g_2)^{r_\alpha} e(g_1, \omega)^{r_\zeta},$$
$$R_3 := u^{r_\zeta}.$$

6）主机 H 计算：

$$c_H := H(ipk \| ps_j \| T_1 \| T_2 \| T_3 \| R_1 \| R_2 \| R_3 \| N_t \| N_v),$$

以及响应值：

$$s_{cs_i} := r_{cs_i} + c_H \times cs_i,$$
$$s_\zeta := r_\zeta + c_H \times \zeta,$$
$$s_\alpha := r_\alpha + c_H \times \alpha;$$

7）主机 H 将 $\sigma_{\mathrm{PBA}} := (\delta_m, ps_j, T_1, T_2, T_3, c_H, s_{cs_i}, s_\alpha, s_\zeta, N_t, N_v)$ 作为 PBA 属性证明发送给验证者 V.

4.5 证明校验(verify)算法

1）验证者 V 接收到示证者 P 的属性证明 σ_{PBA} 后，首先校验 δ_m 签名是否为一个有效 TPM 的签名值，即校验 $Verf_m(\delta_m, T_2, N_v \| N_t) = 1$ 是否成立；

2）验证者 V 然后查看 ps_i 是否满足其所需属性，若符合则验证者 V 利用发布者 PCI 的公钥 ipk 以及 σ_{PBA} 作以下运算：

$$\bar{R}_1 := g_1^{s_\alpha} T_2^{-c_H},$$
$$\bar{R}_2 := e(g_1, g_2)^{ps_i} e(T_1, g_2)^{-s_{cs_i}} e(g_1, g_2)^{s_\alpha} e(g_1, \omega)^{s_\zeta}$$
$$(e(g_1^{ps_i} h_1, g_2) e(T_1, \omega)^{-1})^{c_H},$$
$$\bar{R}_3 := u^{s_\zeta} T_3^{-c_H};$$

3）验证者 V 计算并校验：

$$H(ipk \| ps_i \| T_1 \| T_2 \| T_3 \| \bar{R}_1 \| \bar{R}_2 \| \bar{R}_3 \| N_t \| N_v) \stackrel{?}{=} c_H$$

是否成立；

4）以上 3 个步骤如果有一个不成立则输出 0(拒绝)，否则若以上步骤均成立则表明远程平台具有属性 ps_j，该校验算法输出 1(接受)，如果接受那么验证者 V 则运行证书校验算法.

4.6 证书校验(check)算法

若校验算法的输出为 1，验证者 V 可灵活选择以下 2 种撤销算法的一种作为证书状态校验算法：

1）离线撤销算法 (Check. Offline). 验证者 V 下载列表 $PCRL$，对于每一个属于该列表的表项 $RT_k = \{RT_k^{(1)}, RT_k^{(2)}\}, k \in \{0, \cdots, |\mathrm{PCRL}|\}$，$|\mathrm{PCRL}|$ 表示列表中表项个数，校验等式：

$$e(T_1, RT_k^{(2)}) / e(RT_k^{(1)}, RT_k^{(2)}) = e(T_2, g_2)$$

是否成立，如果成立，则输出 1(未撤销)，否则输出 0(已撤销)；

2）在线撤销算法 (Check. Online). 验证者 V 将该证明 σ_{PBA} 提交给证书校验者 $PCVC$，$PCVC$ 以 (T_1, T_3) 作为输入，使用校验密钥 η 计算：$T_1 / (T_3)^{1/\eta} = PCert_i$，并查询属性证书数据库 $dbase_{\mathrm{Cert}}$，找出匹配的证书是否被撤销，若 $Revflag = 1$ 则输出 1(未撤销)，否则输出 0(已撤销).

5 安全性证明及性能分析

5.1 安全性证明

在本节中依据第 3 节中的安全性定义，对本文所提出的 PBA 方案进行安全性分析，一般情况下，称 PBA 方案是安全的，即满足以下 3 个性质：正确性、配置隐私性和证明的不可伪造性.

5.1.1 正确性

正确性的证明包含 2 个部分：验证等式的正确性以及撤销等式的正确性. 以下对上述 2 个论断分别进行证明.

首先证明 Verify 等式的正确性. 在 PBA 方案中,由一个有效示证者 P 创建的属性证明可以由任意诚实验证者验证通过即有 $\bar{R}_1=R_1$, $\bar{R}_2=R_2$, $\bar{R}_3=R_3$.

