软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007469] [CSTR: 32375.14.jos.007469]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于国密 SM9 的密钥隔离签名*

高 睿[1,2], 丁 昀[1], 高 欣[1], 王化群[1,2]

[1](南京邮电大学 计算机学院、软件学院、网络空间安全学院, 江苏 南京 210023)

[2](江苏省密码技术工程研究中心, 江苏 南京 210023)

通信作者: 王化群, E-mail: wanghuaqun@aliyun.com

摘　要: 签名计算通常在移动电话或小型物联网设备等不安全的物理设备上进行, 这可能导致私钥暴露, 从而引发整个密码系统的崩溃. 密钥隔离签名方案是减轻私钥暴露造成的损害的一种方法. 在密钥隔离密码系统中, 公钥在整个时间周期内保持不变, 固定私钥被存储在物理安全设备上. 在每个离散的时间段开始时, 不安全设备通过与存储固定私钥的物理安全设备的交互以获得当前时间片的临时私钥. 一个安全的基于身份的密钥隔离签名方案需要满足签名不可伪造性和密钥隔离性. 密钥隔离性保证了即使一个攻击者获得了多个时间段的临时私钥, 它也无法伪造其他时间段的签名. SM9 是我国自主设计的商用标识密码算法. 将密钥隔离方法应用于 SM9 基于身份的签名方案中, 解决原方案中存在的私钥暴露问题. 首先给出基于身份的密钥隔离签名的安全模型. 然后构造一个基于身份的 SM9 密钥隔离签名方案. 最后给出详细的安全性证明和实验分析.

关键词: 国密 SM9; 基于身份的签名; 前向-后向安全; 安全性分析

中图法分类号: TP309

中文引用格式: 高睿, 丁昀, 高欣, 王化群. 基于国密SM9的密钥隔离签名. 软件学报. http://www.jos.org.cn/1000-9825/7469.htm

英文引用格式: Gao R, Ding Y, Gao X, Wang HQ. Key-isolated Signature Based on SM9. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7469.htm

Key-isolated Signature Based on SM9

GAO Rui[1,2], DING Yun[1], GAO Xin[1], WANG Hua-Qun[1,2]

[1](School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

[2](Jiangsu Cryptographic Technology Engineering Research Center, Nanjing 210023, China)

Abstract: The computation of signatures is typically performed on physically insecure devices such as mobile phones or small IoT devices, which may lead to private key exposure and subsequently compromise the entire cryptographic system. Key-insulated signature schemes serve as a method to mitigate the damage caused by private key exposure. In a key-insulated cryptosystem, the public key remains constant throughout the entire time period, and the fixed private key is stored on a physically secure device. At the beginning of each time period, the insecure device interacts with the physically secure device storing the fixed private key to obtain the temporary private key for the current time slice. A secure identity-based key-insulated signature scheme must satisfy both unforgeability and key insulation. Key insulation ensures that even if an adversary obtains temporary private keys for multiple time periods, they cannot forge signatures for other periods. SM9 is a commercial identity-based cryptographic standard independently developed by China. This study applies the key-insulated method to the SM9 identity-based signature scheme to resolve the private key exposure issue present in the original scheme. First, a security model for identity-based key-insulated signatures is presented. Then, an identity-based key-insulated signature scheme based on SM9 is constructed. Finally, detailed security proofs and experimental analysis are provided.

Key words: SM9; identity-based signature; forward-backward security; security analysis

* 基金项目: 国家自然科学基金 (U23B2002); 江苏省研究生科研创新计划 (KYCX25_1146, KYCX25_1159)
收稿时间: 2024-12-03; 修改时间: 2025-03-17; 采用时间: 2025-05-12; jos 在线出版时间: 2025-09-10

1 引 言

传统的公钥基础设施 (public key infrastructure, PKI) 框架使用证书来确保用户公钥的真实性. 用户注册时, 证书颁发机构为其公钥和身份信息颁发证书. 用户的证书本质上是绑定用户实体及其公钥的数字签名, 第三方可以通过验证证书的正确性来验证用户的身份. 尽管 PKI 框架提供了诸如机密性、完整性和不可否认性等重要的安全属性, 但其部署和管理的主要挑战在于对支持证书的公钥基础设施的需求[1]. 1984 年, Shamir[2]提出了一种基于身份的密码体系 (identity-based cryptography, IBC) 来解决这一问题. 在基于身份的密码学方案中, 用户的公钥可以直接从其身份信息 (如姓名、电子邮件地址等) 生成, 而无需传统的公钥基础设施框架中的证书, 简化了公钥管理流程.

几乎所有的密码系统都依赖一个强有力的假设, 即用户能够始终安全地保护自己的私钥. 然而, 现实中私钥面临着越来越多的暴露风险. 首先, 密码系统常常运行在物理安全性较低的设备上, 如手机和物联网边缘节点, 这些设备防御攻击的能力有限, 攻击者可以通过攻击设备来获取其中存储的私钥. 其次, 包含侧信道攻击在内的攻击手段[3,4]的不断丰富也增加了防御难度. 在许多情况下, 从被盗设备中获取私钥 (或通过欺骗毫无戒心的用户) 比破坏系统安全所依赖的计算假设要容易得多.

为了减轻私钥暴露带来的危害, 在 PKI 框架下, 通常使用认证撤销列表 (certification revocation list, CRL) 来撤销被暴露私钥对应的公钥. 然而, 在基于身份的密码方案中, 私钥暴露可能意味着暴露私钥的相应实体不能再将其身份信息作为公钥.

在身份基密码系统中, 处理私钥暴露问题的方法主要有两种. 第 1 种通过秘密共享[5], 用户将私钥拆成多个分片并分发给多个服务器. 阈值密码学[6]是这种方法的典型实例. 然而, 门限式共享的成本相对较高, 适用于大公司或认证机构, 而普通用户通常无法采用. 前向安全[7,8]是解决私钥暴露问题的另一个解决方案. 在前向安全方案中, 用户会在每个时间段开始时更新自己的私钥并销毁旧私钥. 因此当前私钥的泄露不会使对手在之前的任何时间段内破坏该方案. 但是, 这样一个完全暴露的私钥必然会损害未来所有系统的安全性, 即不满足后向安全性.

2002 年, Dodis 等人[9]提出了密钥隔离密码学的方法. 在密钥隔离方案中, 固定私钥被存储在物理安全设备中, 这保证了被存储的固定私钥的安全性. 协议的生命周期分为 N 个不同的周期. 在每个周期开始时, 物理不安全设备通过与物理安全设备交互来更新存储在物理不安全设备中的临时私钥. 所有的签名/解密操作都是由物理不安全设备使用临时私钥完成的.

密钥隔离方案需要符合密钥隔离性. k-N 密钥隔离性要求在暴露任意 k 个时间片的私钥的情况下, 未泄漏的时间片的私钥仍然能保证安全并且签名具有不可伪造性. 完美密钥隔离性要求在最坏情况下, 即对手自适应地选择 N 个时间片中的 $N-1$ 个时间片的临时密钥发生泄露时, 未泄漏的时间片的私钥仍然能保证安全. 当将密钥隔离方法应用于基于身份的签名时, 可以有效地解决基于身份的方案的私钥暴露问题.

