基于身份的可穿刺签名方案

杨冬梅，陈越，魏江宏，胡学先
（信息工程大学数据与目标工程学院，河南 郑州 450001）

摘 要：针对已有前向安全的基于身份签名（IBS）方案在实际可用性和密钥更新效率方面存在的不足，基于可穿刺公钥加密方案的思想，提出了基于身份的可穿刺签名（IBPS）方案。具体而言，首先给出了 IBPS 的形式化定义和安全性定义，然后基于布隆过滤器构造了一个具体的 IBPS 方案。在计算性 Diffie-Hellman 假设下，所提方案在随机预言模型下满足存在不可伪造性。性能分析与仿真实验表明，所提方案比传统的前向安全 IBS 方案提供了更实用的细粒度前向安全性，且密钥更新过程更高效。

关键词：私钥泄露；基于身份的签名；可穿刺签名；布隆过滤器
中图分类号：TP309.2
文献标识码：A
DOI: 10.11959/j.issn.1000−436x.2021223

Identity-based puncturable signature scheme

YANG Dongmei, CHEN Yue, WEI Jianghong, HU Xuexian
School of Data and Target Engineering, Information Engineering University, Zhengzhou 450001, China

Abstract: To overcome the shortcomings of available forward-secure identity-based signature (IBS) scheme in terms of actual practicability and secret key update efficiency, the notion of identity-based puncturable signature (IBPS) scheme was proposed based on the idea of puncturable public-key encryption scheme. Specifically, the syntax and security notion of IBPS were given, and then a concrete IBPS scheme was constructed based on Bloom filter. Under the computational Diffie-Hellman assumption, the proposed scheme achieved the existential unforgeability in the random oracle model. The performance analysis and implementation results demonstrate that, compared with traditional forward-secure IBS schemes, the proposed scheme provides more practical fine-grained forward secrecy, and has higher efficiency of secret key update.

Keywords: key exposure, identity-based signature, puncturable signature, Bloom filter

1 引言

在基于公钥基础设施的密码体制中，权威机构通过数字签名为用户颁发公钥证书，保证了用户公钥的真实性和可信性。与此同时，用户证书的颁发、存储和撤销以及认证核查等工作也带来了繁重的管理开销。为此，Shamir[1]在 1984 年首次提出了基于身份的签名（IBS, identity-based signature）方案，即以用户的个人信息（如身份证号、手机号、电子邮箱地址等）作为其公钥，而用户私钥由一个密钥生成中心利用系统主密钥来生成，签名的验证也只需要签名用户的公钥对所对应的身份信息，从而避免了对公钥基础设施的依赖。

随后，围绕 IBS 方案安全性增强、效率提升和功能扩展等方面，密码学者提出了很多新的 IBS 方案，如 Cha[2]给出了完整的 IBS 方案和该方案在随机预言模型下的安全性证明，Paterson 等[3]在标准模型下构造了第一个高效的 IBS 方案，杨小东等[4]提出

收稿日期：2021−09−01；**修回日期**：2021−11−03
通信作者：陈越，cyue2008@126.com
基金项目：国家自然科学基金资助项目（No.62172433, No.62172434）
Foundation Item: The National Natural Science Foundation of China (No.62172433, No.62172434)

了强不可伪造的基于身份服务器辅助验证签名方案，刘翔宇等[5]提出了第一个紧致安全的 IBS 方案，田苗苗等[6]提出了格上基于身份的增量签名方案以及侯红霞等[7]提出了素数阶群上基于非对称对的身份基环签名。在这些传统的 IBS 方案中，假设用户私钥都是绝对安全的。但在现实应用环境中，随着网络攻击手段的多样化，用户私钥泄露事件趋于高发。另一方面，一旦用户私钥被攻击者窃取，攻击者就可以产生任何消息的合法签名，导致用户之前生成的所有签名都将失效。

为降低数字签名方案中由私钥泄露所带来的损失，Anderson[8]在 1997 年首次引入了前向安全数字签名方案的概念，即要求在用户私钥泄露后，之前所生成的签名是不能被伪造的，但他并没有给出具体构造。Bellare 等[9]在 1999 年的美密会上首次对这种密码学原语进行了形式化定义，并基于 RSA（Rivest Shamir Adleman）群上的整数分解问题给出了一个具体构造。随后，学者对前向安全的数字签名方案进行了大量而又深入的研究，提出了很多在安全性和效率方面具有各自优势的相关方案，如 Itkis 等[10]提出了具有高效签名和验证效率的前向安全签名方案，Kozlov 等[11]给出了密钥能快速更新的前向安全签名方案，Libert 等[12]构造了不可信更新环境中前向安全的签名方案，Abdalla 等[13]对文献[9]方案的私钥规模进行了优化等。以上这些前向安全的数字签名方案均是基于公钥证书的。为避免对公钥基础设施的依赖，Yu 等[14]在 IBS 中引入了前向安全性，形式化地定义了前向安全的基于身份签名方案，并给出了一个具体构造。

然而，如 Green 等[15]所指出，上述传统的通过周期性更新密钥实现前向安全性的方法不够灵活，导致其在实际部署中不可用。具体而言，传统的前向安全 IBS 方案[14]以固定的时间周期为单位（如每小时、每天等）更新用户私钥，这导致当前时间周期的私钥泄露之后，攻击者仍能伪造当前时间周期内的签名。此外，传统的前向安全 IBS 方案采用二叉树结构管理用户密钥和时间周期，使这些方案的密钥更新效率较慢。

为克服已有前向安全 IBS 方案在密钥更新灵活性和效率方面存在的不足，本文提出了基于身份的可穿刺签名（IBPS, identity-based puncturable signature）方案。这种新的签名方案实现了细粒度的前向安全性，即每当用户完成对某个消息的签名之后，立即对私钥进行更新，使更新后的私钥不能再用来对同样的消息进行签名，保证了私钥泄露之前所生成签名的安全性。本文主要贡献总结如下。

1) 对传统的 IBS 机制进行了扩展，给出了 IBPS 机制的形式化定义，并通过安全模型刻画了这种签名方案的存在不可伪造性。

2) 将基于布隆过滤器的可穿刺公钥密码方案的构造思想应用到 IBS，基于 Paterson 等[3]的 IBS 方案构造了一个具体的 IBPS 方案，其密钥更新过程只需进行一次布隆过滤器中的元素删除操作。

3) 基于计算性 Diffie-Hellman（CDH, computational Diffie-Hellman）假设，在随机预言模型下证明了所构造 IBPS 方案满足存在不可伪造性，并通过仿真实验展示了方案的实际性能。

2 相关工作

Yu 等[14]基于 Waters[16]提出的基于身份加密方案，利用二叉树结构首次构造了前向安全的 IBS 方案。在这种构造中，系统的生命周期被离散化为 N 个子周期，通过单向地周期性更新私钥和删除前一周期的私钥，用户可以在每个时间周期内使用不同的私钥，从而实现了前向安全性。沿用类似的方法，魏江宏等[17]提出了前向安全的密文策略基于属性加密方案；Wei 等[18]在前向安全 IBS 的基础上通过用户撤销引入了后向安全性；Oh 等[19]进一步考虑了系统主密钥泄露的问题，提出了能同时抵抗用户私钥泄露和系统主密钥泄露的 IBS 方案；杨小东等[20]在标准模型下提出了可撤销的基于身份的代理重签名方案。

上述方案均采用了二叉树结构，导致私钥更新的开销达到 $\mathcal{O}(\log N)$。

事实上，上述方案的构造方法在一定程度上源于前向安全的公钥加密方案[21]，因此也就继承了其局限性。为实现更实用的细粒度前向安全性，Green 等[15]提出了可穿刺公钥加密方案的概念，并基于支持非单调访问结构的基于属性加密方案构造了具体的可穿刺公钥加密方案。随后，Wei 等[22]构造了密文长度固定的可穿刺的基于身份加密方案。为提高可穿刺加密方案中私钥穿刺的效率，Derler 等[23]进一步提出了布隆过滤器加密方案，并基于不同密码学原语给出了多种构造方案。