$$\bar{R}_1=g_1^{s_\alpha}T_1^{-c_H}=g_1^{s_\alpha}g_1^{-c_H\times\alpha}=g_1^{r_\alpha+c_H\times\alpha-c_H\times\alpha}=R_1,$$

$$\bar{R}_2=e(g_1,g_2)^{ps_j}e(T_1,g_2)^{-s_{cs_i}}e(g_1,g_2)^{s_\alpha}e(g_1,\omega)^{s_\zeta}$$
$$(e(g_1^{ps_j}h_1,g_2)e(T_1,\omega)^{-1})^{c_H}=$$
$$R_2\times(e(T_1,g_2)^{-c_Hcs_i}e(g_1,g_2)^{c\times\alpha}e(g_1,\omega)^{c\times\zeta}$$
$$e(g_1^{ps_j}h_1,g_2)^{c_H}e(T_1,\omega)^{-c_H})=$$
$$R_2\times(e(T_1,\omega g_2^{cs_1})^{-c_H}e(g_1^{\zeta},\omega g_2^{cs_i})^{c_H}e(g_1^{ps_j}h_1,g_2)^{c_H})=$$
$$R_2\times 1^{c_H}=R_2,$$

$$\bar{R}_3=u^{s_\zeta}T_3^{-c_H}=u^{s_\zeta}u^{-c_H\times\zeta}=u^{r_\zeta+c_H\times\zeta-c_H\times\zeta}=R_3.$$

其次证明 Check 等式的正确性. 在方案中假设示证者 P 是有效的,并拥有未被撤销的属性证书 $Cert^*=(g_1^{ps}h_1)^{1/(\gamma+cs^*)}$ 向诚实的验证者证明其拥有所需的属性,那么验证者可以通过 Check 算法来校验其是否被撤销:

1) 离线证书校验. 验证者 V 根据 $PCRL$ 列表每一项 $RT_{cs^{(i)}}:=\{RT_{cs^{(i)}}^{(1)},RT_{cs^{(i)}}^{(2)}\}$,其中,$i\in\{0,\cdots,|PCRL|\}$来验证该证书是否被撤销,校验如下等式是否成立:

$$e((g_1^{ps}h_1)^{1/(\gamma+cs^*)}g_1^{\zeta},g_2^{cs^{(i)}})/$$
$$e((g_1^{ps}h_1)^{1/(\gamma+cs^{(i)})},g_2^{cs^{(i)}})=e(g_1^{\zeta\times cs^*},g_2),$$

如果属性证书未被撤销,那么该等式是不成立的,返回 1(未撤销);

2) 在线证书校验. $PCVC$ 根据验证者提交的证书信息,查找属性证书数据库中对应该属性证书的 $Revflag$ 值是否为 1,如果为 1 则返回 1(未撤销).

5.1.2 配置隐私性

定理 1. 假设有一个敌手 A 可以在概率多项式时间内,进行 q_H 次 Hash 质询以及 q_k 次 Attest 质询后以不可忽略的概率 ε 能够破坏 PBA 的配置隐私性,那么就存在一个多项式算法 S 能够以至多 ε/q_k 的概率解决 XDH 难题.

证明. 给定 S 一个扩展的 XDH 实例$(w,x=w^a,y=w^b,z=w^c,\chi_1,\chi_2)$,其中 $w\in_R G_1$, $a,b,c\in_R\mathbb{Z}_q$, χ_1,χ_2 或者$(\chi_1,\chi_2)\in_R G_1^2$ 或者$(\chi_1=w^{a\cdot b},\chi_2=w^{c\cdot b})$,$S$ 决定用其中一种情况与敌手 A 进行如下交互.

1) 初始化. 与 PBA 方案中的初始化算法类似,S 选择 2 个阶为素数 q 的 2 个群 $G_1=\langle g_1:=w\rangle$以及 $G_2=\langle g_2\rangle$,并选择一个 $\gamma\in_R\mathbb{Z}_q^*$ 作为 isk,那么公钥 ipk 为$(G_1,G_2,G_T,e,q,w,h_1,g_2,u:=z,\omega:=g^{\gamma}$"$_2$, $H)$,并将该公钥送给敌手 A.

2) Issue 质询. S 选择一个 $i^*\in\{1,\cdots,q_k\}$, q_k 为敌手质询的次数,S 将这个 i^* 作为挑战阶段敌手 A 挑战的目标值. 当敌手 A 向 S 发起第 $i\in\{1,\cdots,q_k\}$次质询时,S 要处理以下 2 种情况:

(1) 当 $i\neq i^*$ 时,S 选择 $cs_i\in CS$ 以及对应的 ps_i,并运行颁发协议,生成对应的属性证书 $PCert_i$,将该证书发送给敌手 A,然后将$(PCert_i,cs_i,ps_j)$存入质询列表 L_Q 中;

(2) 当 $i=i^*$ 时,由于 S 并不知道 $a=\log_w x$ 的值,因此随机选择 $ps\in_R\{0,1\}^{l_{ps}}$, $f\in_R G_1$,并令证书 $PCert^*=f$,S 将 $PCert^*$ 及 ps 作为应答值,并将$(PCert^*,ps)$保存到 L_Q 中.

3) Hash 质询. 在任何时候敌手 A 都可以对 Hash 函数进行质询,得到在$\mathbb{Z}_q^*$ 内均匀分布的随机挑战值 c.