近年来, 我国为满足自主安全可控的战略需求, 发布了多项密码算法, 其中 SM9 标识密码[10]是我国自主设计的商用标识密码. 然而, SM9 旨在满足信息系统通用的基本安全需求 (即数据机密性和完整性). 目前暂无针对 SM9 设计的密钥隔离签名方案.

本文贡献如下: 本文提出了密钥隔离的 SM9 身份基签名算法. 该方案引入了一个受用户控制的物理安全设备. 私钥生成中心为用户生成固定私钥并存储在物理安全设备中. 每个时间段 i 开始时, 物理安全设备使用固定私钥生成当前时段对应的临时私钥发送给用户的物理不安全设备 (如手机等), 物理不安全设备使用临时私钥对发送消息进行签名. 即使敌手在某些时间段获得了用户泄露的临时私钥, 仍无法伪造其他时间段的签名, 保障了签名系统的前向安全和后向安全.

本文第 2 节介绍密钥隔离密码系统和国密 SM9 密码体制的相关工作. 第 3 节介绍基础知识、身份基密钥隔离签名的系统模型和安全模型. 第 4 节给出密钥隔离 SM9 签名方案的详细构造. 第 5 节给出完整的安全性分析, 证明了本方案符合基于身份的签名不可伪造性和密钥隔离性. 第 6 节给出性能分析与实验结果.

2 相关工作

2.1 密钥隔离密码系统

2002 年, Dodis 等人[9]首次提出了密钥隔离的概念和相应的安全模型, 并且设计了首个密钥隔离加密算法, 在离散对数假设下证明了该方案的安全性. 由于密钥隔离系统具有前向安全性、后向安全性和高效率的优势, 密钥隔离系统逐渐引起了广泛关注. 自 2002 年以来, 关于密钥隔离方案的研究已取得了大量进展.

2003 年, Dodis 等人[11]证明了所有基于标准模型的签名方案都能构造完美的密钥隔离签名方案. 接着他们基于 Okamoto-Schnorr 签名方案和陷门签名提出了一种完美密钥隔离方案. 2003 年, Yum 等人[12]提出了一种基于身份的满足强密钥隔离的签名方案, 该方案中物理安全设备能够进行随机密钥更新并且满足完美密钥隔离性.

2006 年, Zhou 等人[13]基于 SOK-IBS 身份基签名方案设计了符合完美密钥隔离性的身份基签名方案, 该方案的优势在于并未增加原方案的签名长度. 同年, Weng 等人[14]指出方案 [13] 不满足强密钥隔离性, 即当存储在物理安全设备中的密钥泄露时无法保证系统的前后向安全性. 因此 Weng 等人提出了一种更强的“强密钥隔离”模型并设计了一种复合强密钥隔离性的签名方案. Hanaoka 等人[15]提出了平行密钥隔离加密方案. 该方案创新性地引入两个协助器, 通过交替进行密钥更新的方式运作, 且其中一个协助器的密钥与另一个协助器的密钥相互独立. 这一设计极大地降低了协助器密钥泄露的潜在风险, 有效增强了系统整体的安全性.

2014 年, Chen 等人[16]提出了一种属性基密钥隔离签名方案, 为密钥隔离系统增加了访问控制功能. 2017 年, Yu 等人[17]提出了一种强抗密钥泄漏审计方案, 该方案旨在最小化密钥泄露对系统整体安全性的长期影响. 同年, Vasudeva 等人[18]提出了基于身份的密钥隔离聚合签名方案, 有效规避了聚合签名中因密钥暴露而产生的风险, 同时完整保留基于身份签名体制所具备的优势. 2022 年, Shen 等人[19]提出了一种新的密钥更新技术, 支持惰性更新, 即仅允许用户在需要将文件上传到云时才更新私钥. 2024 年, 安睿诚等人[20]在方案 [17] 和方案 [19] 的基础上提出了抗密钥泄露的代理可证数据持有方案, 利用与密钥隔离签名近似的思想设计了一种远程数据完整性检测方案, 进一步拓展了密钥隔离签名的应用范围.

2.2 国密 SM9 标识签名

2008 年, 我国自主设计了国产标识密码-SM9 (商密九号算法)[10], 并于 2020 年成为国家标准, 于 2021 年成为国际标准, 为我国标识密码技术的应用奠定了坚实的基础.

自从 SM9 密码算法发布以来, 许多学者对其进行了安全分析和安全属性上的拓展. Cheng[21]分析了 SM9 系列算法的安全性. 赖建昌等人[22]基于 q-SDH 问题给出了 SM9 密钥交换协议、密钥封装机制和公钥加密算法的安全性证明, 并且对 SM9 数字签名算法和密钥封装算法的安全性进行了分析. 赖建昌等人[23]提出了一种基于商密 SM9 的高效标识广播加密方案. 唐飞等人[24]提出了基于 SM9 的加法同态加密方案. 蒲浪等人[25]基于 SM9 设计了一种公钥可搜索加密方案 SM9-PEKS. Shi 等人[26]提出了一种基于 SM9 的属性基加密方案. 陈荣茂等人[27]通过引入新的安全概念 IND-ID-RCCA 和可重随机化特性, 基于新的困难性假设, 在随机预言模型下证明了其安全性, 结合密码反向防火墙提升了 SM9 身份基方案对颠覆攻击的抗性.

目前, 基于国密 SM9 的身份基密钥隔离签名算法目前还没有人提出. 该算法设计主要的难点在于 SM9 用户私钥的生成方式较为复杂, 无法直接套用常见的密钥隔离签名范式进行临时私钥更新.

3 准备知识

本文用到的符号见表 1, 可忽略函数被记作 $negl(\lambda)$, 概率多项式时间 (probabilistic polynomial time) 被简写为 PPT, $\alpha \xleftarrow{\$} \mathbb{Z}_p$ 代指从模 p 的有限域中随机选取元素 α.

3.1 双线性映射

配对 $e: \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ 是一个双线性映射如果它满足以下性质:

- 双线性: $\forall(P,Q,a,b)\in\mathbb{G}_1\times\mathbb{G}_2\times\mathbb{Z}_p\times\mathbb{Z}_p: e(P^a,Q^b)=e(P^a,Q)^b=e(P,Q^b)^a=e(P,Q)^{ab}$.
- 非退化性: 存在 $P\in\mathbb{G}_1$ 和 $Q\in\mathbb{G}_2$, 满足 $e(P,Q)\neq 1$.
- 可计算性: 映射 e 可以被高效计算.

表 1 符号

符号	含义	符号	含义
λ	系统安全参数	ID	用户身份 (公钥)
i	当前时间片序号	SK_{ID}	用户固定私钥
α	系统主私钥	TSK_i	用户第 i 个时间段的临时私钥
P_{Pub}	系统主公钥	σ	用户签名

$\mathbb{G}_1,\mathbb{G}_2$ 和 $\mathbb{G}_T$ 都是 p 阶乘法循环群. 我们把一个双线性映射的所有参数记作 $bp=BilGen(1^\lambda)=(p,e,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,P_1,P_2)$. 本文中使用的双线性映射 e 来自国密 SM9 自带的双线性映射, 它是第 2 类双线性映射. 它的特点在于: (1) 非对称性, $\mathbb{G}_1\neq\mathbb{G}_2$. (2) 存在一个同态映射 $\phi:\mathbb{G}_1\to\mathbb{G}_2$.