类似于可穿刺的公钥加密方案，可穿刺签名方案同样能提供签名的细粒度前向安全性，由 Bellare

等[24]首次提出。但是，他们给出的通用构造依赖于不可区分混淆和单向函数，具有较高的计算开销，在实际中不可用。Halevi 等[25]虽然也提出了一个可穿刺签名方案，但是需要在私钥穿刺操作后更新公钥，带来了较大的公钥管理负担。Li 等[26]基于布隆过滤器加密方案构造了具有较高穿刺效率的可穿刺签名方案，并将其应用到了区块链中的权益证明协议。

3　预备知识

3.1　布隆过滤器

布隆过滤器[27]是一种经典的随机数据结构，通过一个简短的二进制序列 T 来表示一个集合 $\mathcal{S}$，并能高效地判定一个元素 s 是否属于该集合。具体而言，在集合 $\mathcal{S}$ 上进行查询时，若 $s\in\mathcal{S}$，则以概率 1 返回 1，表示 s 确实在集合 $\mathcal{S}$ 中；若 $s\notin\mathcal{S}$，则以概率 $1-\delta$ 返回 0，表示 s 不在集合 $\mathcal{S}$ 中，其中 δ 为假阳性概率，即当 $s\notin\mathcal{S}$ 时，仍以 δ 的概率返回 1。

定义 1　布隆过滤器[27]。令 $\mathcal{U}$ 为元素取值空间，其上的布隆过滤器 B = (BFGen,BFUpdate, BFCheck) 由 3 个子算法组成，具体定义如下。

BFGen (ℓ,k)，生成算法。以 2 个正整数 $\ell,k\in\mathbb{N}$ 为输入，首先选择 k 个一般的 Hash 函数 $H_1,\cdots,H_k$，其中 $H_j:\mathcal{U}\to[\ell]$，并令 $\boldsymbol{H}=(H_j)_{j\in[k]}$ 和 $T=0^\ell$（长度为 ℓ 且各个位置均为 0 的字符串），最后输出 $(\boldsymbol{H},T)$。

BFUpdate $(\boldsymbol{H},T,u)$，更新算法。以 Hash 函数族 $\boldsymbol{H}=(H_j)_{j\in[k]}$、过滤器当前状态 $T\in\{0,1\}^\ell$ 和一个元素 $u\in\mathcal{U}$ 为输入，首先令 $T'\leftarrow T$，然后对任意的整数 $j\in[k]$，令 $T'[H_j(u)]=1$，其中 $T'[i]$ 表示 T' 的第 i 位，最后返回更新后的过滤器状态 T'。

BFCheck $(\boldsymbol{H},T,u)$，检验算法。同样以 Hash 函数族 $\boldsymbol{H}=(H_j)_{j\in[k]}$、过滤器的当前状态 $T\in\{0,1\}^\ell$ 和一个元素 $u\in\mathcal{U}$ 为输入，然后计算并输出一个比特 $b=\wedge_{j\in[k]}T[H_j(u)]$。

如上定义的布隆过滤器具有完善的完备性、假阳性概率有界、集合表示紧致等特性。

3.2　双线性映射与 CDH 假设

定义 2　双线性映射。令 $\mathbb{G}$ 和 $\mathbb{G}_T$ 是 2 个阶为 p 的乘法循环群，其中 p 是一个长度为 λ（安全参数）的素数，令 g 为 $\mathbb{G}$ 的随机生成元，则双线性映射 $e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ 满足下述条件的映射。

1) 双线性：对任意的群元素 $x,y\in\mathbb{G}$ 和任意的整数 $a,b\in\mathbb{Z}_p$，$e(x^a,y^b)=e(x,y)^{ab}$ 均成立。

2) 非退化性：$e(g,g)\neq 1$。

3) 可计算性：对任意的群元素 $x,y\in\mathbb{G}$，存在一个多项式时间算法能有效计算出 $e(x,y)$。

为方便描述，用一个五元组 $(\mathbb{G},\mathbb{G}_T,g,p,e)$ 来表示如上定义的双线性映射。进一步，定义在群 $\mathbb{G}$ 上的 CDH 问题是指，给定群元素 $g^a,g^b\in\mathbb{G}$，要求计算 g^{ab}，其中 $a,b\in\mathbb{Z}_p^*$ 是随机选取的整数。

定义 3　CDH 假设。对于任意一个概率多项式时间的攻击者 $\mathcal{A}$，给定 $g^a,g^b\in\mathbb{G}$，若其输出 g^{ab} 的概率 $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CDH}}(\lambda)$ 是 λ 的可忽略函数，则称 CDH 假设在群 $\mathbb{G}$ 上成立。

4　可穿刺的 IBS 方案定义

4.1　形式化定义

IBPS 方案可看作一般的 IBS 方案在安全性方面的增强，也可看作可穿刺签名方案针对无公钥基础设施情形下的扩展。粗略而言，可穿刺的签名方案比一般签名方案多了一个私钥穿刺算法，即当用户的私钥在每签名完一个消息后，需要利用该消息在用户私钥上进行穿刺操作，以防用户私钥被再次用来产生该消息的签名。

通过结合一般的 IBS 方案[3]和可穿刺加密/签名方案[23,26]的形式化定义，下面给出 IBPS 方案的形式化定义，共由 5 个多项式时间的算法组成。

Setup(λ)，系统建立算法。输入系统安全参数 λ，输出系统公开参数 pp 和系统主密钥 msk。

Extract(pp,msk,ID)，私钥提取算法。输入系统公开参数 pp、系统主密钥 msk 和用户身份 ID，输出相应的初始私钥 $\mathrm{sk}_{\mathrm{ID}}$。

Punc$(\mathrm{pp},\mathrm{sk}_{\mathrm{ID}},m)$，私钥穿刺算法。输入系统公开参数 pp、用户的当前私钥 $\mathrm{sk}_{\mathrm{ID}}$ 和一个用该私钥签名过的消息 m，输出一个更新过的私钥 $\mathrm{sk}'_{\mathrm{ID}}$，此时也称 $\mathrm{sk}'_{\mathrm{ID}}$ 是被消息 m 穿刺过的私钥。

Sign$(\mathrm{pp},\mathrm{sk}_{\mathrm{ID}},m)$，签名算法。输入系统公开参数 pp、用户当前私钥 $\mathrm{sk}_{\mathrm{ID}}$ 和待签名消息 m，输出对消息 m 的签名 σ。

Verify$(\mathrm{pp},\sigma,m,\mathrm{ID})$，验证算法。输入系统公开参数 pp、签名 σ、签名消息 m 和用户身份 ID，当 σ 是一个关于 ID 和 m 的有效签名时输出 1，否则输出 0。

一个 IBPS 方案是正确的是指，对任意按正确方式生成的公开参数和主密钥 $(\mathrm{pp},\mathrm{msk})\leftarrow \mathrm{Setup}(\lambda)$、任意用户初始私钥 $\mathrm{sk}_{\mathrm{ID}}\leftarrow \mathrm{Extract}(\mathrm{pp},\mathrm{msk},\mathrm{ID})$ 以及任意签名消息集 $M=\{m_1,\cdots,m_q\}$ 和签名消息 $m\in M$，有

$$\mathrm{Verify}\left(\mathrm{pp},\mathrm{Sign}(\mathrm{pp},\mathrm{sk}_{\mathrm{ID}},m),m,\mathrm{ID}\right)\rightarrow 1$$

总成立。对反复穿刺 q 次后的用户私钥 $\mathrm{sk}'_{\mathrm{ID}}\leftarrow \mathrm{Punc}(\mathrm{pp},\mathrm{sk}_{\mathrm{ID}},m_i)\,(i\in[q])$，只要签名消息 $m\notin M$，则

$$\mathrm{Verify}\left(\mathrm{pp},\mathrm{Sign}(\mathrm{pp},\mathrm{sk}'_{\mathrm{ID}},m),m,\mathrm{ID}\right)\rightarrow 1$$