4) Attest 质询. 敌手 A 选择 2 个随机值 N_v 和 N_t 发起第 i 次挑战. 同颁发质询一样,分为 2 种类型:

(1) 当 $i\neq i^*$ 时,S 根据质询列表 L_Q 中对应的属性证书以及(cs_i,ps_j)运行证明协议,计算出证明值 $\sigma_{\mathrm{PBA}}^{(i)}$发送给敌手 A,并将该证明值保存到对应询列表 L_Q 中.

(2) 当 $i=i^*$ 时,S 伪造如下的属性证明.

① S 选择随机数 $r\in_R\mathbb{Z}_q$,计算:

$$T_1^*:=PCert^*\times w^r,T_2^*:=x^r,T_3^*:=u^r.$$

② S 随机选择 $c,s_a,s_r,s_\alpha\in_R\mathbb{Z}_q$.

③ S 计算 $R_1^*:=w^{s_\alpha}T_2^*$, $R_3^*:=u^{s_r}(T_3^*)^c$ 以及

$$R_2^*:=e(w,g_2)^{ps}e(T_1^*,g_2)^{-s_\alpha}e(w,g_2)^{s_\alpha}e(w,\omega)^{s_r}\times$$
$$(e(w^{ps}h_1,g_2)e(T_1^*,\omega)^{-1})^c.$$

④ S 设置 $H(ipk\parallel ps\parallel T_1^*\parallel T_2^*\parallel T_3^*\parallel R_1^*\parallel R_2^*\parallel R_3^*\parallel N_t\parallel N_v):=c$,将该值作为敌手 A 的 Hash 质询. 如果敌手 A 在之前已经进行过对上述参数的 Hash 质询,那么 S 终止并退出输出 $\psi'\in_R\{0,1\}$.

⑤ 最后 S 将 $\sigma_{\mathrm{PBA}}^*:=(\delta_t,ps,T_1^*,T_2^*,T_3^*,c,s_a,s_a,s_r,N_t)$作为对敌手 A 的质询结果.

5) Corrupt 质询和 Revocation 质询. 敌手 A 提交索引 $i\neq i^*$,S 将对应的 cs_i 及撤销令牌 RT_{cs_i} 作为返回值;如果 $i=i^*$,S 终止并退出输出 $\psi'\in_R\{0,1\}$.

6) 挑战. 在挑战阶段,敌手 A 输出 N_t,N_v,i_0,i_1 4 个值,其中,i_0,i_1 是 A 之前所没有 Corrupt 质询和 Revocation 质询过的. 若 $i^*\notin(i_0,i_1)$,S 终止并

退出输出 $\psi' \in_R \{0,1\}$. 否则 S 选择一个值 $\phi \in_R \{0,1\}$ 使得 $i_\phi = i^*$，并按如下步骤生成属性证明：

① S 选择一个随机数 $r \in_R \mathbb{Z}_q$，利用证书 $PCert^*$ 计算 $T_1^* := PCert^* \times y^r$，$T_2^* := \chi_1^r$，$T_3^* := \chi_2^r$；

② 算法的其他步骤与上述 Attest 质询的步骤一致，S 将挑战的证明值 σ_{PBA}^* 发送给 A.

7）输出. 敌手 A 最后将输出关于该证明的猜测 ϕ' 值，如果 $\phi' = \phi$，那么 S 输出它的判定 $\psi' = 1$，即判断出 $(\chi_1 = w^{a \cdot b}, \chi_2 = w^{c \cdot b})$，否则判定 $\psi' = 0$，表示 $(\chi_1, \chi_2) \in_R G_1^2$.

下面对 S 的判定优势进行评估，用 $\psi \in_R \{0,1\}$ 表示 $\psi = 1$ 时 $(\chi_1 = w^{a \cdot b}, \chi_2 = w^{c \cdot b})$，$\psi = 0$ 时 $(\chi_1, \chi_2) \in_R G_1^2$. 并用 $abort$ 代表 S 终止退出的事件，即有 $Pr[\psi = \psi' | abort] = 1/2$. 假设 S 并没有终止退出，那么若 $\psi = 0$，表示 $(\chi_1, \chi_2) \in_R G_1^2$，则该挑战签名 σ_{PBA}^* 中没有任何关于 a, c 的信息，因此显然有 $Pr[\psi' = 0 | \neg abort \wedge \psi = 0] = 1/2$；若 $\psi = 1$，表示 $(\chi_1 = w^{a \cdot b}, \chi_2 = w^{c \cdot b})$，则 S 与 A 对上述真实游戏进行了完美的模拟，并且 A 最终以概率 ε 正确地猜出了 $\phi' = \phi$，因此就有 $Pr[\psi' = 1 | \neg abort \wedge \psi = 1] = 1/2 + \varepsilon$.