3.2 困难问题

q-SDH 假设: 已知 $q+2$ 个元素 $(P,Q,Q^a,Q^{a^2},\ldots,Q^{a^q})\in\mathbb{G}_1\times\mathbb{G}_2^{q+1}$. 要求在 a 未知的情况下, 找到一个二元组 $(c,P^{\frac{1}{c+a}})$, $c\in\mathbb{Z}_q$. 若在多项式时间内解决 q-SDH 问题的概率是可忽略的, 则称 q-SDH 假设成立.

指数知识假设 (knowledge of exponent assumption, KEA): 给定双线性映射 bp, 和一对点 (P,P^α), 任意 PPT 敌手 $\mathcal{A}$ 在不知道 v 的情况下难以构造 $(A,\hat{A})$, 且 $\hat{A}=A^\alpha$. 形式化地, 对于任意 PPT 敌手 $\mathcal{A}$, 它接收输入 (bp,P,P^α), 输出 $(A,\hat{A})$, 且 $\hat{A}=A^\alpha$ 时, 存在一个 PPT 抽取器 $\mathcal{X}_{\mathcal{A}}$, 使得:

$$\Pr\left[\ bp=BilGen(1^\lambda),\alpha\xleftarrow{\$}\mathbb{Z}_p,(A,\hat{A};v)\leftarrow(\mathcal{A}\|\mathcal{X}_{\mathcal{A}})(bp,P,P^\alpha)\ \middle|\ \hat{A}=A^\alpha\wedge A\neq P^v\ \right]\leqslant negl(\lambda),$$

则称 KEA 成立.

3.3 密码隔离的身份基签名的定义

如图 1 所示, 一个密钥隔离的身份基签名方案涉及 3 个实体: 私钥生成中心 (key generation center, KGC)、用户和验证者. 他们具体的职责如下.

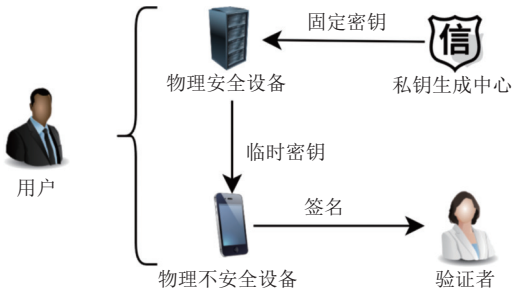

图 1 系统架构图

- 私钥生成中心: 生成系统主私钥和主公钥; 对于注册用户生成其身份 ID 对应的固定私钥并发送给他掌握的物理安全设备.
- 用户: 用户掌管着两类设备: 物理安全设备和物理不安全设备.

- 物理安全设备: 储存固定私钥; 与物理不安全设备交互, 定期更新存储在物理不安全设备上的临时私钥.
- 物理不安全设备: 储存临时私钥; 与物理安全设备交互, 定期更新临时私钥; 生成签名.

- 验证者: 根据当前时间段序号和用户身份验证签名.

下面我们给出了密钥隔离身份基签名方案的形式化定义.

定义 1. 一个密钥隔离身份基签名方案 (identity based key insulated signature, IDKIS) 包含 5 个 PPT 算法 (*Setup*, *FixKeyGen*, *TmpKeyGen*, *Sign*, *Verify*), 算法的形式化定义及简单说明如下.

● $(pp,\alpha) \leftarrow Setup(1^{\lambda})$: 系统初始化算法. 以安全参数 1^{λ} 作为输入, 返回公开参数 pp 和私钥生成中心主私钥 α.

● $(SK_{ID}) \leftarrow FixKeyGen(pp,\alpha,ID)$: 用户固定私钥生成算法. KGC 以公开参数 pp, 主私钥 α 和用户 $ID \in \{0,1\}^*$ 为输入, 计算用户固定私钥 SK_{ID} 并通过安全信道秘密发送给用户的物理安全设备.

● $(TSK_i) \leftarrow TmpKeyGen(pp,SK_{ID},ID,i)$: 用户临时私钥生成算法. 物理安全设备以公开参数 pp, 用户固定私钥 SK_{ID}, 用户 ID 和时间段序号 i 为输入, 返回用户临时私钥 TSK_i, 将临时私钥发送给物理不安全设备.

● $(\sigma) \leftarrow Sign(pp,TSK_i,ID,i,M)$: 用户签名算法. 物理不安全设备以公开参数 pp, 用户临时私钥 TSK_i, 用户 ID, 消息 M 和时间段序号 i 为输入, 返回对消息 M 的签名 σ.

● $(b \in \{0,1\}) \leftarrow Verify(pp,ID,i,\sigma,M)$: 验证签名算法. 验证者以公开参数 pp, 用户 ID, 消息 M, 时间段 i 和 σ 为输入, 返回一个比特 b, $b=0$ 时代表验证失败, $b=1$ 时代表验证通过.

3.4 密钥隔离的身份基签名的安全模型

一个 IDKIS 方案需要满足基于身份的签名不可伪造性和密钥隔离性. 密钥隔离性要求即便在 N 个时间片中有 $N-1$ 个时间片的临时私钥被泄露, 未泄露的时间片的密钥依然能够保持安全, 并且其签名具有不可伪造性. 文献 [13] 给出适应性选择消息, 身份标识和时间段攻击下的存在性不可伪造 (identity based key insulated existentially unforgeable, ID-KI-UF) 安全模型. 安全模型采用挑战者 (challenger) 和攻击者 (adversary) 之间的互动游戏进行定义, 共包括 5 个阶段. 在开始互动游戏之前, 攻击者必须提前告知挑战者挑战的时间段序号. 具体定义如下.

(1) 系统建立阶段: 输入安全参数 1^{λ}, 挑战者运行算法 $(pp,\alpha) \leftarrow Setup(1^{\lambda})$, 生成系统主公私钥对 (P_{Pub},α), 并将主公钥 P_{Pub} 发送给攻击者.

(2) 询问阶段: 攻击者发出询问 $q_1,q_2,\ldots,q_m$, 其中每个询问 q_j 为以下内容之一.

● 固定私钥询问: 攻击者选择标识 ID. 挑战者通过运行算 *FixKeyGen*, 生成与 ID 对应的固定私钥 SK_{ID}. 挑战者将 SK_{ID} 发送给对手.

● 临时私钥询问: 攻击者选择标识 ID 和时间段下标 i, 挑战者通过运行算法 *FixKeyGen*, 生成与 ID 和 i 对应的临时私钥 TSK_i. 挑战者将 TSK_i 发送给对手.

● 签名询问: 攻击者选择标识 ID、消息 M 和时间段下标 i. 挑战者生成签名 σ 发送给攻击者.