总成立。

4.2 安全性定义

为定义 IBPS 方案的存在不可伪造性，在安全模型中给攻击者提供了私钥提取询问、签名询问和私钥穿刺询问。此外，为刻画前向安全性，还允许攻击者询问挑战者身份所对应的私钥，但所返回的私钥必须是被挑战消息穿刺过的。具体而言，IBPS 方案的存在不可伪造性通过一个挑战者 $\mathcal{C}$ 和一个敌手 $\mathcal{A}$ 之间的安全游戏来定义，主要包含以下 3 个阶段。

系统建立阶段。给定安全参数 λ，挑战者 $\mathcal{C}$ 运行系统建立算法 $\mathrm{Setup}(\lambda)\rightarrow(\mathrm{pp},\mathrm{msk})$，然后将系统公开参数 pp 发送给敌手 $\mathcal{A}$，而自己持有系统主密钥 msk。同时，挑战者 $\mathcal{C}$ 初始化 4 个空集 $Q_{\mathrm{sk}}\leftarrow\varnothing$、$Q_{\mathrm{sig}}\leftarrow\varnothing$、$Q_{\sigma}\leftarrow\varnothing$ 和 $Q_{\mathrm{punc}}\leftarrow\varnothing$。

询问阶段。在该阶段，敌手 $\mathcal{A}$ 可以自适应地进行以下 3 种询问。

1) 私钥提取询问。敌手 $\mathcal{A}$ 向挑战者 $\mathcal{C}$ 提交一个用户身份 ID，要求挑战者 $\mathcal{C}$ 返回相应的私钥 $\mathrm{sk}_{\mathrm{ID}}$。接收到该询问后，挑战者 $\mathcal{C}$ 直接运行私钥提取算法 $\mathrm{Extract}(\mathrm{pp},\mathrm{msk},\mathrm{ID})\rightarrow\mathrm{sk}_{\mathrm{ID}}$，同时更新集合 $Q_{\mathrm{sk}}\leftarrow Q_{\mathrm{sk}}\cup\{\mathrm{ID}\}$，然后将 $\mathrm{sk}_{\mathrm{ID}}$ 发送给敌手 $\mathcal{A}$。

2) 签名询问。敌手 $\mathcal{A}$ 向挑战者 $\mathcal{C}$ 提交一个用户身份 ID 和消息 m，要求挑战者 $\mathcal{C}$ 返回相应的签名 σ。挑战者 $\mathcal{C}$ 首先调用私钥生成算法 $\mathrm{Extract}(\mathrm{pp},\mathrm{msk},\mathrm{ID})\rightarrow\mathrm{sk}_{\mathrm{ID}}$，然后运行签名算法 $\mathrm{Sign}(\mathrm{pp},\mathrm{sk}_{\mathrm{ID}},m)\rightarrow\sigma$，同时更新集合 $Q_{\mathrm{sig}}\leftarrow Q_{\mathrm{sig}}\cup\{(\mathrm{ID},m)\}$ 和 $Q_{\sigma}\leftarrow Q_{\sigma}\cup\{\sigma\}$，最后将签名 σ 发送给敌手 $\mathcal{A}$。

3) 穿刺询问。敌手 $\mathcal{A}$ 向挑战者 $\mathcal{C}$ 提交一个用户身份 ID 和消息 m，要求挑战者 $\mathcal{C}$ 返回相应被 m 穿刺过的用户私钥 $\mathrm{sk}'_{\mathrm{ID}}$。为此，挑战者首先通过调用私钥生成算法产生用户的初始私钥，即 $\mathrm{Extract}(\mathrm{pp},\mathrm{msk},\mathrm{ID})\rightarrow\mathrm{sk}_{\mathrm{ID}}$，然后调用私钥穿刺算法 $\mathrm{Punc}(\mathrm{pp},\mathrm{sk}_{\mathrm{ID}},m)\rightarrow\mathrm{sk}'_{\mathrm{ID}}$，同时更新集合 $Q_{\mathrm{punc}}\leftarrow Q_{\mathrm{punc}}\cup\{(\mathrm{ID},m)\}$，最后将穿刺过后的私钥 $\mathrm{sk}'_{\mathrm{ID}}$ 发送给敌手 $\mathcal{A}$。

伪造阶段。敌手 $\mathcal{A}$ 结束询问阶段之后，输出一个关于 (m^*,ID^*) 的签名 σ^*。

若以下条件同时满足，则称敌手 $\mathcal{A}$ 赢得了上述安全性游戏。

① $\mathrm{Verify}(\mathrm{pp},\sigma^*,m^*,\mathrm{ID}^*)\rightarrow 1$。

② $(\mathrm{ID}^*,m^*)\notin Q_{\mathrm{sig}}$ 且 $\sigma^*\notin Q_{\sigma}$。

③ $\mathrm{ID}^*\notin Q_{\mathrm{sk}}$。

④ 对于任意 $m\neq m^*$，都有 $(\mathrm{ID}^*,m)\notin Q_{\mathrm{punc}}$。

定义 4 IBPS 的存在不可伪造性。对任意一个针对 IBPS 方案的概率多项式时间敌手 $\mathcal{A}$，若其赢得上述安全游戏的概率 $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{IBPS}}(\lambda)$ 是 λ 的一个可忽略函数，则称该 IBPS 方案满足存在不可伪造性。

本文考虑一种较弱的存在不可伪造性，即选择性存在不可伪造性，要求敌手 $\mathcal{A}$ 在开始安全性游戏之前就将挑战消息 m^* 提交给挑战者 $\mathcal{C}$。

5 方案具体构造

基于 Paterson 等[3]的 IBS 方案，本节给出一个具体的 IBPS 方案构造，其核心在于如何实现用户密钥的穿刺操作。为此，本文借鉴 Derler 等[23]提出的密钥穿刺机制，利用布隆过滤器来管理用户私钥和签名消息。具体而言，在生成用户私钥时，先初始化一个布隆过滤器 $(\boldsymbol{H},T)\leftarrow\mathrm{BFGen}(\ell,k)$，然后对于 T 的每一个位置 $i\in[\ell]$，生成一个相应的私钥构件 sk_i；在对消息 m 进行签名时，首先利用 $\boldsymbol{H}$ 将其映射到 T 的 k 个位置，从中选取一个满足 $T[i']\neq 1$ 的位置 i'，然后利用相应的用户私钥构件 $\mathrm{sk}_{i'}$ 进行签名；在完成对 m 的签名之后，将 m 添加到布隆过滤器中并更新其状态 $\mathrm{BFUpdate}(\boldsymbol{H},T,m)\rightarrow T'$，同时，在所有满足 $T'[i]=1$ 的位置，将相应的用户私钥构件设置为 $\mathrm{sk}_i\leftarrow\perp$。可以看出，通过上述密钥穿刺过程，用户私钥不能对已签名过的消息进行再次签名。具体构造的 IBPS 方案由以下 5 个算法组成。

$\mathrm{Setup}(\lambda)$。给定系统安全参数 λ 之后，系统建立算法首先生成双线性群 $(\mathbb{G},\mathbb{G}_T,g,p,e)$；然后，分别定义用户身份标识空间 $\mathrm{ID}=\{0,1\}^{\gamma}$ 和签名消息空间 $\mathcal{M}=\{0,1\}^{\eta}$，选择 2 个 Hash 函数

$H:\{0,1\}^*\to\{0,1\}^\gamma$ 和 $H':\{0,1\}^*\to\{0,1\}^\eta$，以及另外一个 Hash 函数 $\tilde{H}:\mathbb{N}\to\mathbb{G}$；选择随机群元素 $g_2,u_0,v_o\in\mathbb{G}$，以及 2 个随机向量 $\boldsymbol{u}=(u_1,\cdots,u_\gamma)\in\mathbb{G}^\gamma$ 和 $\boldsymbol{v}=(v_1,\cdots,v_\eta)\in\mathbb{G}^\eta$；最后，选择一个随机整数 $\alpha\in\mathbb{Z}_p$，计算 $g_1=g^\alpha$，并令系统主密钥为 $\text{msk}=g_2^\alpha$，系统公开参数为 $\text{pp}=\{(\mathbb{G},\mathbb{G}_T,g,p,e),H,H',\tilde{H},\boldsymbol{u},\boldsymbol{v},u_0,v_0,g_1,g_2\}$。

