计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202220809
61(4): 1070−1084, 2024

支持等式测试及密码逆向防火墙的 SM9 标识加密方案

熊 虎 林 烨 姚 婷
(电子科技大学信息与软件工程学院 成都 610054)
(网络与数据安全四川省重点实验室(电子科技大学) 成都 610054)
(xionghu.uestc@gmail.com)

SM9 Identity-Based Encryption Scheme with Equality Test and Cryptographic Reverse Firewalls

Xiong Hu, Lin Ye, and Yao Ting
(*School of Information and Software Engineering*, *University of Electronic Science and Technology of China*, *Chengdu* 610054)
(*Network and Data Security Key Laboratory of Sichuan Province* (*University of Electronic Science and Technology of China*), *Chengdu* 610054)

Abstract The identity-based encryption with equality test (IBEET) scheme solves the problem of certificate management in traditional equality test schemes and gets wide attention. However, the existing IBEET systems are difficult to resist penetration attacks and based on foreign cipher algorithm designs without independent intellectual property rights. To deal with this challenge, we propose a SM9 identity-based encryption scheme with equality test and cryptographic reverse firewalls (SM9-IBEET-CRF). The cipher reverse firewalls (CRF) which are deployed in the upstream channel between users and cloud server can re-randomize the information to protect against penetration attacks. This scheme expands SM9 identity-based encryption algorithm to IBEET, improves its efficiency and enriches the research of secret algorithm in cloud computing. We give the definition of SM9-IBEET-CRF and corresponding security models. In random oracle model, the scheme formalizes the IBE-IND-CCA and IBE-OW-CCA security into the BDH difficulty assumption by considering two different opponents. At the same time, we demonstrate that CRF deployment provides functionality-maintaining, security-preserving and exfiltration-resistant by considering the third opponent. The experimental simulation and analysis results show the effectiveness of the scheme.

Key words SM9; equality test; cryptographic reverse firewall (CRF); identity-based encryption; penetration attack

摘 要 支持等式测试的标识加密(identity-based encryption with equality test, IBEET)体制解决了传统等式测试方案中证书管理的问题,得到了广泛的关注.但现有的IBEET体制难以抵抗渗透攻击,且都是基于国外密码算法设计,不具有自主知识产权.基于此,提出一种支持等式测试并具有密码逆向防火墙的SM9标识加密方案(SM9 identity-based encryption scheme with equality test and cryptographic reverse firewalls, SM9-IBEET-CRF).该方案在用户与云服务器的上行信道间部署密码逆向防火墙(cryptographic reverse firewalls, CRF),对用户发出的信息执行重随机化以达到抵抗渗透攻击的作用.该方案拓展国密算法SM9至IBEET领域中,提升其运行效率并丰富国密算法在云计算领域的研究.给出了SM9-IBEET-CRF的形式化定义和安全模型,并在随机预言机模型中考虑2种不同的敌手将此方案在选择密文攻击下的不可区分性与单向

收稿日期: 2022-09-14; **修回日期**: 2023-05-22
基金项目: 国家重点研发计划项目(2022YFB2701402); 厅市共建智能终端四川省重点实验室开放课题(SCITLAB-1019)
This work was supported by the National Key Research and Development Program of China (2022YFB2701402) and the Open Project of Intelligent Terminal Key Laboratory of Sichuan Province (SCITLAB-1019).

性分别形式化地规约到BDH困难假设上. 同时，该方案通过考虑第3种敌手证明CRF的部署为其带来维持功能性、保留安全性以及抵抗渗透性. 实验仿真和分析结果展示了该方案的有效性.

关键词 SM9；等式测试；密码逆向防火墙；标识加密；渗透攻击

中图法分类号 TP391

云计算的快速发展带动了云存储的普及，越来越多的用户和企业选择将数据存储在云服务器中[1]. 云服务器通常由不可信第三方维护和管理，存在诸多安全隐患. 如果将数据直接存储在云端，可能会使数据被未授权实体访问，从而导致隐私泄露. 一种保护数据隐私和实现安全数据共享的方式是采用公钥加密算法将数据加密后再存储到云服务器[2]，但这种方式极大地阻碍了数据的可用性. 当用户对存储在云服务器的加密数据进行搜索时，最直接的方法是将所有加密数据下载到本地后执行解密再对明文信息进行搜索，显然这种方法极其繁琐且低效.

为了解决数据机密性和可搜索性之间的矛盾，Boneh等人[3]提出了支持关键词检索的公钥加密的概念. 在PKEKS系统中，发送者利用接收者的公钥和关键词生成关键词密文，并附在对应的文件密文后上传到云服务器. 接收者可以利用自己的私钥和关键词生成陷门上传至云服务器，随后云服务器利用陷门对关键词密文执行检索以判断关键词密文是否包含陷门中嵌入的关键词. 由于在检测过程中云服务器无法获取密文对应的关键词信息以及接收者私钥. 大量的PKEKS方案被提出以进一步提升公钥可搜索加密的安全性[4-5]、效率[6-7]和功能[8]. 尽管PKEKS支持对于密文的搜索功能，但其存在功能上的限制：无法对不同公钥加密的2条信息进行检索. 同时还存在着严重的安全隐患：关键词空间远小于密钥空间，攻击者可以借此实施关键词猜测攻击.