(3) 伪造阶段: 攻击者输出 $(ID^*,M^*,(i^*,\sigma^*))$, 即标识 ID^* 对消息 M^* 在时间段 i^* 的伪造签名 (i^*,σ^*), 其中, ID^* 没有在固定私钥询问阶段被询问过, 二元组 (ID^*,i^*) 没有在临时私钥询问阶段被询问过, 三元组 (ID^*,M^*,i^*) 没有在临时私钥询问阶段被询问过. 若 (i^*,σ^*) 是对消息 M^* 的有效签名, 则攻击者获胜.

定义攻击者的优势 $Adv_{\mathcal{A}}^{\text{ID-KI-UFM}}(\lambda)$ 为赢得以上 ID-KI-UF 游戏的概率.

定义 2. 在 ID-KI-UF 安全模型中, 如果对任意 PPT 攻击者 $\mathcal{A}$, $Adv_{\mathcal{A}}^{\text{ID-KI-UFM}}(\lambda)$ 都是可忽略的, 则称标识签名算法是 ID-KI-UF 安全的.

4 密钥隔离的 SM9 身份基签名算法

在本节中, 我们提供了一个 IDKIS 的具体构造, 包含以下 5 个算法.

● 系统初始化算法 *Setup*.

-运行双线性映射生成算法得到双线性映射实例 $bp = BilGen(1^{\lambda}) = (p,e,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,P_1,P_2)$, 其中 $e(\mathbb{G}_1,\mathbb{G}_2) \rightarrow \mathbb{G}_T$. $\mathbb{G}_1$ 上有生成元 P_1, $\mathbb{G}_2$ 上有生成元 P_2.

-产生哈希函数 $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p$、$H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_p$ 和 $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_p$.

-私钥生成中心随机选择主私钥 $\alpha \xleftarrow{\$} \mathbb{Z}_p$, 计算 $P_{\text{Pub}} = P_2^{\alpha}$, 随机选择群元素 $\beta \xleftarrow{\$} \mathbb{Z}_p$, 计算 $B = P_2^{\beta}$.

-输出公开参数 $pp=\{bp,H_1,H_2,H_3,P_{\text{Pub}},B\}$.

● 用户固定私钥生成算法 *FixKeyGen* (KGC 执行).

-用户将自己身份 ID 发送至私钥生成中心处.

-私钥生成中心计算 $Q_{ID}=H_1(ID)$, 再计算用户固定私钥 $SK_{ID}=P_1^{\frac{\alpha}{Q_{ID}+\alpha}}$.

-私钥生成中心通过秘密安全信道将私钥发送给用户.

-用户将固定私钥存储在物理安全设备中.

● 用户临时私钥生成算法 *TmpKeyGen* (物理安全设备执行).

-计算 $S_{ID,i}=P_1^{\rho}$, $C_i=B^{\rho}$.

-计算 $t_{ID,i}=H_2(ID\|S_{ID,i}\|i)$.

-计算 $T_{ID,i}=SK_{ID}^{\rho}=P_1^{\frac{\alpha\cdot\rho}{Q_{ID}+\alpha}}$.

-计算 $J_i=P_1^{\frac{\rho}{t_{ID,i}+\rho}}$.

-得到临时私钥为 $TSK_i=(S_{ID,i},C_i,T_{ID,i},J_i)$, 使用安全信道发送 TSK_i 给物理不安全设备.

● 用户签名算法 *Sign* (物理不安全设备执行).

-随机选取 $r\xleftarrow{\$}\mathbb{Z}_p$.

-计算群 $\mathbb{G}_T$ 中的元素 $w_1=e(S_{ID,i},P_{\text{Pub}})^r$, $w_2=e(S_{ID,i},P_2)^r$.

-计算 $h=H_3(ID\|M\|i\|w_1\|w_2)$.

-计算 $l=r-h$, $S_i=T_{ID,i}^l$.

-计算 $V_i=J_i^l$.

-得到签名为 $\sigma=(S_{ID,i},C_i,S_i,h,V_i)$.

● 验证签名算法 *Verify* (验证者执行).

-收到签名 $\sigma=(S_{ID,i},C_i,S_i,h,V_i)$ 和消息 M.

-验算 $e(P_1,C_i)\stackrel{?}{=}e(S_{ID,i},B)$, 若不成立则验证不通过, 输出“0”.

-计算 $Q_{ID}=H_1(ID)$, $t_{ID,i}=H_2(ID\|S_{ID,i}\|i)$.

-计算群 $\mathbb{G}_T$ 中的元素 $w_1'=e(S_i,P_2^{Q_{ID}}\cdot P_{\text{Pub}})\cdot e(S_{ID,i},P_{\text{Pub}})^h$.

-计算群 $\mathbb{G}_T$ 中的元素 $w_2'=e(V_i,P_2^{t_{ID,i}}\cdot\psi(S_{ID,i}))\cdot e(S_{ID,i},P_2)^h$.

-验算整数 $h\stackrel{?}{=}H_3(ID\|M\|i\|w_1'\|w_2')$ 是否成立.

正确性证明:

定理 1. 对于一个在时间段 i 的合法签名 $\sigma=(S_{ID,i},C_i,S_i,h,V_i)$, 验签算法 *Verify* 总是返回 1.

证明: 根据签名生成算法得到:

$$e(P_1,C_i)=e(P_1,B^{\rho})=e(P_1,P_2^{\rho\beta})=e(P_1,P_2^{\rho\beta})=e(S_{ID,i},B),$$

$$\begin{aligned}
w_1' &= e(S_i,P_2^{Q_{ID}}\cdot P_{\text{Pub}})\cdot e(S_{ID,i},P_{\text{Pub}})^h\\
&= e(T_{ID,i}^l,P_2^{Q_{ID}}\cdot P_2^{\alpha})\cdot e(S_{ID,i},P_{\text{Pub}})^h\\
&= e(P_1^{(r-h)\cdot\frac{\alpha\cdot\rho}{Q_{ID}+\alpha}},P_2^{Q_{ID}}\cdot P_2^{\alpha})\cdot e(S_{ID,i},P_{\text{Pub}})^h\\
&= e(P_1,P_2)^{(r-h)\cdot\frac{\alpha\cdot\rho}{Q_{ID}+\alpha}\cdot(Q_{ID}+\alpha)}\cdot e(P_1,P_2)^{h\cdot\alpha\cdot\rho}\\
&= e(P_1,P_2)^{(r-h)\cdot\alpha\cdot\rho}\cdot e(P_1,P_2)^{h\cdot\alpha\cdot\rho}\\
&= e(P_1,P_2)^{r\cdot\alpha\cdot\rho}\\
&= e(P_1^{\rho},P_2^{\alpha})^r\\
&= e(S_{ID,i},P_{\text{Pub}})^r\\
&= w_1,
\end{aligned}$$

$$
\begin{aligned}
w_2' &= e(V_i, P_2^{t_{ID,i}} \cdot \psi(S_{ID,i})) \cdot e(S_{ID,i}, P_2)^h \\
&= e(J_i^l, P_2^{t_{ID,i}} \cdot P_2^{\rho}) \cdot e(S_{ID,i}, P_2)^h \\
&= e(P_1^{(r-h)\cdot \frac{\rho}{t_{ID,i}+\rho}}, P_2^{t_{ID,i}} \cdot P_2^{\rho}) \cdot e(S_{ID,i}, P_2)^h \\
&= e(P_1, P_2)^{(r-h)\cdot \frac{\rho}{t_{ID,i}+\rho}\cdot(t_{ID,i}+\rho)} \cdot e(P_1, P_2)^{h\cdot\rho} \\
&= e(P_1, P_2)^{(r-h)\cdot\rho} \cdot e(P_1, P_2)^{h\cdot\rho} \\
&= e(P_1, P_2)^{r\cdot\rho} \\
&= e(P_1^{\rho}, P_2)^r \\
&= e(S_{ID,i}, P_2)^r \\
&= w_2,
\end{aligned}
$$

因此合法生成的签名均可通过验签算法. 证毕.