综上，S 对于 XDH 难题的判定优势概率如下：

$$
\begin{aligned}
&| Pr[S((w, x = w^a, y = w^b, z = w^c, \chi_1, \chi_2) \wedge \\
&(\chi_1 = w^{a \cdot b}, \chi_2 = w^{c \cdot b})) = 0] - \\
&Pr[S((w, x = w^a, y = w^b, z = w^c, \\
&\chi_1, \chi_2) \wedge (\chi_1, \chi_2) \in_R G_1^2)) = 0] | = \\
&| Pr[\psi' = 0 | \psi = 1] - Pr[\psi' = 0 | \psi = 0] | = \\
&| 1 - Pr[\psi' = 1 | \psi = 1] - Pr[\psi' = 0 | \psi = 0] | = \\
&| 1 - Pr[abort] Pr[\psi' = 1 | abort \wedge \psi = 1] - \\
&Pr[\neg abort] Pr[\psi' = 1 | \neg abort \wedge \psi = 1] - \\
&Pr[abort] Pr[\psi' = 0 | abort \wedge \psi = 0] - \\
&Pr[\neg abort] Pr[\psi' = 0 | \neg abort \wedge \psi = 0] | = \\
&| 1 - Pr[abort](1/2 + 1/2) - \\
&Pr[\neg abort]((1/2 + \varepsilon) + 1/2) | = \\
&Pr[\neg abort]\varepsilon.
\end{aligned}
$$

$Pr[\neg abort]$ 代表 S 没有终止退出的概率，首先 S 在 Attest 质询中构造特殊证明时退出的概率最多为 q_H/q，这是一个可以忽略的值；其次在 Corrupt 质询和 Revocation 质询中，S 没有退出的概率为 $1/q_k$；最后在挑战阶段，仅当 A 选择了 $i^* \in (i_0, i_1)$ 时 S 才不会退出，这个概率为 $1/q_k$. 计算如下概率 $Pr[\neg abort] \geqslant (1 - q_H/q) \times (1 - 1/q_k) \times 1/q_k \approx 1/q_k$，因此 S 判定 XDH 难题的概率至少为 ε/q_k.

5.1.3 证明的不可伪造性

定理 2. 假设有一个敌手 A 可以在概率多项式时间内，进行 q_H 次 Hash 质询以及 q_k 次 Attest 质询后，以不可忽略的概率 ε 能够破坏 PBA 证明的不可伪造性，那么就存在一个多项式算法 S 能够以最多 ε 的概率解决 l-SDH 难题.

证明. 证明的思路是给定 S 一个 l-SDH 难题实例，即 $l-1$ 个 SDH 对 (A_i, x_i)，其中，$i \in \{1, \cdots, l-1\}$，S 通过与敌手 A 的交互，可以产生一对新的 SDH 对 (A, x)，那么 S 就可以解决 l-SDH 难题. 具体证明过程如下：

1）初始化. S 运行 Setup(1^k) 算法，产生的公钥 ipk 为 $(G_1, G_2, G_T, e, q, g_1, h_1, g_2, u = g_1^\eta, \omega, H) \wedge g_1 = \varphi(g_2)$，校验密钥 ivk 为 η，注意 S 并不知道 $isk = \gamma$. 并将 $l-1$ 个 SDH 对 (A_i, cs_i) 给予 S，其中 cs_i 为合理的配置信息，并且这些配置信息有对应的属性 ps_i. S 选择一个合理的配置-属性对 (cs, ps) 以及一个随机数 $a \in_R \mathbb{Z}_q^*$，并将 ipk 中的 h_1 进行如下的变换：

$$h_1 = h^* := \varphi((\omega \times g_2^{cs})^a \times g_2^{-ps}) = g_1^{a(\gamma + cs) - ps}.$$

S 将变换之后的 h_1 及其他公钥参数发送给 A.

2）Issue 质询. 根据定义 6，对于敌手 A 的 $q_k = l$ 次质询请求，S 对 2 种情况分别进行不同的应答.

① S 随机选择一次敌手的质询 i^*，并设置 $(PCert_{i^*} = g_1^a, cs_{i^*} = cs, ps_{i^*} = ps)$，将此作为对 A 的应答；

② 在其他 $l-1$ 次质询中，S 使用一个 SDH 对 (A_i, cs_i) 及与之对应的 ps_i，计算如下的证书：

$$
\begin{aligned}
PCert_i &:= (g_1^{ps_i} h_1)^{1/(\gamma + cs_i)} = \\
&(g_1^{ps_i + a(\gamma + cs) - ps})^{1/(\gamma + cs_i)} = \\
&A_i^{ps_i + a(\gamma + cs) - ps} = A_i^{ps_i - a(cs_i - cs) - ps} \times g_1^a.
\end{aligned}
$$

S 将生成的 $(PCert_i, ps_i)$ 作为对 A 的返回值.

3）Hash 质询. 在任何时候敌手 A 都可以对 Hash 函数进行质询，得到在 $\mathbb{Z}_q^*$ 内均匀分布的随机挑战值 c.

4）Attest 质询. 敌手 A 提交索引 i，2 个随机值 N_v, N_t，S 运行 Attest 协议产生对应的属性证明 $\sigma_{\mathrm{PBA}}^{(i)}$，并将其发送给 A.