为了解决上述问题，Yang等人[9]在2010年提出了支持等式测试的公钥加密(public-key encryption with equality test, PKEET)方案. PKEET允许任何用户对不同公钥加密生成的密文进行对比以判断其中是否包含相同的明文. 由于对比的密文空间远大于密钥空间，关键词猜测攻击无法对PKEET系统生效. 受Yang等人[9]的启发，国内外学者围绕PKEET的授权测试[10-11]、通用构造[12]和应用场景(无证书[13]、签密[14]、异构[15])等展开大量研究，一系列具有等式测试功能的加密方案被陆续提出. 在传统支持等式测试的公钥加密方案中，公钥是一个不可读的字符串，需要公钥基础设施[16](public key infrastructure, PKI)系统来签发公钥证书以绑定用户的身份与公钥. 公钥证书包括用户的身份信息、权威机构的签名和各种参数，以结构化数据的形式存储. 这种复杂且昂贵的证书管理方式在实际应用场景中带来了棘手的证书管理问题. 基于此，Ma[17]受标识密码体制思想启发[18]，构造了支持等式测试的标识加密(identity-based encryption with equality test, IBEET)体制. 在IBEET中，用户的公钥根据其身份信息生成，而用户的私钥由私钥生成中心(private key generator, PKG)构建.

目前已有的IBEET方案虽然避免了证书管理的问题，却存在着严重的安全隐患：大部分IBEET方案都难以抵抗渗透攻击. 斯诺登事件表明[19]，攻击者可以在正常使用条件下秘密设置后门以泄露用户隐私. 受此启发，Bellare等人[20]提出的算法篡改攻击(algorithm-substitution attacks, ASA)表明攻击者可以非法占据用户的个人设备来篡改加密算法，从被篡改的密文中获取明文信息. 具体来说，Bellare等人[20]构建了颠覆加密(subverting encryption)框架，对ASA的受害者实施IV篡改攻击(IV-replacement attacks)和偏密文攻击(the biased-ciphertext attack). 攻击者利用颠覆密钥篡改随机化输入IV，继而跟踪篡改密文以获取明文. 目前大量的无状态随机化密码算法被证明几乎无法抵抗这类ASA. 因此，一旦用户遭遇ASA，云服务器上的所有加密数据都有被泄露的风险. 考虑到ASA的危害性，密码逆向防火墙(cryptographic reverse firewalls, CRF)的概念由Mironov等人[21]在斯诺登事件之后提出. CRF可以被认为是部署在用户与外部世界的一个实体，通过重随机化用户发送和接收的信息将其映射到与原始输出相同的空间中，达到抵抗隐私泄露的作用. CRF具有3个性质：维持功能性、保留安全性和抵抗渗透性. 到目前为止，CRF已被成功应用于密钥协商协议[22]、基于属性的加密体制[23]和基于签名的不经意电子信封[24]. 将CRF部署在云服务器与用户之间，分别负责用户密文与陷门的重随机化. 由于用户输出的是重随机化结果，即便算法遭到篡改，用户隐私也不会泄露. 基于此，有必要为IBEET构建CRF.

另外，目前IBEET都是基于国外密码算法设计

且存在计算与通信开销大的问题，还没有出现支持国产商用密码算法的高效等式测试方案. SM9 作为一种具有效率优势的双线性对标识密码算法(identity-based cryptographic algorithm)，可以良好地拓展至IBEET 领域中. 2016 年我国国家密码管理局正式发布 SM9 密码算法，其相关标准为“GM/T 0044—2016 SM9 标识密码算法”. SM9 标识密码算法主要包括 4 个部分：数字签名算法、密钥交换协议、密钥封装机制和标识加密算法. 其中 SM9 标识加密算法于 2021 年正式成为国标(ISO/IEC 18 033-5:2015/AMD 1:2021). 虽然 SM9 在密码技术和网络安全领域占据越来越重要的地位，但关于 SM9 在云环境安全下的研究却寥寥无几. 因此，为了实现我国密码算法国产自主，提高其在信息安全领域的核心竞争力，亟需推进国密算法在云计算场景中的研究应用. 基于此，本文提出了一种支持等式测试并具有密码逆向防火墙的 SM9 标识加密方案(SM9 identity-based encryption scheme with equality test and cryptographic reverse firewalls, SM9-IBEET-CRF). 本文的贡献有 3 点：

1) 本文将 SM9 标识加密算法应用于等式测试这一密码学原语，提出了支持等式测试的 SM9 标识密码方案(SM9 identity-based encryption scheme with equality test, SM9-IBEET). 利用 SM9 是标识密码算法这一性质，避免了传统等式测试算法中证书管理的问题. 与传统 IBEET 方案相比，具有更强的安全性和计算开销的优势. 同时，丰富国产商用密码算法在云计算领域的研究.

2) 解决传统 IBEET 体制难以抵抗渗透攻击的问题. 本文将 CRF 部署在用户与云服务器之间的上行信道，分别实现用户密文、陷门的重随机化. 攻击者即使非法占据用户设备，由于其输出结果要经过 CRF 的处理，无法造成明文信息泄露.