5 安全性证明

定理 2. 设密钥隔离的 SM9 签名算法中的哈希函数 H_1, H_2, H_3 是随机预言机, 如果 q-SDH 假设和 KEA 假设成立, 则密钥隔离的 SM9 签名算法是安全的.

证明: 假设在 ID-KI-UF 安全模型中, 存在 PPT 攻击者 $\mathcal{A}$, 在询问 q_{H_i} 次随机预言机 H_i ($i = 1,2,3$), q_E 次固定私钥生成预言机, q_T 次临时私钥生成预言机和 q_S 次签名预言机之后, 能以不可忽略的优势 ϵ 伪造签名, 则可构造一个 PPT 模拟器 $\mathcal{B}$ 以不可忽略的优势攻破 q-SDH 问题. $q+2$ 个元素 $(P, Q, Q^a, Q^{a^2}, \ldots, Q^{a^q}) \in \mathbb{G}_1 \times \mathbb{G}_2^{q+1}$ 是一个 q-SDH 问题实例. $\mathcal{B}$ 需要找到一个二元组 $(c, P^{\frac{1}{c+a}})$, 其中 $c \in \mathbb{Z}_q$.

● 系统建立阶段: $\mathcal{B}$ 首先执行以下操作, 隐式地将系统主私钥 α 设置为 a.

(1) 随机选择 $j^* \in [1,q]$, 从 $c \in \mathbb{Z}_q$ 中随机选取 q 个两两不同的数 $x^*, x_1, x_2, \ldots, x_{j^*-1}, x_{j^*+1}, \ldots, x_q$ 并定义多项式 $f(z) = \prod_{j=1, j\neq j^*}^{q} (z + x_j) = \sum_{j=0}^{q-1} c_j z^j \bmod q$, 其中 c_i 为多项式 $f(z)$ 的系数.

(2) 计算群 $\mathbb{G}_2$ 的生成元 $P_2 = Q^{f(a)} = Q^{a^j \sum_{j=0}^{q-1} c_j z^j}$, 计算群 $\mathbb{G}_1$ 的生成元 $P_1 = \psi(P_2) = P^{f(a)}$, 计算群 $\mathbb{G}_2$ 的元素 $P_{\text{Pub}} = Q^{af(a)} = Q^{a^{j+1} \sum_{j=0}^{q-1} c_j z^j}$ 作为主公钥.

(3) 对于任意的 $j \in [1,q]\backslash j^*$, 定义 $f_j(z) = \frac{f(z)}{z + x_j} = \sum_{i=0}^{q-2} d_j z^j \bmod p$, 计算 $P_1^{\frac{a}{a+x_j}} = P^{af_j(a)}$. 因此, 对于任意的 $j \in [1,q]\backslash j^*$, 二元组 $(x_j, V_j = P_1^{\frac{a}{a+x_j}})$ 是可以被计算的.

(4) 运行初始化设置的其余算法, 输出公开参数 $pp = \{bp, H_1, H_2, H_3, P_{\text{Pub}}, B\}$.

● 哈希询问阶段: $\mathcal{A}$ 询问预言机 H_1、H_2 和 H_3.

(1) H_1 询问: 在任何时间 $\mathcal{A}$ 都能询问预言机 H_1, 为了回复询问, $\mathcal{B}$ 维护一个元组列表 H_1^{list}, 列表中存储二元组 (ID, x). 当 $\mathcal{A}$ 选择标识 ID_j 问询预言机 H_1 时, $\mathcal{B}$ 按以下流程回复: 如果被问询的标识 ID_j 已经存在于列表 H_1^{list} 中, 那么 $\mathcal{B}$ 回复相应的 x_j. 否则, 记 ID_j 为第 j 个新标识的问询. 如果 $j = j^*$, 令 $H_1(ID_j) = x^*$, 将 x^* 发送给 $\mathcal{A}$ 并在 H_1^{list} 中添加二元组 (ID_j, x^*). 如果 $j \neq j^*$, 令 $H_1(ID_j) = x_j$, 将 x_j 发送给 $\mathcal{A}$ 并在 H_2^{list} 中添加二元组 (ID_j, x_j).

(2) H_2 询问: 在任何时间 $\mathcal{A}$ 都能询问预言机 H_2, 为了回复询问, $\mathcal{B}$ 维护一个元组列表 H_2^{list}, 列表中存储三元组 (ID, i, y). 当 $\mathcal{A}$ 选择标识和问询时间段 (ID_j, i_j) 问询预言机 H_2 时, $\mathcal{B}$ 按以下流程回复: 如果被问询的 (ID_j, i_j) 已经存在于列表 H_2^{list} 中, 那么 $\mathcal{B}$ 回复相应的 y_j. 否则, 随机选取 $y_j \xleftarrow{\$} \mathbb{Z}_p$, 令 $H_2(ID_j \| i_j) = y_j$, 并且将 y_j 发送给 $\mathcal{A}$ 并在 H_2^{list} 中添加三元组 (ID_j, i_j, y_j).

(3) H_3 询问: 在任何时间 $\mathcal{A}$ 都能询问预言机 H_3, 为了回复询问, $\mathcal{B}$ 维护一个元组列表 H_3^{list}, 列表中存储六元组 (ID, M, i, w_1, w_2, z). 当 $\mathcal{A}$ 选择 $(ID_j, M_j, i_j, w_{1,j}, w_{2,j})$ 问询预言机 H_3 时, $\mathcal{B}$ 按以下流程回复: 如果被问询的 $(ID_j, M_j, i_j, w_{1,j}, w_{2,j})$ 已经存在于列表 H_3^{list} 中, 那么 $\mathcal{B}$ 回复相应的 z_j. 否则, 随机选取 $z_j \xleftarrow{\$} \mathbb{Z}_p$, 令 $H_3(ID_j \| i_j) = y_j$, 并且将 z_j 发送

给 $\mathcal{A}$ 并在 H_3^{list} 中添加六元组 $(ID_j, M_j, i_j, w_{1,j}, w_{2,j}, z_j)$.

(4) 固定私钥询问: 记 ID_j 为第 j 个被询问的新身份, 若 $j = j^*$, 则 $\mathcal{B}$ 停止模拟. 若 $j \neq j^*$, $\mathcal{B}$ 询问 H_1 预言机得到 x_j, 返回 $SK_{ID} = P_1^{\frac{a}{Q_{ID}+a}}$ 作为 ID_j 的固定签名私钥.