5）Corrupt 质询. 敌手 A 提交索引 i，S 将对应的元组 cs_i 作为返回值，并将 $(PCert_i, cs_i, ps_i)$ 元组存放 L_q 质询列表中，并更新 $PCRL$ 列表.

6）输出. 在进行了 q_k 次质询后，敌手 A 输出了一个关于消息 $N_t^* \parallel N_v^*$ 及有效属性 ps^* 的属性证明 σ_{PBA}^*.

S 对签名 σ^*_{PBA} 有效性进行校验,若有 $Verify(ipk,\sigma^*_{\mathrm{PBA}},N_t^* \parallel N_v^*) \wedge \mathrm{Check}(\sigma^*_{\mathrm{PBA}},PCRL) \rightarrow 1$ 成立,那么 S 为计算出 $PCert^*,cs^*$,需对 A 进行重置(rewind)操作,即 S 控制 Hash 函数,H 利用同样的 T_1^*,T_2^*,T_3^* 以及 R_1^*,R_2^*,R_3^* 设置 2 个不同的 Hash 值 c',c'',此时 A 会生成不同的 $s'_{cs^*},s'_\alpha,s'_\zeta$ 以及 $s''_{cs^*},s''_\alpha,s''_\zeta$,由此 S 通过计算 $c'=c'-c''$ 得出:$cs^*=(s'_{cs^*}-s''_{cs^*})/c'$;否则 S 将终止并退出. S 利用校验密钥 ivk 从 T_2^* 中提取出证书 $Cert^*$ 即

$$PCert^* = T_2^*/(T_3^*)^{1/\eta} = (g_1^{ps^*}h_1)^{1/(\gamma+cs^*)}.$$

这样 S 得到了一个有效的$(PCert^*,cs^*)$对,值得注意的是,cs^* 可能不属于之前质询过的任何 cs_i 及 cs,那么有以下 2 种能力敌手类型需要考虑:

① 敌手类型 1. 生成的 $cs^* \notin \{cs_1,\cdots,cs_{l-1},cs\}$,$S$ 按如下方式计算一个有效的 SDH 实例:

$$A^* := (PCert^* \times g_1^{-a})^{1/(ps^*-a(cs^*-cs)-ps)} = g_1^{1/(\gamma+cs^*)}.$$

S 得到了一个额外的 SDH 实例(A^*,cs^*). 注意,这里要保证$(ps^*-a(cs^*-cs)-ps)\neq 0$,否则 S 将终止并退出.

② 敌手类型 2. 生成的 $cs^* \in \{cs_1,\cdots,cs_{l-1},cs\}$,并且 $PCert^* \notin \{PCert_1,\cdots,PCert_{l-1},PCert_{i^*}\}$,这时如果 $cs^* \neq cs$,S 终止并退出;如果 $cs^*=cs$,那么 S 计算:

$$A^* := (PCert^* \times g_1^{-a})^{1/(ps^*-ps)} = g_1^{1/(\gamma+cs)}.$$

S 得到了一个有效的 SDH 实例(A^*,cs). 注意,这里要保证 $ps^* \neq ps$,否则 S 将终止并退出,同时这意味敌手产生了一个伪造其他属性 ps^* 的属性证明.

上述敌手类型 1 中 S 的终止退出概率为 $1/q$;敌手类型 2 中 S 的终止退出概率为 $1/q_k^2$. 综上,如果敌手 A 可以以 ε 概率伪造出一个有效的属性证明,那么 S 可以以 $\varepsilon((1-1/q)+(1-1/q_k^2))/2<\varepsilon$ 的概率创建新的 SDH 实例,因此 S 可以解决 l-SDH 难题.

5.2 性能分析

在本节将 PBA 方案与现有具有代表性的 PBA 方案[11-14]进行了比较. 在本文中评估的指标有 2 种:证明值长度及计算代价. 在进行方案性能对比之前,各方案的安全级别统一定义为 128 b 级别. 另外需要定义安全参数长度及相关算法的计算代价等,参见如下:

1) 安全参数定义. (1)双线性配对算法安全参数,本文采用了高效的 BN 椭圆曲线算法[24],那么在 128 b 级别下 $l_q=255$ b,群 G_1 中的元素大小为 256 b,群 G_T 中元素大小为 3 072 b;(2)Hash 算法采用 SHA1- 160 b,因此有 $l_H=160$ b,$l_{cs}=160$ b,$l_{ps}=160$ b;(3)RSA 算法的安全参数,在 128 b 级别下 $l_n=2048$ b,$l_p=1632$ b,$l_q=208$ b,那么在文献[11-13]中对应的 $l_e=368$ b,$l_{e'}=120$ b,$l_v=2536$ b;(4)TPM 产生的身份签名长度定义为$|\delta_m|$.