Extract(pp,msk,ID)。给定系统公开参数 pp、系统主密钥 msk 和用户身份 $\text{ID}\in\{0,1\}^*$，私钥提取算法首先将 ID 映射到系统定义的身份标识空间，即令 $\boldsymbol{\theta}=H(\text{ID})=(\theta_1,\cdots,\theta_\gamma)\in\{0,1\}^\gamma$；然后，初始化一个布隆过滤器 $(\boldsymbol{H},T)\leftarrow\text{BFGen}(\ell,k)$，其中 $T=0^\ell$；选择随机整数 $r_1,r_2\in\mathbb{Z}_p$，对任意整数 $i\in[\ell]$，按如下方式计算私钥构件 sk_i。

$$\text{sk}_i=g_2^\alpha\left(u_0\prod_{j=1}^{\gamma}u_j^{\theta_j}\right)^{r_1}\tilde{H}(i)^{r_2} \tag{1}$$

最后，按照如下结构输出用户私钥。

$$\text{sk}_{\text{ID}}=\left\{(\boldsymbol{H},T),\{\text{sk}_i\}_{i\in[\ell]},k_1=g^{r_1},k_2=g^{r_2}\right\} \tag{2}$$

Punc(pp,sk_{ID},m)。给定系统公开参数 pp、用户的当前私钥 $\text{sk}_{\text{ID}}=\left\{(\boldsymbol{H},T),\{\text{sk}_i\}_{i\in[\ell]},k_1,k_2\right\}$ 和一个消息 $m\in\{0,1\}^*$，令 $\boldsymbol{\omega}=H'(m)=(\omega_1,\cdots,\omega_\eta)\in\{0,1\}^\eta$；然后，将消息 m 添加到布隆过滤器的集合中，并对其状态进行更新，即令 $T'\leftarrow\text{BFUpdate}(\boldsymbol{H},T,m)$；对任意 $i\in[\ell]$，按照如下方式对相应的私钥构件进行更新。

$$\text{sk}_i'=\begin{cases}\text{sk}_i, & T'[i]=0\\ \perp, & T'[i]\neq 0\end{cases} \tag{3}$$

最后，输出更新私钥 $\text{sk}'_{\text{ID}}=\{(\boldsymbol{H},T'),\{\text{sk}'_i\}_{i\in[\ell]},k_1,k_2\}$。

Sign(pp,sk_{ID},m)。给定公开参数 pp、用户当前私钥 $\text{sk}_{\text{ID}}=\left\{(\boldsymbol{H},T),\{\text{sk}_i\}_{i\in[\ell]},k_1,k_2\right\}$ 和待签名消息 m，若 $\text{BFCheck}(\boldsymbol{H},T,m)\to 1$，则输出错误符号 $\perp$，否则，存在一个指标 $i^*\in\{i_1,\cdots,i_k\}$ 使 $\text{sk}_{i^*}\neq\perp$，其中 $i_j=H_j(m)$ $(j\in[k])$；令 $\boldsymbol{\omega}=H'(m)=(\omega_1,\cdots,\omega_\eta)$，选取随机指数 $r_m\in\mathbb{Z}_p$，计算

$$\sigma_0=\text{sk}_{i^*}\left(v_0\prod_{j=1}^{\eta}v_j^{\omega_j}\right)^{r_m},\ \sigma_1=k_1,\ \sigma_2=k_2,\ \sigma_3=g^{r_m} \tag{4}$$

最后，返回签名 $\sigma=\{i^*,\sigma_o,\sigma_1,\sigma_2,\sigma_3\}$。

Verify(pp,σ,m,ID)。给定公开参数 pp、签名 $\sigma=\{i^*,\sigma_o,\sigma_1,\sigma_2,\sigma_3\}$、消息 m 和用户身份 ID，首先令 $\boldsymbol{\theta}=H(\text{ID})=(\theta_1,\cdots,\theta_\gamma)$ 和 $\boldsymbol{\omega}=H'(m)=(\omega_1,\cdots,\omega_\eta)$；然后，验证下述等式是否成立。

$$\begin{aligned}e(\sigma_0,g)=e(g_1,g_2)e\left(u_0\prod_{j=1}^{\gamma}u_j^{\theta_j},\sigma_1\right)\cdot\\ e(\tilde{H}(i^*),\sigma_2)e\left(v_0\prod_{j=1}^{\eta}v_j^{\omega_j},\sigma_3\right)\end{aligned} \tag{5}$$

若等式成立，则说明 σ 是一个关于 ID 和 m 的有效签名，输出 1；否则输出 0。

方案正确性　依据私钥生成算法可知，私钥构件 sk_{i^*} 的值为 $\text{sk}_{i^*}=g_2^\alpha\left(u_0\prod_{j=1}^{\gamma}u_j^{\theta_j}\right)^{r_1}\tilde{H}(i^*)^{r_2}$，将其代入式(5)可得

$$\begin{aligned}&e(\sigma_0,g)=e\left(g_2^\alpha\left(u_0\prod_{j=1}^{\gamma}u_j^{\theta_j}\right)^{r_1}\tilde{H}(i^*)^{r_2}\left(v_0\prod_{j=1}^{\eta}v_j^{\omega_j}\right)^{r_m},g\right)=\\ &e(g_2^\alpha,g)e\left(\left(u_0\prod_{j=1}^{\gamma}u_j^{\theta_j}\right)^{r_1},g\right)e\left(\tilde{H}(i^*)^{r_2},g\right)\cdot\\ &e\left(\left(v_0\prod_{j=1}^{\eta}v_j^{\omega_j}\right)^{r_m},g\right)=e(g_1,g_2)e\left(u_0\prod_{j=1}^{\gamma}u_j^{\theta_j},k_1\right)\cdot\\ &e\left(\tilde{H}(i^*),k_2\right)e\left(v_0\prod_{j=1}^{\eta}v_j^{\omega_j},g^{r_m}\right)=\\ &e(g_1,g_2)e\left(u_0\prod_{j=1}^{\gamma}u_j^{\theta_j},\sigma_1\right)\cdot\\ &e\left(\tilde{H}(i^*),\sigma_2\right)e\left(v_0\prod_{j=1}^{\eta}v_j^{\omega_j},\sigma_3\right)\end{aligned} \tag{6}$$

6　安全性证明及性能分析

6.1　安全性证明

定理 1　IBPS 方案安全性。对于依据安全参数 λ 生成的双线性群 $(\mathbb{G},\mathbb{G}_T,g,p,e)$，若 CDH 假设在群 $\mathbb{G}$ 上成立，则所构造的 IBPS 方案在随机预言模型下满足选择性的存在不可伪造性。具体地，有

$$\text{Adv}_{\mathcal{C}}^{\text{CDH}}(\lambda)\geqslant\frac{\text{Adv}_{\mathcal{A}}^{\text{IBPS}}(\lambda)}{16q_{\text{punc}}(q_{\text{sk}}+q_{\text{sig}})(\gamma+1)(\eta+1)} \tag{7}$$

其中，q_{punc} 是私钥穿刺询问次数，q_{sk} 是私钥提取询

问次数，q_{sig} 是签名询问次数。

证明 为证明定理 1，主要采用 Waters[16]提出的证明技巧，同时以随机预言机的方式模拟 Hash 函数 $\tilde{H}$，对其进行扩展。下面，给出挑战者 $\mathcal{C}$ 模拟安全性游戏各个阶段的细节。

系统建立阶段 在初始阶段，挑战者 $\mathcal{C}$ 收到一个双线性群 $(\mathbb{G},\mathbb{G}_T,g,p,e)$，以及群 $\mathbb{G}$ 上 CDH 问题的实例 (g^a,g^b)，其中 a 和 b 是选自 $\mathbb{Z}_p^*$ 中的随机整数，$\mathcal{C}$ 的目标是计算出 g^{ab}。此外，敌手 $\mathcal{A}$ 需在安全性游戏开始之前提交挑战消息 m^*。