(5) 临时私钥询问: 记 (ID_j, i_j) 为第 j 个被询问的新二元组, 若 $j = j^*$, 则 $\mathcal{B}$ 停止模拟. 若 $j \neq j^*$, $\mathcal{B}$ 询问 H_1 预言机得到 x_j, 进而得到 ID_j 的固定签名私钥 $SK_{ID,i_j} = P_1^{\frac{a}{x_j+a}}$. 进一步地, 运行临时私钥生成算法得到临时私钥 TSK_{i_j} 并返回给 $\mathcal{A}$.

(6) 签名询问: 记 (ID_j, M_j, i_j) 为第 j 个被询问的新三元消息组, 如果 $j \neq j^*$, 那么 $\mathcal{B}$ 可以获得对应的固定/临时签名私钥, 进而生成有效签名. 如果 $j = j^*$, 那么 $\mathcal{B}$ 随机选取 $h \xleftarrow{\$} \mathbb{Z}_p$, $\rho \xleftarrow{\$} \mathbb{Z}_p$, $V_i \xleftarrow{\$} \mathbb{G}_1$, $S_i \xleftarrow{\$} \mathbb{G}_1$, $S_{ID,i} = P_1^{\rho}$, $C_i = B^{\rho}$, 计算群 $\mathbb{G}_T$ 中的元素 $w_1 = e(S_i, P_2^{Q_{ID}} \cdot P_{\text{Pub}}) \cdot e(S_{ID,i_j}, P_{\text{Pub}})^h$. 计算群 $\mathbb{G}_T$ 中的元素 $w_2 = e(V_i, P_2^{t_{ID_j,i_j}} \cdot \psi(S_{ID,i_j})) \cdot e(S_{ID,i_j}, P_2)^h$. 最后定义 $H_3(ID_j \| M_j \| i_j \| w_1 \| w_2) = h$. 输出签名 $\sigma = (S_{ID,i_j}, C_{i_j}, S_{i_j}, h, V_{i_j})$.

● 伪造阶段: $\mathcal{A}$ 输出伪造签名 $(\sigma^*, M^*, ID^*, i^*)$, 其中 $\sigma^* = (S_{ID,i}^*, C_i^*, h^*, S_i^*, V_i^*)$, 令 ID^* 的下标为 j. 若 $j \neq j^*$, $\mathcal{B}$ 停止模拟并输出失败. 若 $j = j^*$, 根据分叉引理, 考虑到签名为 $(S_{ID,i}^*, C_i^*, h^*, S_i^*, V_i^*)$, 可以认为是一个 3 次信息传递的零知识证明协议. 若存在一个攻击算法 $\mathcal{A}$ 能在时间 t 内以 $\epsilon > \dfrac{10(q_s+1)(q_s+q_h)}{2^\lambda}$ 的概率成功伪造签名 $(S_{ID,i}, h, S_i, V_i)$, 则存在一个图灵机 $\mathcal{A}'$ 通过 $\mathcal{A}$ 的帮助, 以相同的输入 (pp, M, ID, i) 在时间 $t' < 120686 q_h t/\epsilon$ 内输出两个有效的签名 $(S_{ID,i}, C_i, h_1, S_{i,1}, V_{i,1})$ 和 $(S_{ID,i}, C_i, h_2, S_{i,2}, V_{i,2})$, 其中 $h_1 \neq h_2, S_1 \neq S_2$. 据此, $\mathcal{B}$ 运行图灵机 $\mathcal{A}'$, 获得两个关于 (M^*, ID^*, i^*) 的有效签名 $(S_{ID,i}^*, C_i^*, h_1^*, S_{i,1}^*, V_{i,1}^*)$ 和 $(S_{ID,i}^*, C_i^*, h_2^*, S_{i,2}^*, V_{i,2}^*)$, 且满足验证等式 $e(P_1, C_i^*) = e(S_{ID,i}^*, B)$ 和

$$\begin{cases} e(S_1^*, P_2^{H_1(ID^*)} P_{\text{Pub}}) \cdot e(S_{ID,i}^*, P_{\text{Pub}})^{h_1^*} = e(S_2^*, P_2^{H_1(ID^*)} P_{\text{Pub}}) \cdot e(S_{ID,i}^*, P_{\text{Pub}})^{h_2^*} \\ e(V_1^*, P_2^{H_2(ID^* \| i^*)} S_{ID,i}^*) \cdot e(S_{ID,i}^*, P_2)^{h_1^*} = e(V_2^*, P_2^{H_2(ID^* \| i^*)} S_{ID,i}^*) \cdot e(S_{ID,i}^*, P_2)^{h_2^*} \end{cases}.$$

根据 KEA 假设, 对于满足 $e(P_1, C_i^*) = e(S_{ID,i}^*, B)$ 的三元组 $(C_i^*, S_{ID,i}^*, B)$, 存在一个提取器 $\mathcal{E}$, 能够提取 ρ^* 使得 $S_{ID,i}^* = P_1^{\rho^*}$, 结合之前提到的两个验证等式可以得到 $e((S_1^* - S_2^*)^{a^{-1}(h_2^* - h_1^*)^{-1}}, P_2^{x^*+a}) = e(P_1^{\rho^*}, P_2)$.

令 $Y^* = (S_1^* - S_2^*)^{(h_2^* - h_1^*)^{-1} \rho^{*-1}}$, 那么根据等式有 $Y^* = P_1^{\frac{a}{x^*+a}}$. 又有 $\dfrac{zf(z)}{z+x^*} = \dfrac{\gamma}{z+x^*} + \sum_{i=0}^{k-1} \gamma_i z^i, P_1 = P^{f(a)}$, 其中 γ 和 γ_i 是可以计算的系数并且 γ 不为 0. 那么我们可以得到 $X^* = \dfrac{1}{\gamma}\left(Y^* - \sum_{i=0}^{k-1} \gamma_i \psi\left(a^i Q\right)\right) = \dfrac{1}{a+x^*} P$. 二元组 (x^*, X^*) 即为 q-SDH 问题的解.

综上所述, 如果 $\mathcal{A}$ 在伪造阶段输出针对 $ID^* = ID_{i^*}$ 的伪造签名. 这件事发生的概率为 $\dfrac{1}{q_{H_1}}$, 相应地, $\mathcal{B}$ 成功模拟的概率为 $\dfrac{1}{q_{H_1}}$. 因此, 若 $\mathcal{A}$ 能以不可忽略的概率 ϵ 成功伪造有效签名, 那么 $\mathcal{B}$ 能以 $\dfrac{\epsilon}{q_{H_1}}$ 的概率成功求解 q-SDH 问题.

6 性能分析与实验

6.1 安全性比较

如表 2 所示, 本文提出的方案在 SM9 身份基签名的基础上增加了密钥隔离机制, 进而确保了签名系统的前向安全、后向安全性. 相对的, 代价是在系统模型中引入了一个物理安全设备, 略微增大了系统运行的负担.