3) 本文实现了形式化的安全性证明. 严谨的安全性分析证明其满足选择密文攻击下的不可区分性(IND-CCA)和选择密文攻击下的单向性(OW-CCA). CRF 的设置使其具备抵抗算法篡改攻击的能力. 经过大量的实验仿真证明，SM9-IBEET-CRF 在计算与通信开销上具有一定的优势，适用于云计算场景. 相关工作有：

1) 支持等式测试的公钥加密体制. 支持等式测试的公钥加密方案首先由 Yang 等人[9] 在 2010 年提出. 该方案允许任何用户对不同公钥加密的密文进行比较，解决了关键词检索公钥加密方案的局限性. 接下来，Tang[25] 为 PKEET 提出了具有细粒度的授权机制，以确保只有被授权的用户才有能力执行等式测试. 此外，Tang[10] 提出了混合粒度授权的支持等式测试的公钥加密方案(all-or-nothing PKE-ET, AoN-PKE-ET)来实现粗细粒度授权. 然而，在实际中云服务器与用户需要交互来授权，导致了方案的不可拓展性. 为应对这一挑战，Tang[26] 和 Ma[11] 分别提出了带有灵活授权机制的等式测试方案，其基本思想是用户单独对云服务器进行授权. 为了解决 PKEET 中的密钥管理问题，Ma[17] 将标识加密体制集成到 PKEET 中，提出一种 IBEET 方案. Qu 等人[13] 提出的无证书等式测试加密方案(certificateless-based encryption with equality test, CLEET)，旨在同时避免密钥托管和证书管理的问题.Wang 等人[14] 将签密的概念引入等式测试中，有效降低了计算和通信开销. Xiong 等人[15] 提出的异构签密等式测试方案(heterogeneous signcryption scheme with equality test, HSC-ET)，则实现了 IBE 与 PKE 的异构等式测试系统. 目前已有的 PKEET 体制难以抵抗渗透攻击.

2) 密码逆向防火墙. 斯诺登事件爆发后，为了保护用户隐私和维持密码方案安全性，Mironov 等人[21] 提出 CRF 的概念. CRF 位于用户计算机与外部世界之间，通过修改用户设备输入和输出，为受到篡改算法攻击的用户提供安全保护. 2016 年，Dodis 等人[22] 设计了一种具有逆向防火墙的消息传输协议，缩短公钥交换轮数至 4 轮. 同年，Chen 等人[24] 基于可延展的平滑映射哈希函数(smooth projective Hash function, SPHF)为多个密码协议设计了 CRF 框架. 2018 年，Ma 等人[23] 将逆向防火墙的概念引入属性基加密体制，提出了一种基于密文在线/离线属性的加密算法. 2019 年，Zhou 等人[27] 提出具有逆向防火墙的标识加密体制(identity-based encryption with cryptographic reverse firewalls, IBE-CRF)，为 IBE 设计了 2 种 CRF 方案. 目前还不存在具有逆向防火墙的 PKEET 方案.

1 基础知识

1.1 非对称双线性对

给定 3 个循环群G_1,G_2,G_T，它们的阶均为素数N，P_1为G_1的生成元，P_2为G_2的生成元，存在非对称双线性对$e:G_1\times G_2\to G_T$，满足 3 个条件：

1) 双线性. 对任意$P\in G_1,Q\in G_2,a,b\in\mathbb{Z}_N$，有$e([a]P,[b]Q)=e(P,Q)^{ab}$.

2) 非退化性. $e(P_1,P_2)\neq 1$.

3) 可计算性. 对任意$P\in G_1,Q\in G_2$,存在有效的

算法计算$e(P,Q)$.

1.2 BDH 假设

BDH 假设首先由 Boneh 等人[28]在对称双线性配对中提出，然后被拓展到非对称双线性对中. 本文使用 Boyen 等人[29]在非对称双线性对中推广的 BDH 假设.

1）BDH 问题. 给定$(P_1,P_2,[a]P_1,[a]P_2,[b]P_i,[c]P_j)$，其中$a,b,c\in\mathbb{Z}_N^*$，$i,j\in\{1,2\}$，计算$e(P_1,P_2)^{abc}$是困难的.

2）BDH 假设. 给定一个 BDH 问题实例，不存在一个 PPT 攻击者$\mathcal{A}$具有不可忽略的优势计算出$e(P_1,P_2)^{abc}$. 其中$\mathcal{A}$的优势被定义为

$$Pr[\mathcal{A}(P_1,P_2,[a]P_1,[a]P_2,[b]P_i,[c]P_j)=e(P_1,P_2)^{abc}].$$

1.3 密码逆向防火墙（CRF）

CRF 位于用户计算机与外部实体之间，只能修改用户的输入输出消息. 对于用户而言，他们并不知道 CRF 的存在.

CRF 是一种具有状态的算法$\mathcal{W}$，它以当前状态和消息作为输入，输出更新后的状态和消息. 对于初始状态为σ的参与方$P=(receive,next,output)$和逆向防火墙$\mathcal{W}$，它们的组成被定义为

$$\begin{aligned}&\mathcal{W}^{\circ}P=(receive_{\mathcal{W}^{\circ}P}(\sigma,m))=\ receive_P(\sigma,\mathcal{W}(m))=\\&next_{\mathcal{W}^{\circ}P}=\mathcal{W}(next_P(\sigma))=\\&output_{\mathcal{W}^{\circ}P}(\sigma)=\ output_P(\sigma).\end{aligned}$$

当组成方$\mathcal{W}^{\circ}P$参与协议时，$\mathcal{W}$的状态被初始化为系统公开参数 *params*，如果$\mathcal{W}$是与参与方 P 共同组成的，则我们称$\mathcal{W}$为参与方 P 的逆向防火墙.