挑战者 $\mathcal{C}$ 首先令 $l_u=2(q_e+q_s)$ 和 $l_v=2q_s$，选择 2 个随机整数 $k_u\in\{0,\cdots,\gamma\}$ 和 $k_v\in\{0,\cdots,\eta\}$，并假设对于给定的 q_e、q_s、γ 和 η，$l_u(\gamma+1)<p$ 和 $l_v(\eta+1)<p$ 分别成立；然后，$\mathcal{C}$ 选择一个随机整数 $x'\in\mathbb{Z}_{l_u}$ 和一个随机向量 $\boldsymbol{x}=(x_1,\cdots,x_\gamma)\in\mathbb{Z}_{l_u}^\gamma$，以及另外一个随机整数 $z'\in\mathbb{Z}_{l_v}$ 和一个随机向量 $\boldsymbol{z}=(z_1,\cdots,z_\eta)\in\mathbb{Z}_{l_v}^\eta$；挑战者进一步选择 2 个随机整数 $y'\in\mathbb{Z}_p$ 和 $w'\in\mathbb{Z}_p$、2 个随机向量 $\boldsymbol{y}=(y_1,\cdots,y_\gamma)\in\mathbb{Z}_p^\gamma$ 和 $\boldsymbol{w}=(w_1,\cdots,w_\eta)\in\mathbb{Z}_p^\eta$。为方便描述，将用户身份 ID 和签名消息 m 进行压缩映射后定义为如下函数

$$F(\text{ID})=x'+\sum_{j=1}^{\gamma}\theta_i x_i-l_u k_u,\quad J(\text{ID})=y'+\sum_{j=1}^{\gamma}\theta_j y_j;$$

$$K(m)=z'+\sum_{j=1}^{n}\omega_j z_j-l_v k_v,\quad L(m)=w'+\sum_{j=1}^{n}\omega_j w_j$$

在上述定义的基础上，按如下方式设置所构造 IBPS 方案的系统公开参数

$$g_1=g^a,g_2=g^b,\ u_0=g_2^{-l_u k_u+x'}g^{y'},u_i=g_2^{x_i}g^{y_i},i\in[\gamma];$$
$$v_0=g_2^{-l_v k_v+z'}g^{w'},\ m_j=g_2^{z_j}g^{w_j},j\in[\eta]$$

上述赋值方式意味着系统主密钥被隐式地设置为 $\text{msk}=g_2^\alpha=g_2^a=g^{ab}$。此外，对于任意经过压缩后的用户身份标识 ID 和签名消息 m，有

$$u_0\prod_{i=1}^{\gamma}u_i^{\theta_i}=g_2^{F(\text{ID})}g^{J(\text{ID})},\quad v_0\prod_{j=1}^{\eta}v_j^{\omega_j}=g_2^{K(m)}g^{L(m)}$$

成立。

最后，$\mathcal{C}$ 初始化 4 个空集 $Q_{\text{sk}}\leftarrow\varnothing$、$Q_{\text{sig}}\leftarrow\varnothing$、$Q_\sigma\leftarrow\varnothing$ 和 $Q_{\text{punc}}\leftarrow\varnothing$，并将按照上述方式生成的系统公开参数 $\text{pp}=\{(\mathbb{G},\mathbb{G}_T,g,p,e),\boldsymbol{u},\boldsymbol{v},u_0,v_0,g_1,g_2\}$ 发送给攻击者 $\mathcal{A}$，同时还向其提供 Hash 函数的查询服务。

询问阶段 在该阶段，$\mathcal{A}$ 可以自适应地进行私钥提取查询、签名查询和私钥穿刺查询，这 3 种询问同时还涉及 Hash 函数 $H,H',\tilde{H}$ 的询问。对于 H 和 H'，挑战者将它们模拟成一般的抗碰撞 Hash 函数。对于 Hash 函数 $\tilde{H}$，挑战者 $\mathcal{C}$ 按照如下方式进行模拟：对任意 $i\in[\ell]$，选择一个随机整数 $t_i\in\mathbb{Z}_p$，若有 $i\in\{H_j(m^*):j\in[k]\}$，则直接令 $\tilde{H}(i)=g^{t_i}$，否则令 $\tilde{H}(i)=g_2g^{t_i}$。下面给出 $\mathcal{C}$ 响应 3 种查询的方式。

1) 私钥提取询问。给定一个用户身份 ID，$\mathcal{C}$ 先令 $H(\text{ID})=(\theta_1,\cdots,\theta_\gamma)$，此时 H 被模拟成一个一般的抗碰撞 Hash 函数。进一步，若有 $F(\text{ID})=0 \bmod p$，则 $\mathcal{C}$ 终止模拟，否则 $\mathcal{C}$ 选择随机整数 $r_1',r_2\in\mathbb{Z}_p$，令 $r_1=r_1'-a/F(\text{ID})$。然后，$\mathcal{C}$ 计算 ID 所对应的私钥构件 sk_i

$$\begin{aligned}\text{sk}_i&=g_2^\alpha\left(u_0\prod_{j=1}^{\gamma}u_j^{\theta_j}\right)^{r_1}\tilde{H}(i)^{r_2}=\\&g^{ab}(g_2^{F(\text{ID})}g^{J(\text{ID})})^{r_1'-\frac{a}{F}(\text{ID})}\tilde{H}(i)^{r_2}=\\&g^{ab}g_2^{r_1'F(\text{ID})}g_2^{-a}g^{r_1'J(\text{ID})}g^{-a\frac{J(\text{ID})}{F(\text{ID})}}\tilde{H}(i)^{r_2}=\\&g_2^{r_1'F(\text{ID})}g^{r_1'J(\text{ID})}(g^a)^{-\frac{J(\text{ID})}{F(\text{ID})}}H(i)^{r_2}\end{aligned}\tag{8}$$

以及 $k_1=g^{r_1}=g^{r_1'}(g^a)^{-\frac{1}{F(\text{ID})}}$ 和 $k_2=g^{r_2}$。

从式(8)中可以看出，尽管 $\mathcal{C}$ 不知道系统主密钥，但上述私钥构件对其都是可计算的，并且与按照真实的私钥提取算法所产生的私钥是同分布的。最后，$\mathcal{C}$ 将 ID 的私钥 $\text{sk}_{\text{ID}}=\{(\boldsymbol{H},T),\{\text{sk}_i\}_{i\in[\ell]},k_1,k_2\}$ 发送给敌手 $\mathcal{A}$，同时更新集合 $Q_{\text{sk}}\leftarrow Q_{\text{sk}}\cup\{\text{ID}\}$。

2) 签名询问。给定用户身份 ID 和待签名消息 m，挑战者 $\mathcal{C}$ 首先将它们映射到相应的二进制字符串，即 $H(\text{ID})=(\theta_1,\cdots,\theta_\gamma)$、$H'(m)=(\omega_1,\cdots,\omega_\eta)$。此时若有 $F(\text{ID})\neq 0 \bmod p$，则 $\mathcal{C}$ 通过调用私钥提取询问先生成相应的用户私钥 sk_{ID}，然后直接利用 IBPS 方案的签名算法即可生成关于 ID 和 m 的正确签名，并返回给敌手 $\mathcal{A}$。另一方面，若 $F(\text{ID})=0 \bmod p$ 且 $K(m)=0 \bmod p$，则 $\mathcal{C}$ 终止安全性游戏，否则 $\mathcal{C}$ 选择随机整数 $r_1,r_2,r_m'\in\mathbb{Z}_p$，并令 $r_m=r_m'-a/K(m)$，然后按照下述方式生成签名

$$\sigma_0=g_2^\alpha\left(u_0\prod_{j=1}^{\gamma}u_j^{\theta_j}\right)^{r_1}\tilde{H}(i^*)^{r_2}\left(v_0\prod_{j=1}^{\eta}v_j^{\omega_j}\right)^{r_m}=$$

$$g^{ab}(g_2^{F(\mathrm{ID})}g^{J(\mathrm{ID})})^{r_1}\tilde{H}(i^*)^{r_2}(g_2^{K(m)}g^{L(m)})^{r'_m-\frac{a}{K(m)}}=$$
$$g^{ab}g^{r_1J(\mathrm{ID})}\tilde{H}(i)^{r_2}g_2^{r'_mK(m)}g^{-ab}g^{r'_mL(m)}g^{-\frac{aL(m)}{K(m)}}=$$
$$g^{r_1J(\mathrm{ID})}\tilde{H}(i)^{r_2}(g^b)^{r'_mK(m)}g^{r'_mL(m)}(g^a)^{\frac{aL(m)}{K(m)}}$$

$$\sigma_1=g^{r_1},\ \sigma_2=g^{r_2},\ \sigma_3=g^{r_m}=g^{r'_m}g^{\frac{a}{K(m)}} \tag{9}$$