表 2 算法安全性比较

方案	不可伪造性	前向安全性	后向安全性
SM9	√	×	×
本文工作	√	√	√

6.2 性能理论分析

表 3 提供了本方案的运算成本以及和原 SM9 标识签名的比较. 与 SM9 相比, 本文提出的方案的计算时间和通讯开销均有所上升, 但是增加的开销在接受范围内.

KGC 生成固定私钥的开销与 SM9 一致. 密钥隔离机制添加了临时私钥生成算法, 因此带来了物理安全设备一定运算量上的提升, 并且增加了物理安全设备和不安全设备之间的 2 个群元素的通讯量. 物理不安全设备签名时增加了 2 次配对运算和 3 次 $\mathbb{G}_1$ 群上的指数运算.

表 3　算法计算开销比较

方案	固定私钥生成	临时私钥生成	签名	验签
SM9	$1E_1$+1*Hash*+3M	无	$1E_1$+1E_T+1*Hash*+1M	$1E_2$+1E_T+1P+2*Hash*
本文工作	$1E_1$+1*Hash*+3M	$3E_1$+1E_2+1*Hash*+4M	$4E_1$+2P+1*Hash*+1M	$2E_1$+6P+3*Hash*
KS-SOK-IBS[13]	$1E_2$+1*Hash*	$1E_1$+1E_2+E_T+1*Hash*	$1E_1$+1E_2+1E_T+1*Hash*	4P+3*Hash*

注: $e(\mathbb{G}_1,\mathbb{G}_2)\to\mathbb{G}_T$ 是一个双线性映射. E_1 代指群 $\mathbb{G}_1$ 中的指数运算; E_2 代指群 $\mathbb{G}_2$ 中的指数运算; E_T 代指群 $\mathbb{G}_T$ 中的指数运算; M 代表 $\mathbb{Z}_p$ 域中的计算; P 代表双线性配对运算; *Hash*代指哈希运算

本文的签名算法在时间段 i 中输出的一个签名为 $\sigma=(S_{ID,i},C_i,S_i,h,V_i)$.通讯开销为 1 个 $\mathbb{G}_1$ 群元素, 3 个 $\mathbb{G}_2$ 群元素和 1 个有限域元素. 注意到签名方案中在某一确定的时段中, 签名的某些部分是固定不变的. 具体地说, 签名中 $S_{ID,i}$ 和 C_i 在同一时间段 i 中是固定的. 换言之, 如果在同一个时间段内, 用户与验证者之间不断地对不同的信息执行“签名-验签”算法, $S_{ID,i}$ 和 C_i 只需要传递一次, 这大大减少了通讯开销和验证时间.

6.3 实　验

本节我们评估了基于国密 SM9 的密钥隔离签名的性能. 在一个有 3.20 GHz 处理器和 8 GB 内存的虚拟机上使用 Go 语言 (Go 版本 go1.20.2 linux/amd64) 和 gmsm 库实现了本文的方案. 实验遵循 SM9 标识密码算法[28]中双线性对的参数选取. $\mathbb{G}_1$ 中的一个点被存储为 32 个字节. 当需要存储 $\mathbb{G}_2$ 中的一个点时, 会将其映射到 $\mathbb{G}_1$ 群上再存储, 因此也是 32 个字节 (因为 SM9 使用的曲线是 II 型双线性映射). 哈希输出结果为 32 个字节.

计算开销如表 4 所示, 私钥生成中心需要花费 0.079 ms 为用户生成私钥, 并需要使用秘密安全信道传输 0.032 KB 的私钥给物理安全设备. 每个时间段开始时, 物理安全设备需要消耗 0.57 ms 产生临时私钥并且发送 0.064 KB 的数据给物理不安全设备. 签名时物理不安全设备需要计算 2.66 ms. 产生签名的长度为 0.16 KB. 验证者验证签名时需要消耗 7.26 ms, 开销增长的主要原因是验签算法涉及了 6 个双线性配对操作. 当用户和验证者在同一时间段内多次通讯时, 由于签名中的 $S_{ID,i}$ 和 C_i 不需要重复传输, 实际单个签名大小只有 0.9 KB 的数据 (关系图如图 2 所示), 对 100 条不同消息执行签名操作只需要传递 9.66 KB 的数据.

表 4　实测算法计算开销比较

方案	固定私钥生成 (ms)	临时私钥生成 (ms)	签名 (ms)	验签 (ms)	签名大小 (KB)	100次签名大小 (KB)
SM9	0.079	无	0.84	1.26	0.064	6.4
本文工作	0.079	0.57	2.66	7.23	0.16	9.66
KS-SOK-IBS[13]	0.34	0.43	1.19	4.70	0.96	64.3

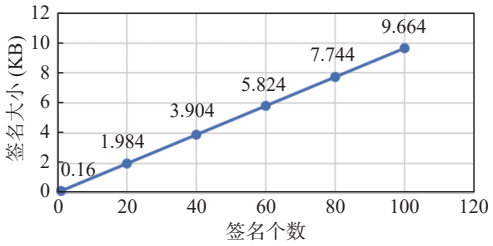

图 2　同一时间段签名大小和签名个数之间的关系

7 总　结

本文提出了一种基于国密 SM9 的密钥隔离签名方案, 解决了国产标识密码中存在的私钥暴露问题, 实现了前

向-后向安全性. 根据密钥隔离签名的安全模型, 我们基于随机预言机模型证明了我们的方案符合前向-后向安全性和不可伪造性. 理论分析和实验证明了本文提出的方案具有很高的实用性.

References

[1] Yum DH, Lee PJ. Identity-based cryptography in public key management. In: Proc. of the 1st European Public Key Infrastructure Workshop. Samos Island: Springer, 2004. 71–84. [doi: 10.1007/978-3-540-25980-0_6]

[2] Shamir A. Identity-based cryptosystems and signature schemes. In: Proc. of the 1985 Advances in Cryptology. Berlin: Springer, 1985. 47–53. [doi: 10.1007/3-540-39568-7_5]

[3] Wang JN, Zhu B, Yu WX, Wang W, Hu FL. Side channel analysis attack based on deep learning LSTM. Computer Engineering, 2021, 47(10): 140–146 (in Chinese with English abstract). [doi: 10.19678/j.issn.1000-3428.0059210]

[4] Peng P, Zhang ML, Zheng D. Side channel attack fused with CNN-LSTM. Computer Engineering and Applications, 2023, 59(6): 268–276 (in Chinese with English abstract). [doi: 10.3778/j.issn.1002-8331.2111-0511]

[5] Shamir A. How to share a secret. Communications of the ACM, 1979, 22(11): 612–613. [doi: 10.1145/359168.359176]

[6] Desmedt Y, Frankel Y. Threshold cryptosystems. In: Proc. of the 1990 Annual Int'l Cryptology Conf. New York: Springer, 1990. 307–315. [doi: 10.1007/0-387-34805-0_28]

[7] Anderson R. Two remarks on public key cryptology. Technical Report, UCAM-CL-TR-549. Cambridge: University of Cambridge, 2002. [doi: 10.48456/tr-549]

[8] Bellare M, Miner SK. A forward-secure digital signature scheme. In: Proc. of the 19th Annual Int'l Cryptology Conf. on Advances in Cryptology. Santa Barbara: Springer, 1999. 431–448. [doi: 10.1007/3-540-48405-1_28]