显然，参与方 P 希望获得管理并部署多台防火墙的权力. 这种多个防火墙的组合$(\mathcal{W}^{\circ}\mathcal{W}^{\circ}\cdots^{\circ}\mathcal{W}^{\circ}P)$只会增强系统的安全性，而不会破坏其初始协议的功能.

定义 1. CRF 具有维持功能性. 对任意逆向防火墙$\mathcal{W}$与参与方 P，令$\mathcal{W}^{1^{\circ}}P=\mathcal{W}^{\circ}P$，当任意多项式边界$k\geqslant 2$时，令$\mathcal{W}^{k^{\circ}}P=\mathcal{W}^{\circ}(\mathcal{W}^{k-1^{\circ}}P)$. 在$k\geqslant 1$的协议$\mathcal{P}$中，如果$\mathcal{W}^{k^{\circ}}P$维持了参与方$P_i$的功能$\mathcal{F}$，也就是说当$\left(\mathcal{W}^{k^{\circ}}P_i=\mathcal{W}^{\circ}P_i\right)\cap\left(\mathcal{F}_{\mathcal{W}^{\circ}P_i}=\mathcal{F}_{P_i}\right)=1$时，CRF 具有维持功能性.

定义 2. CRF 具有保留安全性. 对满足安全性$\mathcal{S}$，功能$\mathcal{F}$的协议$\mathcal{P}$与逆向防火墙$\mathcal{W}$：

1）对于任意改变协议功能$\mathcal{F}$的 PPT 攻击者$P_{\mathcal{A}}^*$，如果协议$\mathcal{P}_{P\Rightarrow\mathcal{W}^{\circ}P_{\mathcal{A}}^*}$仍满足安全性$\mathcal{S}$，则称 CRF 对于协议参与方 P 具有强保留安全性.

2）对于任意不改变协议功能$\mathcal{F}$的 PPT 攻击者$\hat{P}$，如果协议$\mathcal{P}_{P\Rightarrow\mathcal{W}^{\circ}\hat{P}}$仍满足安全性$\mathcal{S}$，则称 CRF 对于协议参与方 P 具有弱保留安全性.

当 PPT 攻击者$P_{\mathcal{A}}^*$破坏协议功能$\mathcal{F}$时，由于协议参与方P可以很快感知到这种破坏，在云计算场景中攻击效果较差，是一种罕见的攻击方式. 因此，本文主要证明 CRF 对于不破坏协议功能的 ASA 安全性.

定义 3. CRF 具有抵抗渗透性.CRF 具有阻止参与方 P 对消息篡改泄露的能力. 使用 Mironov 等人[21]定义的泄露游戏$LEAK(\mathcal{P},P,J,\mathcal{W},\lambda)$，如图 1 所示：$\mathcal{P}$代表密码协议，$P$ 为协议正常参与方，J 代表被敌手控制的协议参与方，$\mathcal{W}$为逆向防火墙，λ表示系统参数.$P_{\mathcal{A}}^*,P_{\mathcal{B}}^*$代表被敌手控制的协议参与方，$\sigma_{P_{\mathcal{B}}^*}$为协议运行后$P_{\mathcal{B}}^*$的状态，敌手在游戏 *LEAK* 中的优势被定义为

1）对于任意改变协议功能$\mathcal{F}$的 PPT 攻击者$P_{\mathcal{A}}^*$，其优势$Adv_{\mathcal{A},\mathcal{W}}^{LEAK}(\lambda)$是可以忽略不计的，则称 CRF 对于协议参与方 P 具有强抵抗渗透性.

2）对于任意不改变协议功能$\mathcal{F}$的 PPT 攻击者$\hat{P}_{\mathcal{A}}$，其优势$Adv_{\mathcal{A},\mathcal{W}}^{LEAK}(\lambda)$是可以忽略不计的，则称 CRF 对于协议参与方 P 具有弱抵抗渗透性.

与定义 2 相同，本文主要讨论 CRF 对于不破坏协议功能的 ASA 安全性.

2 方案定义

本节我们给出了本文方案的系统模型 SMP-IBEET-CRF、形式化定义，并且通过考虑 3 种不同的敌手来定义安全模型.

2.1 系统模型

系统模型如图 2 所示，在 SM9-IBEET-CRF 中，存在 5 种实体.

1）数据上传者. 生成密文并将其上传到云服务器的实体.

2）数据接收者. 可以从云服务器下载密文并解密，或者可以委托云服务器执行等式测试的实体.

3）云服务器. 存储密文，在收到用户请求后可以

proc. $LEAK(\mathcal{P},P,J,\mathcal{W},\lambda)$
$(P_{\mathcal{A}}^*,P_{\mathcal{B}}^*,I)\leftarrow\mathcal{A}(1^{\lambda})$;
$b\xleftarrow{\$}\{0,1\}$;
if b=1, $P^*\leftarrow\mathcal{W}^{\circ}P_{\mathcal{A}}^*$;
else $P^*\leftarrow\mathcal{W}^{\circ}P$;
$\mathcal{T}^*\leftarrow\mathcal{P}_{P}\Rightarrow P^*J\Rightarrow P_{\mathcal{B}}^*(I)$;
$b\leftarrow\mathcal{A}\left(\mathcal{T}^*,\sigma_{P_{\mathcal{B}}^*}\right)$;
output $(b^*=b)$.