可以看出，上述签名构件对 $\mathcal{C}$ 都是可计算的，且与 IBPS 方案中的签名算法所产生的签名同分布。最后，$\mathcal{C}$ 将关于 ID 和 m 的签名 $\sigma=\{i^*,\sigma_o,\sigma_1,\sigma_2,\sigma_3\}$ 返回给敌手 $\mathcal{A}$，同时更新集合 $Q_{\mathrm{sig}}\leftarrow Q_{\mathrm{sig}}\cup\{(\mathrm{ID},m)\}$ 和 $Q_\sigma\leftarrow Q_\sigma\cup\{\sigma\}$。

3) 私钥穿刺询问。在响应该询问之前，挑战者 $\mathcal{C}$ 先猜测敌手 $\mathcal{A}$ 的第 j 次该询问是关于挑战身份 ID^* 的查询，若猜测失败，则终止安全性游戏。对于关于其他用户身份的私钥穿刺询问，$\mathcal{C}$ 先调用私钥提取询问得到相应的用户私钥，然后利用 IBPS 方案的私钥穿刺算法即可得到穿刺后的私钥，并将其返回给敌手 $\mathcal{A}$。特别地，对于 $\mathcal{A}$ 的第 j 次询问，由安全模型中对敌手的约束可知，此时 $\mathcal{A}$ 提供的就是挑战身份 ID^* 和挑战消息 m^*。为生成穿刺后的用户私钥，$\mathcal{C}$ 选择随机整数 $r_1,r'_2\in\mathbb{Z}_p$，令 $r_2=r'_2-a$，对任意 $i\in[\ell]$，若 $i\in\{H_j(m^*):j\in[k]\}$，则令 $\mathrm{sk}_i\leftarrow\perp$，否则按照下述方式计算 sk_i

$$\mathrm{sk}_i=g_2^{\alpha}\left(u_0\prod_{j=1}^{\gamma}u_j^{\theta_j}\right)^{r_1}\tilde{H}(i)^{r_2}=$$
$$g^{ab}(g_2^{F(\mathrm{ID})}g^{J(\mathrm{ID})})^{r_1}\tilde{H}(i)^{r'_2}(g_2g^{t_i})^{-a}=$$
$$(g_2^{F(\mathrm{ID})}g^{J(\mathrm{ID})})^{r_1}\tilde{H}(i)^{r'_2}g^{-at_i} \tag{10}$$

以及 $k_1=g^{r_1}$ 和 $k_2=g^{r_2}=g^{r'_2}g^{-a}$。可以看出，上述私钥值对 $\mathcal{C}$ 都是可计算的。最后，$\mathcal{C}$ 将生成的穿刺私钥返回给 $\mathcal{A}$，并更新集合 $Q_{\mathrm{punc}}\leftarrow Q_{\mathrm{punc}}\cup\{(\mathrm{ID},m)\}$。

伪造阶段　敌手 $\mathcal{A}$ 结束询问阶段之后，输出一个关于 (m^*,ID^*) 的签名 $\sigma^*=\{i^*,\sigma_0^*,\sigma_1^*,\sigma_2^*,\sigma_3^*\}$，同时满足下述条件。

① $\mathrm{Verify}(\mathrm{pp},\sigma^*,m^*,\mathrm{ID}^*)\rightarrow 1$。

② $(\mathrm{ID}^*,m^*)\notin Q_{\mathrm{sig}}$ 且 $\sigma^*\notin Q_\sigma$。

③ $\mathrm{ID}^*\notin Q_{\mathrm{sk}}$。

④ 对于任意 $m\neq m^*$，都有 $(\mathrm{ID}^*,m)\notin Q_{\mathrm{punc}}$。

此时，若 $F(\mathrm{ID}^*)\neq 0 \bmod p$ 或 $K(m^*)\neq 0 \bmod p$，则挑战者 $\mathcal{C}$ 直接终止安全性游戏。最终，若挑战者 $\mathcal{C}$ 没有终止安全性优势，则可按照下述方式计算出 g^{ab}。

$$\frac{\sigma_0}{\sigma_1^{J(\mathrm{ID}^*)}\sigma_2^{L(m^*)}\sigma_3^{t_i^*}}=$$
$$\frac{g_2^{\alpha}\left(u_0\prod_{j=1}^{\gamma}u_j^{\theta_j^*}\right)^{r_1}\tilde{H}(i^*)^{r_2}\left(v_0\prod_{j=1}^{\eta}v_j^{\omega_j^*}\right)^{r_{m^*}}}{g^{r_1J(\mathrm{ID}^*)}g^{r_2t_i^*}g^{r_{m^*}L(m^*)}}=$$
$$\frac{g^{ab}(g_2^{F(\mathrm{ID}^*)}g^{J(\mathrm{ID}^*)})^{r_1}g^{r_2t_i^*}(g_2^{K(m^*)}g^{L(m^*)})^{r_{m^*}}}{g^{r_1J(\mathrm{ID}^*)}g^{r_2t_i^*}g^{r_{m^*}L(m^*)}}=g^{ab} \tag{11}$$

从而解决了给定的 CDH 问题实例。

概率分析　挑战者 $\mathcal{C}$ 能计算出 g^{ab} 的前提是不终止安全性游戏，而由 Waters[16]的“artificial abort”分析技术可得，$\mathcal{C}$ 不终止安全性游戏的概率为

$$\frac{1}{16q_{\mathrm{punc}}(q_{\mathrm{sk}}+q_{\mathrm{sig}})(\gamma+1)(\eta+1)}$$

从而有

$$\mathrm{Adv}_{\mathcal{C}}^{\mathrm{CDH}}(\lambda)\geqslant\frac{\mathrm{Adv}_{\mathcal{A}}^{\mathrm{IBPS}}(\lambda)}{16q_{\mathrm{punc}}(q_{\mathrm{sk}}+q_{\mathrm{sig}})(\gamma+1)(\eta+1)} \tag{12}$$

6.2　性能分析

表 1 给出了本文所提 IBPS 方案与其他相关签名方案在计算开销和存储开销方面的比较。在比较过程中，用 e 表示一次模指数运算，p 表示一次双线性对运算，γ 表示用户身份的长度，η 表示签名消息的长度，ℓ 表示布隆过滤器的数组长度，N 表示时间周期总数，$\mathbb{G}$ 表示一个群元素。在比较过程

表 1　相关签名方案的计算/存储开销比较

签名方案	计算开销				存储开销		
	Extract	Sign	Punc/Update	Verify	pp	sk	σ
Paterson 等[3]方案	2e	2e	N/A	3p	$(\gamma+\eta+5)\mathbb{G}$	$2\mathbb{G}$	$3\mathbb{G}$
Yu 等[14]方案	$\mathcal{O}(\log N)$e	2e	$\mathcal{O}(\log N)$e	5p	$\mathcal{O}(\log N)\mathbb{G}$	$\mathcal{O}(\log^2 N)\mathbb{G}$	$4\mathbb{G}+\mathbb{Z}_p$
本文所提方案	$(2\ell+2)$e	2e	0	5p	$(\gamma+\eta+5)\mathbb{G}$	$(2\ell+2)\mathbb{G}$	$4\mathbb{G}+\mathbb{Z}_p$