[9] Dodis Y, Katz J, Xu SH, Yung M. Key-insulated public key cryptosystems. In: Proc. of the 2002 Advances in Cryptology — EUROCRYPT. Amsterdam: Springer, 2002. 65–82. [doi: 10.1007/3-540-46035-7_5]

[10] Cheng ZH. The SM9 cryptographic schemes. 2017. https://eprint.iacr.org/2017/117.pdf

[11] Dodis Y, Katz J, Xu SH, Yung M. Strong key-insulated signature schemes. In: Proc. of the 6th Int'l Workshop on Practice and Theory in Public Key Cryptography. Miami: Springer, 2002. 130–144. [doi: 10.1007/3-540-36288-6_10]

[12] Yum DH, Lee PJ. Efficient key updating signature schemes based on IBS. In: Proc. of the 9th IMA Int'l Conf. on Cryptography and Coding. Cirencester: Springer, 2003. 167–182. [doi: 10.1007/978-3-540-40974-8_14]

[13] Zhou Y, Cao ZF, Chai ZC. Identity based key insulated signature. In: Proc. of the 2nd Int'l Conf. on Information Security Practice and Experience. Hangzhou: Springer, 2006. 226–234. [doi: 10.1007/11689522_21]

[14] Weng J, Liu SL, Chen KF, Li XX. Identity-based key-insulated signature with secure key-updates. In: Proc. of the 2nd SKLOIS Conf. on Information Security and Cryptology. Beijing: Springer, 2006. 13–26. [doi: 10.1007/11937807_2]

[15] Hanaoka G, Hanaoka Y, Imai H. Parallel key-insulated public key encryption. In: Proc. of the 2006 Int'l Workshop on Public Key Cryptography. New York: Springer, 2006. 105–122. [doi: 10.1007/11745853_8]

[16] Chen JH, Long Y, Chen KF, Guo J. Attribute-based key-insulated signature and its applications. Information Sciences, 2014, 275: 57–67. [doi: 10.1016/j.ins.2014.02.021]

[17] Yu J, Wang HQ. Strong key-exposure resilient auditing for secure cloud storage. IEEE Trans. on Information Forensics and Security, 2017, 12(8): 1931–1940. [doi: 10.1109/TIFS.2017.2695449]

[18] Vasudeva Reddy P, Gopal PVSSN. Identity-based key-insulated aggregate signature scheme. Journal of King Saud University — Computer and Information Sciences, 2017, 29(3): 303–310. [doi: 10.1016/j.jksuci.2015.09.003]

[19] Shen WT, Yu J, Yang M, Hu JK. Efficient identity-based data integrity auditing with key-exposure resistance for cloud storage. IEEE Trans. on Dependable and Secure Computing, 2023, 20(6): 4593–4606. [doi: 10.1109/TDSC.2022.3228699]

[20] An RC, Wang HQ. Proxy provable data possession with key-exposure resilient. Computer Science, 2024, 51(12): 310–316 (in Chinese with English abstract). [doi: 10.11896/jsjkx.231100085]

[21] Cheng ZH. Security analysis of SM9 key agreement and encryption. In: Proc. of the 14th Int'l Conf. on Information Security and Cryptology. Fuzhou: Springer, 2018. 3–25. [doi: 10.1007/978-3-030-14234-6_1]

[22] Lai JC, Huang XY, He DB, Wu W. Security analysis of SM9 digital signature and key encapsulation. Scientia Sinica Informationis, 2021, 51(11): 1900–1913 (in Chinese with English abstract). [doi: 10.1360/SSI-2021-0049]

[23] Lai JC, Huang XY, He DB. An efficient identity-based broadcast encryption scheme based on SM9. Chinese Journal of Computers, 2021, 44(5): 897–907 (in Chinese with English abstract). [doi: 10.11897/SP.T.1016.2021.00897]

[24] Tang F, Ling GW, Shan JY. Additive homomorphic encryption schemes based on SM2 and SM9. Journal of Cryptologic Research, 2022,

9(3): 535–549 (in Chinese with English abstract). [doi: 10.13868/j.cnki.jcr.000532]
[25] Pu L, Lin C, Wu W, He DB. A public-key encryption with keyword search scheme from SM9. Journal of Cyber Security, 2023, 8(1): 108–118 (in Chinese with English abstract). [doi: 10.19363/J.cnki.cn10-1380/tn.2023.01.08]
[26] Shi Y, Ma ZY, Qin RF, Wang XP, Wei WJ, Fan HF. Implementation of an attribute-based encryption scheme based on SM9. Applied Sciences, 2019, 9(15): 3074. [doi: 10.3390/app9153074]
[27] Chen RM, Chen JR, Huang XY, Wang Y. RCCA-SM9: Securing SM9 on corrupted machines. Science China Information Sciences, 2024, 67(11): 212103. [doi: 10.1007/s11432-023-3877-9]
[28] GM/T 0044-2016. Identity-based cryptographic algorithms SM9. 2016 (in Chinese). https://www.antpedia.com/standard/8380068-1.html

附中文参考文献

[3] 王俊年, 朱斌, 于文新, 王皖, 胡钒梁. 基于深度学习 LSTM 的侧信道分析. 计算机工程, 2021, 47(10): 140–146. [doi: 10.19678/j.issn.1000-3428.0059210]
[4] 彭佩, 张美玲, 郑东. 融合 CNN-LSTM 的侧信道攻击. 计算机工程与应用 2023, 59(6): 268–276. [doi: 10.3778/j.issn.1002-8331.2111-0511]
[20] 安睿诚, 王化群. 抗密钥泄露的代理可证数据持有. 计算机科学, 2024, 51(12): 310–316. [doi: 10.11896/jsjkx.231100085]
[22] 赖建昌, 黄欣沂, 何德彪, 伍玮. 国密 SM9 数字签名和密钥封装算法的安全性分析. 中国科学: 信息科学, 2021, 51(11): 1900–1913. [doi: 10.1360/SSI-2021-0049]
[23] 赖建昌, 黄欣沂, 何德彪. 一种基于商密 SM9 的高效标识广播加密方案. 计算机学报, 2021, 44(5): 897–907. [doi: 10.11897/SP.T.1016.2021.00897]
[24] 唐飞, 凌国玮, 单进勇. 基于国密 SM2 和 SM9 的加法同态加密方案. 密码学报, 2022, 9(3): 535–549. [doi: 10.13868/j.cnki.jcr.000532]
[25] 蒲浪, 林超, 伍玮, 何德彪. 基于 SM9 的公钥可搜索加密方案. 信息安全学报, 2023, 8(1): 108–118. [doi: 10.19363/J.cnki.cn10-
[27] 1380/tn.2023.01.08]
[28] GM/T 0044-2016. SM9 标识密码算法. 2016. https://www.antpedia.com/standard/8380068-1.html

作者简介

高睿, 博士生, 主要研究领域为零知识证明, 区块链安全.
丁昀, 硕士生, 主要研究领域为密码学.
高欣, 硕士生, 主要研究领域为格密码, 签名技术.
王化群, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为应用密码学, 云计算安全.