Fig. 1 Illustration of *LEAK*

图 1 *LEAK* 示意图

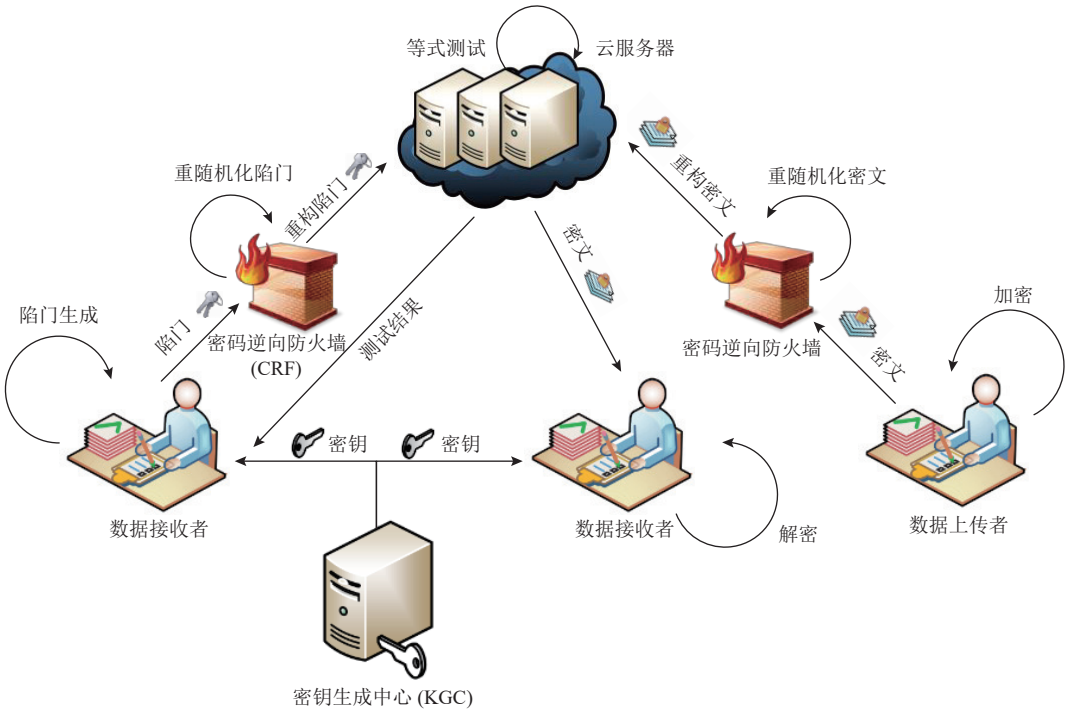


Fig. 2 Illustration of SM9-IBEET-CRF

图 2 SM9-IBEET-CRF 示意图

执行等式测试但无法解密的实体.

4) 密钥生成中心(KGC). 为用户秘密地生成并且分配密钥的实体.

5) 密码逆向防火墙(CRF). 部署在用户(数据上传者和数据接收者)与云服务器的上行信道中, 重随机化用户密文与陷门, 再发送给云服务器的实体.

KGC 初始化系统, 根据用户的身份来生成其私钥, 并秘密传输给用户. 数据上传者计算接收者的公钥来生成密文, 然后上传到云服务器. 密文在上传过程中会受到 CRF 的重随机化处理, 而数据上传者并不知道有这个过程. 在任何时候, 数据接收者都可以从云服务器下载密文, 并使用 KGC 生成的私钥来解密数据. 当接收者想要测试其存储在云服务器的密文时, 可以利用自己的私钥计算出陷门并将其发送给云服务器进行测试, 但是并没有给云服务器提供解密的能力. 陷门在上传过程中会受到 CRF 的重随机化处理, 而数据接收者不会知道有这个过程.

2.2 形式化定义

SM9-IBEET-CRF 方案由 8 个算法组成.

1) 系统建立 *Setup*. 输入安全参数 k, KGC 运行该算法并生成系统公开参数 *params* 和系统主密钥, 包括消息空间.

2) 私钥提取 *KeyExtract*. 输入系统公开参数 *params*、用户 *ID* 以及主私钥 s, KGC 运行该算法生成用户身份所对应的私钥 d.

3) 陷门生成 *Trapdoor*. 输入用户 *ID* 以及用户私钥 d, 输出对应的陷门 td.

4) 加密 *Encrypt*. 输入明文 M 和用户 *ID*, 输出密文 C.

5) 重随机化密文 *ReEncrypt*. 输入密文 C, CRF 运行该算法输出对应的重随机化密文 C'.

6) 密文解密 *Decrypt*. 输入密文 C、用户 *ID* 和用户私钥 d, 解密输出明文 M.

7) 重随机化陷门 *ReTrapdoor*. 输入陷门 td, CRF 运行该算法输出对应的重随机化陷门 td'.

8) 等式测试 *Test*. 分别输入 ID_A 对应的密文 C_A 和陷门 td_A, ID_B 对应的密文 C_B 和陷门 td_B, 云服务器执行等式测试, 若 C_A 和 C_B 的内容为相同的明文, 则输出 1, 否则输出 0.

根据 IBEET 的安全模型, SM9-IBEET 需要考虑 3 种类型的敌手.

1) Ⅰ型敌手. 这类敌手没有目标用户的陷门, 不能执行等式测试, 其目的是在 2 个挑战密文中做区分. 我们针对这种类型的敌手定义了 IBE-IND-CCA 安全游戏 Game 1.