中忽略运算时间非常小的 Hash 运算。

从表 1 中可以看出，相比于 Paterson 等[3]的 IBS 方案，本文所提 IBPS 方案在私钥提取算法上具有较大的计算开销，这主要是由于为了实现细粒度的前向安全性，额外地为布隆过滤器所对应的二进制序列的每一个位置上都生成了一个私钥构件。另一方面，相比于 Yu 等[14]的前向安全 IBS 方案，本文所提 IBPS 方案的密钥穿刺/更新开销主要是 Hash 运算和删除运算，几乎可以忽略，而 Yu 等[14]的前向安全 IBS 方案的密钥更新开销与总的时间周期数目成对数关系。此外，在存储开销方面，本文所提 IBPS 方案的公开参数规模与 Paterson 等[3]的 IBS 方案相当，而 Yu 等[14]的前向安全 IBS 方案具有更大的公开参数规模。在签名长度方面，本文所提方案与 Yu 等[14]的前向安全 IBS 方案相同，均略高于 Paterson 等[3]的 IBS 方案。特别地，Yu 等[14]的前向安全 IBS 方案的私钥规模与总的时间周期总数的对数成平方关系，具有较高的存储开销。

表 2 给出了相关签名方案在安全性质方面的比较。从表 2 可以看出，只有本文所提方案具有细粒度的前向安全性，但其安全性却是在随机预言模型下证明的，而非标准模型。此外，在安全性假设方面，Yu 等[14]方案的安全性基于一个更强的 q-型 Diffie-Hellman 指数困难性问题假设。

表 2　　相关签名方案的安全性质比较

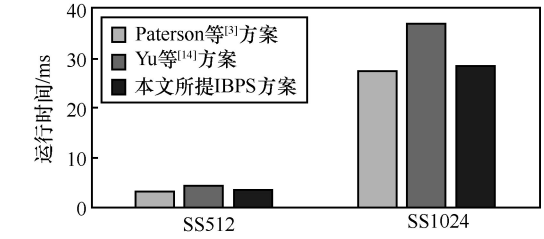
签名方案	标准模型	细粒度前向安全性	安全性假设
Paterson 等[3]方案	✓	×	CDH
Yu 等[14]方案	✓	×	q-DHE
本文所提方案	×	✓	CDH

6.3 仿真实验

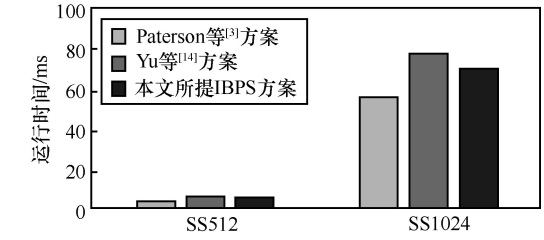
为验证所构造 IBPS 方案的实际性能，将其在 Charm[28]框架下进行了实现，并测试了各个算法的实际运行时间。所有的测试实验均是在具有 16 GB 内存和 i7-1160G7@1.20GHz CPU 的 PC 上进行的。

在仿真实验中，布隆过滤器中的集合规模设置为 $n=2^{20}$，即允许用户在一年中每天向其中添加 2^{12} 个元素，同时布隆过滤器的假阳性概率设置为 $\delta=10^{-3}$，表示布隆过滤器的字符串长度为 $\ell=1.7\times10^{7}$，Hash 函数个数为 $k=10$。在一般的前向安全 IBS 方案中，设定总的时间周期数为 $N=2^{19}$，即允许用户在一年的时间里每隔一分钟更新一次私钥。所采用的 Hash 函数均为 SHA-256，椭圆曲线为 Charm 提供的 SS512 曲线和 SS1024 曲线。

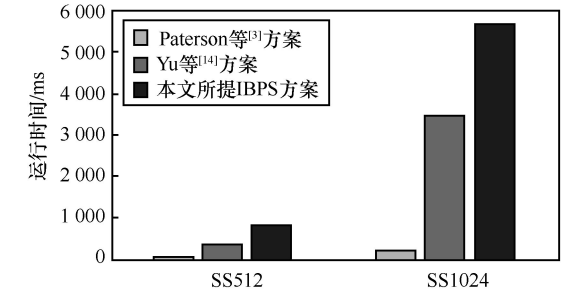
图 1 给出了相关基于身份签名方案中各个算法运行时间对比。从图 1 可以看出，本文所提 IBPS 方案的私钥抽取算法的运行时间要远高于其他 2 个方案，但该算法一般都是离线运行且执行一次，故在实际中不影响系统性能。另一方面，本文所提 IBPS 方案的私钥更新效率非常高。具体地，在 SS512 曲线下，私钥穿刺算法的运行时间仅有 2.32 s，而 Yu 等[14]方案中算法的运行时间则接近 350 s，比本文所提 IBPS 方案高了近 150 倍。



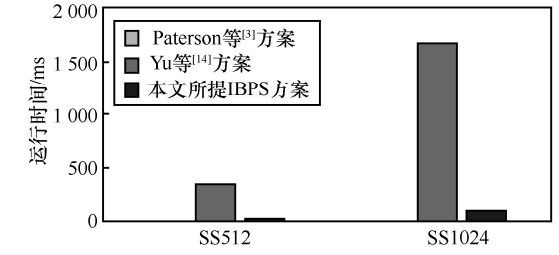
(a) 签名算法



(b) 验证算法



(c) 私钥抽取算法



(d) 私钥穿刺/更新算法

图 1　相关基于身份签名方案中各个算法运行时间对比

图 2 给出了相关 IBS 方案的存储开销。从图 2 可以看出，由于产生了私钥构件的多个备份，本文所提 IBPS 方案具有较大的私钥规模。

6.4　方案应用

本文所提 IBPS 方案可以在任何部署过 IBS 的场景中使用，以增强用户私钥的前向安全性。下面以 IBPS 在监控系统的应用为例，如图 3 所示。目前的物联网技术和无线通信技术支持部署大规模的监控系统，物联网设备、智能手机、无线摄像头、无线相机等向云服务器提供视频和图像，即向监控系统提供监控数据。一个安全可靠的监控系统，应该做到设备提供的视频和图像确保是真实的，不是被修改或伪造的；设备需要诸如公钥证书等凭证来提供视频和图像的真实性；任何攻击或私钥泄露不会破坏所有以前的监控数据，即应当保证监控数据的前向安全。

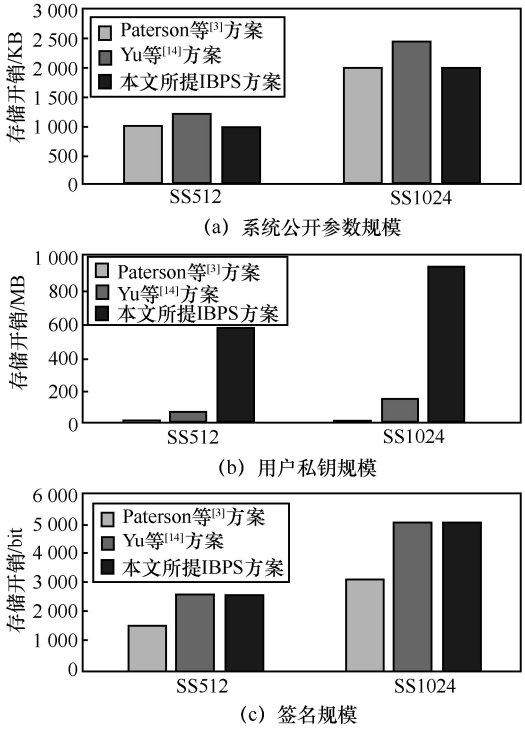


图 2　相关 IBS 方案的存储开销

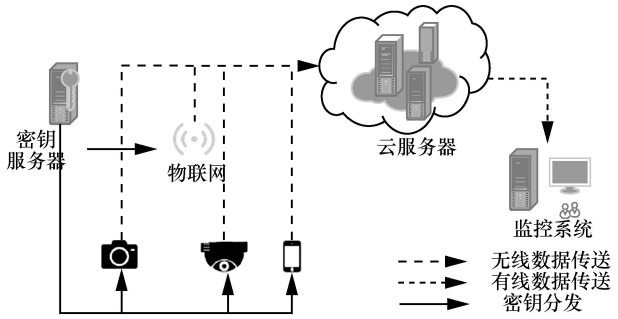


图 3　IBPS 在监控系统的应用

本文所提 IBPS 方案为解决上述实际问题提供了可靠对策。图 3 中的每个设备将其身份标识注册到云服务器，密钥服务器负责为每个设备分发与其身份标识对应的私钥，同时，把自己的公钥发送给云服务器。当设备发送数据时，先对数据签名，然后把签名和数据一起发送，而云服务器只接受其签名有效的数据。这样，一方面保证了数据来源的真实性，另一方面云服务器利用密钥服务器的公钥验证接收的数据，而设备不需要携带证书。每发送完一次数据，该设备都进行密钥穿刺操作，完成私钥的更新。这保证了任何设备在私钥泄露的情况下，该设备所有以前的签名仍然有效。