2) 计算代价定义. (1)配对运算指数算法,采用 $S_i,M_i(i=1,2,T)$ 分别表示在群 G_1,G_2,G_T 中的一次单指数运算及一次多指数运算,使用 BN 椭圆曲线 S_T,M_T 计算代价是 S_1,M_1 约 14 倍,S_2,M_2 是 S_1,M_1 约 4 倍[25],另外 P 来表示一次配对运算,很明显配对运算要比在群中的运算代价要大很多[24];(2)RSA 运算指数算法,采用 $S_i,M_i(i=N,P)$ 来表示在循环群 G_N,G_P 里的一次单指数运算及一次多指数运算,值得说明的是 S_N,M_N 与 S_T,M_T 的计算代价几乎相同[25];(3)其他运算代价,用$\langle\delta_m\rangle$表示一次 TPM 的身份签名操作(AIK 或 DAA)计算代价,用$|PCRL|$表示在 PCRL 列表中表项的个数,$|N_{cs}|$表示所接受的配置信息数.

值得说明的是,在各 PBA 方案比较中,将 Attest 算法分为主机 H 和 TPM M 2 个部分,这是因为它们的计算能力不一样;另外,本文的 PBA 方案证明方法中的 Attest,Verify,离线撤销算法都可以进行计算上的简化:

1) Attest 中 R_2 的计算代价为 $4P$,可以将 R_2 的计算进行如下简化:

$$R_2 = e(g_1,g_2)^{ps_i+r_\alpha}e(T_1,g_2)^{-r_{cs_i}}e(g_1,\omega)^{r_\zeta},$$

其中 $e(g_1,g_2),e(g_1,\omega),e(T_1,g_2)$ 可以预先进行计算,因此 R_2 的计算代价减少为 $3S_T$.

2) Verify 中 $\bar{R}_2$ 计算代价可以减少为 $1P+3M_T$,即

$$\bar{R}_2 = e(T_1,\omega^{-c_H}g_2^{-s_{cs_i}}) \times e(g_1,g_2)^{ps_i+ps_i\times c_H+s_\alpha}e(g_1,\omega)^{s_\zeta}e(h_1,g_2)^{c_H},$$

其中 $e(g_1,g_2),e(g_1,\omega),e(h_1,g_2)$ 这 3 个值都是可以预先计算的.

3) Check 中的离线校验算法可以简化为:校验等式中的 $e(RT_k^{(1)},RT_k^{(2)})$ 可以预先由服务器计算,并将该值以及 RT_k 作为 $PCRL$ 列表里的表项,那么整个撤销算法的计算代价为$(1+|PCRL|)\times P$.

基于上述的定义,本文 PBA 方案与其他 PBA 方案[11-14]比较如表 1 所示:

Table 1 Computational Cost and Attestation Length Comparisons of Property-Based Attestation Schemes

表 1 PBA 方案计算代价及证明长度比较

Performance	Scheme in Ref[11]	Scheme in Ref[12]	Scheme in Ref[13]	Scheme in Ref[14]	Our Scheme
Attest(Host)	$1M_P+2S_N+1M_N+[\delta_m]$	$\lvert N_{cs}\rvert\times M_P+[\delta_m]$	$3S_N+4M_N$	$2M_1+4S_T+2S_1+[\delta_m]$	$4S_1+3S_T+[\delta_m]$
Attest(TPM)	$1M_P$	$1M_P$	$2S_N+2M_N+[\delta_m]$	$1S_T+1M_T$	$1S_1$
Verify	$1M_N+1M_P$	$\lvert N_{cs}\rvert\times M_P$	$1S_N+5M_N$	$4M_1+4P$	$2M_1+1P+3M_T$
Check. Online			$1S_N+2M_N$	$1M_1$	$1S_1$
Check. Offline	$3S_P+4M_P+\lvert PCRL\rvert(1S_P+3M_P)$				$(1+\lvert PCRL\rvert)\times P$
Attestation Len/b	$7162+\lvert\delta_m\rvert$	$(\lvert N_{cs}\rvert+1)\times 208+\lvert\delta_m\rvert$	$23433+\lvert\delta_m\rvert$	$6027+\lvert\delta_m\rvert$	$2093+\lvert\delta_m\rvert$

由表 1 直观可以看出，本文提出的 PBA 方案在计算代价及证明值长度方面明显优于其他 PBA 方案，尤其是在 Attest 协议中 TPM 的计算代价要明显小于其他方案，文献[14]方案中 TPM 的计算代价是 $1S_T+1M_T$，而在 32 b-33 MHz 的模拟 TPM 上计算 $1S_T$ 需要 3 s 左右的时间[25]，$1S_T+1M_T$ 则会大于 6 s，因此会影响 PBA 方案的整体实用性能，而本文提出方案中的 TPM 计算代价仅为 $1S_1$.