安全游戏 Game 1 中让 $\mathcal{A}_1$ 表示Ⅰ型敌手, 挑战者 C 与 $\mathcal{A}_1$ 按如下顺序进行游戏:

① 系统建立 *Setup*.挑战者 C 执行系统建立 *Setup*

算法生成系统参数 *params* 和主私钥对.C保存主私钥对并将 *params* 发送给$\mathcal{A}_1$.

② 阶段 1. $\mathcal{A}_1$可以自适应地执行查询：

i)公钥查询. 当接收到身份为 ID_i 的公钥询问时，C通过运用主私钥对计算，生成公钥Q_i并发送给$\mathcal{A}_1$.

ii)私钥查询. 当接收到身份为 ID_i 的公钥询问时，C通过执行私钥提取算法 *KeyExtract* 生成私钥d_i并发送给$\mathcal{A}_1$.

iii)解密查询. 当接收到身份为 ID_i 以及密文C的密钥解密询问时，C执行密文解密算法 *Decrypt* 生成明文 M 并返回给$\mathcal{A}_1$.

③挑战. 敌手$\mathcal{A}_1$将身份 ID^*以及消息M_0^*, M_1^*(二者长度相同)发送给C，且在阶段 1，ID^*对应的私钥没有被询问到，则C随机选取$\rho \in \{0,1\}$，并且计算$C^* = Encrypt(M_\rho, d^*, ID^*)$并发送给$\mathcal{A}_1$.

④阶段 2. $\mathcal{A}_1$像在阶段 1 一样发出询问，但是 ID^*对应的私钥以及密文C^*不可以被询问到.

⑤ 猜测. $\mathcal{A}_1$输出$\rho' \in \{0,1\}$.

定义 4. 如果对于任意多项式时间攻击者$\mathcal{A}_1$，在 IND-CCA 游戏中的优势$Adv^{\text{IBE-IND-CCA}}_{\text{SM9-IBEET,Type-I}}(\mathcal{A}_1) = \left|2Pr\left[\rho = \rho'\right] - 1\right|$都是可忽略的，则 SM9-IBEET 方案是满足 IBE-IND-CCA 安全的.

2)Ⅱ型敌手. 这类敌手拥有目标用户密文的陷门，因此可以执行挑战密文的等式测试，其目的是为了揭示挑战密文对应的消息. 我们针对这种类型的敌手定义了 IBE-OW-CCA 安全游戏 Game 2.

安全游戏 Game 2 中让$\mathcal{A}_2$表示Ⅱ型敌手，挑战者C与$\mathcal{A}_2$按如下顺序进行游戏：

①系统建立 *Setup*. 挑战者C执行系统建立 *Setup* 算法生成系统参数 *params* 和主私钥对. C保存主私钥对并将 *params* 发送给$\mathcal{A}_2$.

②阶段 1.$\mathcal{A}_2$可以自适应地执行查询：

i)公钥查询. 当接收到身份为 ID_i 的公钥询问时，C通过运用主私钥对计算，生成公钥Q_i并发送给$\mathcal{A}_2$.

ii)私钥查询. 当接收到身份为 ID_i 的公钥询问时，C通过执行私钥提取算法 *KeyExtract* 生成私钥d_i并发送给$\mathcal{A}_2$.

iii)陷门查询. 当接收到身份为 ID_i 的陷门询问时，C通过执行陷门生成算法 *Trapdoor* 生成陷门td_i并发送给$\mathcal{A}_2$.

iv)解密查询. 当接收受到身份为 ID_i 以及密文C的密钥解密询问时，C执行密文解密算法 *Decrypt* 生成明文 M 并返回给$\mathcal{A}_2$.

③挑战. 敌手$\mathcal{A}_2$将身份 ID^*发送给C，且在阶段 1，ID^*对应的私钥没有被询问到. 则C随机选取消息M^*，计算$C^* = Encrypt(M^*, d^*, ID^*)$并发送给$\mathcal{A}_2$.

④阶段 2. $\mathcal{A}_2$像在阶段 1 一样发出询问，但是 ID^*对应的私钥、陷门以及密文C^*不可以被询问到.

⑤猜测. $\mathcal{A}_2$输出M'.

定义 5. 如果对于任意多项式时间攻击者$\mathcal{A}_2$，在 IBE-OW-CCA 游戏中的优势$Adv^{\text{IBE-OW-CCA}}_{\text{SM9-IBEET,Type-II}}(\mathcal{A}_2) = \left|Pr\left[M^* = M'\right]\right|$都是可忽略的，则 SM9-IBEET 方案是满足 IBE-OW-CCA 安全的.

为证明 CRF 的部署带来的 ASA 安全性，SM9-IBEET-CRF 还需考虑Ⅲ型敌手.

3)Ⅲ型敌手. 其具备 ASA 能力，在保持算法功能不变的前提下，可以替换除了 CRF 重随机化以外的算法，然后对系统发起攻击. 针对这种类型的敌手，我们可以证明 CRF 的部署没有改变原 SM9-IBEET 的功能与安全性，同时增强了 ASA 安全性. 基于此，我们定义了 ASA 安全游戏 Game 3.

安全游戏 Game 3 中让$\mathcal{A}_3$表示Ⅲ型敌手，挑战者C与$\mathcal{A}_3$按如下顺序进行游戏.

① 篡改阶段.$\mathcal{A}_3$选择一些篡改的算法 *Setup**, *KeyExtract**, *Encrypt**, *Decrypt**, *Trapdoor**, *Test**发送给C，C收到后用篡改算法来替换自己的原始算法.