7　结束语

本文针对已有的前向安全的基于身份签名方案的灵活性不强、私钥更新效率较低的问题，通过引入可穿刺公钥加密思想，提出了基于身份的可穿刺签名方案。基于 Paterson 等[3]的基于身份签名方案，利用布隆过滤器构造了一个具体的基于身份的可穿刺签名方案，以较高的存储开销为代价，实现了签名的细粒度前向安全性和用户私钥的高效更新。安全性证明表明，本文所提 IBPS 方案在随机预言模型下满足存在不可伪造性。理论分析和实验结果表明，本文所提 IBPS 方案在安全性和效率方面具有优势，更适合在实际应用中部署。

参考文献：

[1] SHAMIR A. Identity-based cryptosystems and signature schemes[C]// Advances in Cryptology. Berlin: Springer, 1984: 47-53.

[2] CHA J C. An identity-based signature from gap Diffie-Hellman groups[C]//Public Key Cryptography — PKC 2003. Berlin: Springer, 2003: 18-30.

[3] PATERSON K G, SCHULDT J C N. Efficient identity-based signatures secure in the standard model[C]//Information Security and Privacy. Berlin: Springer, 2006: 207-222.

[4] 杨小东, 杨苗苗, 高国娟, 等. 强不可伪造的基于身份服务器辅助验证签名方案[J]. 通信学报, 2016, 37(6): 49-55.
YANG X D, YANG M M, GAO G J, et al. ID-based server-aided verification signature scheme with strong unforgeability[J]. Journal on Communications, 2016, 37(6): 49-55.

[5] 刘翔宇, 刘胜利, 谷大武. 紧致安全的基于身份的签名方案[J]. 密码学报, 2021, 8(1): 132-141.
LIU X Y, LIU S L, GU D W. Tightly secure identity-based signature scheme[J]. Journal of Cryptologic Research, 2021, 8(1): 132-141.

[6] 田苗苗, 陈静, 仲红. 格上基于身份的增量签名方案[J]. 通信学报, 2021, 42(1): 108-117.
TIAN M M, CHEN J, ZHONG H. Identity-based incremental signa-

ture scheme from lattices[J]. Journal on Communications, 2021, 42(1): 108-117.

[7] 侯红霞, 张明瑞, 赵艳琦, 等. 素数阶群上基于非对称对的身份基环签名[J]. 通信学报, 2021, 42(9): 155-164.
HOU H X, ZHANG M R, ZHAO Y Q, et al. ID-based ring signature on prime order group from asymmetric pairing[J]. Journal on Communications, 2021, 42(9): 155-164.

[8] ANDERSON R. Two remarks on public key cryptology[C]//Invited Lecture at the 4th ACM Conference on Computer and Communications Security. New York: ACM Press, 1997: 1-5.

[9] BELLARE M, MINER S K. A forward-secure digital signature scheme[C]//Advances in Cryptology — CRYPTO' 99. Berlin: Springer, 1999: 431-448.

[10] ITKIS G, REYZIN L. Forward-secure signatures with optimal signing and verifying[C]//Advances in Cryptology — CRYPTO 2001. Berlin: Springer, 2001: 332-354.

[11] KOZLOV A, REYZIN L. Forward-secure signatures with fast key update[C]//Security in Communication Networks. Berlin: Springer, 2003: 241-256.

[12] LIBERT B, QUISQUATER J J, YUNG M. Forward-secure signatures in untrusted update environments: efficient and generic constructions[C]//Proceedings of the 14th ACM Conference on Computer and Communications Security. New York: ACM Press, 2007: 266-275.

[13] ABDALLA M, REYZIN L. A new forward-secure digital signature scheme[C]//Advances in Cryptology-ASIACRYPT 2000. Berlin: Springer, 2000: 116-129.

[14] YU J, HAO R, KONG F Y, et al. Forward-secure identity-based signature: security notions and construction[J]. Information Sciences, 2011, 181(3): 648-660.

[15] GREEN M D, MIERS I. Forward secure asynchronous messaging from puncturable encryption[C]//Proceedings of 2015 IEEE Symposium on Security and Privacy. Piscataway: IEEE Press, 2015: 305-320.

[16] WATERS B. Efficient identity-based encryption without random oracles[C]//Lecture Notes in Computer Science. Berlin: Springer, 2005: 114-127.

[17] 魏江宏, 刘文芬, 胡学先. 前向安全的密文策略基于属性加密方案[J]. 通信学报, 2014, 35(7): 38-45.
WEI J H, LIU W F, HU X X. Forward-secure ciphertext-policy attribute-based encryption scheme[J]. Journal on Communications, 2014, 35(7): 38-45.

[18] WEI J H, LIU W F, HU X X. Forward-secure identity-based signature with efficient revocation[J]. International Journal of Computer Mathematics, 2017, 94(7): 1390-1411.

[19] OH H, KIM J, SHIN J S. Forward-secure ID based digital signature scheme with forward-secure private key generator[J]. Information Sciences, 2018, 454/455: 96-109.

[20] 杨小东, 李雨潼, 王晋利, 等. 标准模型下可撤销的基于身份的代理重签名方案[J]. 通信学报, 2019, 40(5): 153-162.
YANG X D, LI Y T, WANG J L, et al. Revocable identity-based proxy re-signature scheme in the standard model[J]. Journal on Communications, 2019, 40(5): 153-162.

[21] CANETTI R, HALEVI S, KATZ J. A forward-secure public-key encryption scheme[J]. Journal of Cryptology, 2007, 20(3): 265-294.

[22] WEI J H, CHEN X F, WANG J F, et al. Enabling (end-to-end) encrypted cloud emails with practical forward secrecy[J]. IEEE Transactions on Dependable and Secure Computing, 2021, PP(99): 1.

[23] DERLER D, GELLERT K, JAGER T, et al. Bloom filter encryption and applications to efficient forward-secret 0-RTT key exchange[J]. Journal of Cryptology, 2021, 34(2): 13.

[24] BELLARE M, STEPANOVS I, WATERS B. New negative results on differing-inputs obfuscation[C]//Advances in Cryptology – EUROCRYPT 2016. Berlin: Springer, 2016: 792-821.

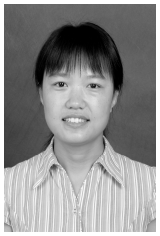
[25] HALEVI S, ISHAI Y, JAIN A, et al. Non-interactive multiparty computation without correlated randomness[C]//Advances in Cryptology – ASIACRYPT 2017. Berlin: Springer, 2017: 181-211.

[26] LI X Y, XU J, FAN X, et al. Puncturable signatures and applications in proof-of-stake blockchain protocols[J]. IEEE Transactions on Information Forensics and Security, 2020, 15: 3872-3885.

[27] BLOOM B H. Space/time trade-offs in hash coding with allowable errors[J]. Communications of the ACM, 1970, 13(7): 422-426.

[28] AKINYELE J A, GARMAN C, MIERS I, et al. Charm: a framework for rapidly prototyping cryptosystems[J]. Journal of Cryptographic Engineering, 2013, 3(2): 111-128.

[作者简介]



杨冬梅（1977− ），女，河南夏邑人，信息工程大学博士生，主要研究方向为应用密码学、大数据安全等。

陈越（1965− ），男，河南开封人，博士，信息工程大学教授、博士生导师，主要研究方向为网络与信息安全、大数据安全。

魏江宏（1987− ），男，甘肃通渭人，博士，信息工程大学讲师，主要研究方向为应用密码学、数据安全与隐私保护、机器学习安全等。

胡学先（1982− ），男，湖北红安人，博士，信息工程大学副教授，主要研究方向为密码协议、大数据安全、隐私保护等。