6 进一步的讨论

针对 2.3 节中提到的第 2 种情况，本地验证者撤销[15]存在向后关联性的问题[16]，文献[16]提出了向后不可关联(back unlinkability)的概念，即若群中一个成员被撤销，那么在撤销之前由这名成员之前所产生的群签名仍保持匿名性. 具体的方法是利用了时间间隔(time intervals)的概念，对于所有的时间间隔点 $1,\cdots,T$，群管理员会生成关于每个群成员的撤销令牌 $RT_i(i\in\{1,\cdots,T\})$，目的是实现群签名的可追踪性. 若有群成员在时间间隔点 j^* 被撤销，那么群管理者将所有大于等于 j^* 时间间隔点的关于该成员的撤销令牌 $RT_j(j\geqslant j^*)$ 发布到撤销列表 RL 中，这样对于验证者可以根据不同时间间隔来下载对应时间间隔的 RL_j 列表，以判断该名成员在时间间隔点 j 是否被撤销. 与上述撤销令牌生成方法稍有不同，在本文提出的 PBA 方案中，发布者 PCI 仅需在某个撤销时间点 j^* 之后生成关于撤销的属性证书撤销令牌即可，这是因为如果追踪一个属性证书非法持有者，可以利用证书校验者 $PCVC$ 的 ivk 来实现，这里不作赘述.

根据向后不可关联的思想[16]，对于本文所提出 PBA 方案中的算法可以作如下的改进：

1) Setup 算法改进如下：

发布者 PCI 选择 T 个随机数 $r_1,\cdots,r_T$，然后生成对应的 $v_1:=g_1^{r_1},\cdots,v_T:=g_1^{r_T}$，则 PCI 的公钥为

$ipk:=\{G_1,G_2,G_T,e,q,g_1,h_1,v_1,\cdots,v_T,g_2,u,\omega,H\}$，其他参数不变.

2) Issue 算法不作改进.

3) Revoke 算法改进如下：

若有属性证书在时间间隔点 j^* 被撤销，该算法中的步骤 1 不作改进，而步骤 2 改进.

则发布者 PCI 发布以下参数作为撤销令牌：

$$RT_{cs_i}^j:=\{RT_{cs_i}^{j(1)},RT_{cs_i}^{j(2)}\}=\{PCert_i,g_2^{r_j\times cs_i}\},$$

其中，$j\geqslant j^*$ 并将该撤销令牌更新到 $PCRL_j$ 列表中.

4) Attest 算法改进如下：

在时间间隔点 j，TPM 计算承诺值 $C_{cs_i}:=v_j^{cs_i\times\xi}$，则知识证明的过程为

$$SPK_j\{(cs_i,\zeta,PCert_i):$$
$$T_1=PCert_i\times g_1^{\xi},$$
$$T_2=v_j^{cs_i\times\zeta},$$
$$T_3=u^{\zeta},$$
$$e(PCert_i,\omega g_2^{cs_i})=e(g_1^{ps_i}h_1,g_2)\}(N_v,N_t).$$

并令 $R_1:=v_{j^*}^{r_\delta}$，其他过程及参数保持不变.

5) Verify 算法改进如下：

在时间间隔点 j，验证者将 $\bar{R}_1$ 更改为 $\bar{R}_1:=v_{j^*}^{s_\delta}T_2^{-c_H}$，其他的过程及参数保持不变.

6) Check 算法改进如下：

在时间间隔点 j：① 离线校验方法为根据 $PCRL_j$ 列表中的撤销令牌信息，作如下校验：

$$e(T_1,RT_k^{j(2)})/e(RT_k^{j(1)},RT_k^{j(2)})=e(T_2,g_2),$$

②在线校验方法不做改进.

通过以上的改进，本文提出的 PBA 方案具备了向后不可关联性 BU 的特点. 本节改进后的 PBA 方案的安全性证明过程同 5.1 节，性能分析同 5.2 节.

7 结 论

基于属性的证明解决了在认证过程中平台配置信息泄露的问题.本文基于属性证明的思想并利用本地验证者撤销的方法,提出了一个高效的具有灵活撤销机制的 PBA 方案,该方案提供了离线和在线的属性证书状态校验机制,并且在随机预言模型下证明该方案是安全的.在性能方面,本文提出的 PBA 方案比现有的 PBA 方案更高效且具有可靠的属性证书撤销功能.并且在第 6 节对 PBA 方案作了进一步的讨论,使其满足在 2.3 节中提出的第 2 种情况.在未来的工作中,将基于现有的证书管理体制并结合本文所提出的 PBA 方案,搭建面向实际应用环境的 PBA 体系结构框架.