②系统建立 *Setup*.挑战者C执行系统建立 *Setup** 算法生成系统参数 *params* 和主私钥对.C保存主私钥对并将 *params* 发送给$\mathcal{A}_3$.

③阶段 1.$\mathcal{A}_3$可以自适应地执行查询：

i) 公钥查询. 当接受到身份为 ID_i 的公钥询问时，C通过运用主私钥对计算，生成公钥Q_i并发送给$\mathcal{A}_3$.

ii)私钥查询. 当接受到身份为 ID_i 的公钥询问时，C执行私钥提取算法 *KeyExtract** 生成私钥 d_i 并发送给$\mathcal{A}_3$.

iii) 陷门查询. 当接受到身份为 ID_i 的陷门询问时，C执行陷门生成算法 *Trapdoor** 生成陷门 td_i，然后运行陷门重随机化算法 *ReTrapdoor* 生成陷门td_i的重随机化陷门td_i'并发送给$\mathcal{A}_3$.

iv) 解密查询. 当接受到身份为 ID_i 以及密文C的密钥解密询问时，C执行密文解密算法 *Decrypt** 生成明文并返回给$\mathcal{A}_3$.

④挑战. 敌手$\mathcal{A}_3$将身份 ID^*以及消息M_0^*, M_1^*(二者长度相同)发送给C，且在阶段 1，ID^*对应的私钥没有被询问到，则C随机选取$\rho \in \{0,1\}$，并且计算$C^* = Encrypt^*(M_\rho, d^*, ID^*)$，然后再计算$C^{*'} = ReEncrypt(C^*)$并发送给$\mathcal{A}_3$.

⑤阶段 2. $\mathcal{A}_3$像在阶段 1 一样发出询问，但是 ID^* 对应的私钥以及密文$C^*,C^{*'}$不可以被询问到.

⑥猜测. $\mathcal{A}_3$输出$\rho' \in \{0,1\}$.

定义 6. 如果对于任意多项式时间攻击者$\mathcal{A}_3$，在 ASA 游戏中的优势$Adv_{\text{SM9ET-CRF}}^{\text{ASA,Type-III}}(\mathcal{A}_3)=\left|2Pr\left[\rho=\rho'\right]-1\right|$都是可忽略的，则 SM9-IBEET-CRF 方案是满足 ASA 安全的.

3 具体方案

本文方案由 8 个算法组成，具体构造过程介绍如下.

1）系统建立 *Setup*.

① 初始化系统，输出系统参数$params=\left\langle G_1,G_2,G_T,e,P_1,P_2,P_{\text{pub1}},P_{\text{pub2}},KDF,MAC,EUC\right\rangle$. 其中 e为双线性对映射$e:G_1\times G_2\to G_T$，G_1，G_2的阶均为N，P_1为G_1的生成元，P_2为G_2的生成元. 消息空间$\mathcal{M}\in\{0,1\}^*$，用户的身份$id\in\{0,1\}^*$均为比特串.

② KGC 随机选取$s,s'\in[1,N-1]$作为主私钥对(s,s')，并计算主公钥$P_{\text{pub1}}=[s]P_1$，$P_{\text{pub2}}=[s']P_1$.

③ 获取 KGC 公布的 5 个哈希函数：$H_1:\{0,1\}^*\to\mathbb{Z}_N^*$，$H_2:G_T\to G_2$，$H_3:G_1\to\{0,1\}^*$，$H_4:\{0,1\}^*\to G_2$，$H_5:G_T\to\{0,1\}^*$.

④ 使用 SM9 规定的密钥派生函数$KDF(Z,klen)$，输入比特串Z、非负整数$klen$，输出长度为$klen$的密钥数据比特串K.

⑤消息认证码函数$MAC(K_2,Z)$. 输入为比特长度K_2_len的密钥K_2，比特串消息Z. 其作用是防止消息数据Z被非法篡改.

⑥ 拓展欧几里得函数$EUC(r)$. 输入$r\in[1,N-1]$，运行拓展欧几里得算法计算输出 r 的逆元.

2）私钥提取 *KeyExtract*. 输入系统公开参数 *params*、用户身份ID_A和主私钥对 s，KGC 按①~③方式生成ID_A的私钥d_A：

① 在有限域F_N上计算$t_1=H_1(ID_A)+s$，若$t_1=0$则需要重新产生主私钥；

② 否则，计算$t_2=st_1^{-1}$，$t_2'=s't_1^{-1}$；

③ 然后计算$d_A=(d_{A1},d_{A2})$，此处的(s,s')是主私钥对，即

$$d_{A1}=[t_2]P_2=[s(H_1(ID_A)+s)^{-1}]P_2,$$

$$d_{A2}=[t_2']P_2=[s'(H_1(ID_A)+s)^{-1}]P_2.$$

3）陷门生成 *Trapdoor*. 输入用户身份ID_A，私钥d_A. 输出陷门$td_A=[t_2']P_2$.