参 考 文 献

[1] Trusted Computing Group. TPM Main Part 1: Design Principles Specif-cation, Version 1.2 Revision 62 [S]. Geneva: International Organization for Standardization (ISO), 2009

[2] Jaeger T, Sailer R, Shankar U. PRIMA: Policy-reduced integrity measurement architecture [C] //Proc of the 11th ACM Symp on Access Control Models and Technologies. New York: ACM, 2006: 19-28

[3] Vivek H, Deepak C, Michael F. Semantic remote attestation: A virtual machine directed approach to trusted computing [C] //Proc of the 3rd Conf on Virtual Machine Research and Technology Symp. Berkeley, CA: USENIX Association, 2004: 3-3

[4] Marchesini J, Smith S, Wild O, et al. Open-source applications of TCPA hardware [C] //Proc of the 20th Annual Computer Security Applications Conf. Piscataway, NJ: IEEE, 2004: 294-303

[5] Seshadri A, Perrig A, Doorn L V, et al. SWATT: Software-based attestation for embedded devices [C] //Proc of the IEEE Security & Privacy Conf. Piscataway, NJ: IEEE, 2004: 272-282

[6] Poritz J, Schunter M, Herreweghen E, et al. Property attestation scalable and privacy—Friendly security assessment of peer computers, RZ3548 [R]. Zürich: IBM Research, 2004

[7] Sadeghi A, Stuble C. Property-based attestation for computing platforms: Caring about properties, not mechanisms [C] //Proc of the 2004 Workshop on New Security Paradigms. New York: ACM, 2004: 67-77

[8] Kühn U, Selhorst M, Stüble C. Realizing property-based attestation and sealing on commonly available hard-and software [C] //Proc of the 2007 ACM Workshop on Scalable Trusted Computing. New York: ACM, 2007: 50-57

[9] Korthaus R, Sadeghi A, Stüble C. A practical property-based bootstrap architecture [C] //Proc of the 2009 ACM Workshop on Scalable Trusted Computing. New York: ACM, 2009: 29-38

[10] Kostiainen K, Asokan N, Ekberg J. Practical property-based attestation on mobile devices [G] //LNCS 6740: Proc of TRUST 2011. Berlin: Springer, 2011: 78-92

[11] Chen L, Landfermann R, Lohr H, et al. A protocol for property-based attestation [C] //Proc of the 1st ACM Workshop on Scalable Trusted Computing. New York: ACM, 2006: 7-16

[12] Chen L, Lohr H, Manulis M, et al. Property-based attestation without a trusted third party [G] //LNCS 5222: Proc of the 11th Int Conf on Information Security. Berlin: Springer, 2008: 31-46

[13] Qin Yu, Feng Dengguo. Component property based remote attestation [J]. Journal of Software, 2009, 20(6): 1625-1641 (in Chinese)
(秦宇,冯登国. 基于组件属性的远程证明[J]. 软件学报, 2009, 20(6): 1625-1641)

[14] Feng Dengguo, Qin Yu. A property-based attestation protocol for TCM [J]. Science China Service F: Information Sciences, 2010, 53(3): 454-464

[15] Boneh D, Shacham H. Group signatures with verifier-local revocation [C] //Proc of the 11th ACM Conf on Computer and Communications Security. New York: ACM, 2004: 168-177

[16] Nakanishi T, Funabiki N. A short verifier-local revocation group signature schemes with backward unlinkability [J]. IEICE Trans on Fundamentals, 2007, E90-A(9): 1793-1802

[17] Wei Lingbo, Liu Jianwei. Shorter verifier-local revocation group signature with backward unlinkability [G] //LNCS 6487: Proc of Pairing 2010. Berlin: Springer, 2010: 136-146

[18] Zhou Sujing, Lin Dongdai. Shorter verifier-local revocation group signatures from bilinear maps [G] //LNCS 4301: Proc of CANS 2006. Berlin: Springer, 2006: 126-143

[19] Boneh D, Boyen X, Shacham H. Short group signatures [G] //LNCS 3152: Proc of CRYPTO 2004. Berlin: Springer, 2004: 41-55

[20] Boneh D, Boyen X. Short signatures without random oracles and the SDH assumption in bilinear groups [J]. Journal of Cryptology, 2008, 21(2): 149-177

[21] Bellare M, Shi Haixia, Zhang Chong. Foundations of group signatures: The case of dynamic groups [G] //LNCS 3376: Proc of CT-RSA 2005, Berlin: Springer, 2005: 136-153

[22] Chaum D, Heijst E. Group signatures [G] //LNCS 547: Proc of EUROCRYPT'91. Berlin: Springer, 1991: 241-246

[23] Chen L, Page D, Smart N. On the design and implementation of an efficient DAA scheme [G] //LNCS 6035: Proc of CARDIS 2010. Berlin: Springer, 2010: 223-237

[24] Barreto P, Naehrig M. Pairing-friendly elliptic curves of prime order [G] //LNCS 3897: Proc of the 12th Int Workshop on Selected Areas in Cryptography. Berlin: Springer, 2005: 319-331

[25] Chen L, Morrissey P, Smart N. Pairings in trusted computing [G] //LNCS 5209: Proc of Pairing 2008. Berlin: Springer, 2008: 1-17

Zhou Fucai, born in 1964. Professor and PhD superviosr. His current research interests include cryptograhpy and information security and assurance.

Yue Xiaohan, born in 1982. PhD. His current research interests include information security and trusted computing.

Bai Hongbo, born in 1970. PhD candidate. His current research interests include public key cryptography and privacy preserving.

Xu Jian, born in 1978. PhD candidate and lecturer. Member of China Computer Fedaration. His main research interests include cryptography and network security.