4）加密 *Encrypt*. 输入系统公开参数 *params*、用户身份ID_A，运算生成用户公钥Q_A. 对于消息长度为 *mlen* 比特的比特串$M\in\{0,1\}^*$，*mlen* 为密钥K_1的比特长度，K_2_len为$MAC(K_2,Z)$中密钥K_2的比特长度，过程运算有：

① $Q_A=[H_1(ID_A)]P_1+P_{\text{pub1}}$；

② 随机选取$r_1,r_2\in[1,N-1]$；

③ $C_1=[r_1]Q_A$；

④$g=e(P_{\text{pub1}},P_2)$；

⑤ $w=g^{r_1}$；

⑥ 计算K_1,K_2：

i）$klen=mlen+K_2_len$；

ii）$K_1\|K_2=KDF(H_3(C_1)\|H_5(w)\|ID_A,klen)$；

⑦ $C_2=M\oplus K_1$；

⑧ $C_3=MAC(K_2,C_2)$；

⑨ $C_4=[r_2]H_4(M)H_2(e(P_{\text{pub2}},[r_1]P_2)^{r_2})$；

⑩ $C_5=[r_2]P_1$；

⑪ $C_6=[r_1r_2]Q_A$；

⑫ 输出$C=(C_1,C_2,C_3,C_4,C_5,C_6)$作为密文.

5）重随机化密文 *ReEncrypt*. CRF 收到密文$C=(C_1,C_2,C_3,C_4,C_5,C_6)$后，随机选取$r_3\in[1,N-1]$，然后计算$C$的重随机化密文$C'=(C_1,C_2,C_3,C_4,C_5,[r_3]C_6)$，并发送给云服务器.

6）解密 *Decrypt*. 输入$C'=(C_1,C_2,C_3,C_4,C_5,C_6')$，私钥$d_A=(d_{A1},d_{A2})$和用户身份$ID_A$.

① 验证$C_1\in G_1$，若不成立则无法解密；

② $w'=e(C_1,d_{A1})$；

③ $klen=mlen+K_2_len$；

④ $K_1'\|K_2'=KDF(H_3(C_1)\|H_5(w')\|ID_A,klen)$；

⑤ $M'=C_2\oplus K_1'$；

⑥ 若$C_3\neq MAC(K_2',C_2)$，则解密失败，密文完整性有误；

⑦ 输出M'作为消息的明文.

7）重随机化陷门 *ReTrapdoor*. 输入陷门td，CRF 运行$EUC(r_3)$生成r_3的逆元$r_4\in[1,N-1]$，计算td的重随机化陷门$td'=[r_4]td=[r_4t_2']P_2$.

8）等式测试 *Test*. 输入 2 个用户的密文$C_\alpha'=(C_{1,\alpha},C_{2,\alpha},C_{3,\alpha},C_{4,\alpha},C_{5,\alpha},C_{6,\alpha}')$，$C_\beta'=(C_{1,\beta},C_{2,\beta},C_{3,\beta},C_{4,\beta},C_{5,\beta},C_{6,\beta}')$和 2 个陷门$td_\alpha'$，$td_\beta'$，然后按 3 个步骤进行测试.

①$X_\alpha=\dfrac{C_{4,\alpha}}{H_2\left(e\left(C'_{6,\alpha},td'_\alpha\right)\right)}$；

②$X_\beta=\dfrac{C_{4,\beta}}{H_2\left(e\left(C'_{6,\beta},td'_\beta\right)\right)}$；

③ 若$e(C_{\alpha,5},X_\beta)=e(C_{\beta,5},X_\alpha)$，则$M_\alpha=M_\beta$.

4 安全性分析

4.1 SM9-IBEET正确性分析

本节首先对SM9-IBEET进行正确性分析.

1）验证密文解密过程的正确性

采用用户私钥d以及密文消息$C=(C_1,C_2,C_3,C_4,C_5,C_6)$来验证:

$$\begin{aligned}M'_{\alpha}=&C_{2,\alpha}\oplus KDF(H_3(C_{1,\alpha})\|H_5(e(C_{1,\alpha},d_{1,\beta}))\|ID_{\beta},klen)=\\&C_{2,\alpha}\oplus KDF(H_3(C_{1,\alpha})\|H_5(w_{1,\alpha})\|ID_{\beta},klen)=\\&C_{2,\alpha}\oplus K'_{1,\beta},\end{aligned}$$

其中$K'_{1,\beta}$为KDF函数结果左边的$mlen$比特.

2）计算消息认证码函数

$u=MAC(K'_{2,\beta},C_{2,\alpha})$（$K'_{2,\beta}$为$KDF$函数结果右边的$K_2_len$比特）.

若$u=C_{3,\alpha}$，则消息认证完整性的结果通过，解密结果正确，输出明文M'_{α}.

3）验证等式测试计算结果的正确性

第1层计算的安全性如下所示，采用用户的陷门以及部分密文验证:

$$\begin{aligned}X_{\alpha}=&\frac{C_{4,\alpha}}{H_2\left(e\left(C_{6,\alpha},td_{\alpha}\right)\right)}=\\&\frac{\left[r_{2,\alpha}\right]H_4(M)H_2\left(e\left(P_{\text{pub2}},\left[r_{1,\alpha}\right]P_2\right)^{r_{2,\alpha}}\right)}{H_2\left(e\left(\left[r_{1,\alpha}\right]\left[r_{2,\alpha}\right]\left[t_{1,\alpha}\right]P_1,\left[t'_{2,\alpha}\right]P_2\right)\right)}=\\&[r_{2,\alpha}]H_4(M_{\alpha}),\end{aligned}$$

$$\begin{aligned}X_{\beta}=&\frac{C_{4,\beta}}{H_2\left(e\left(C_{6,\beta},td_{\beta}\right)\right)}=\\&\frac{\left[r_{2,\beta}\right]H_4(M)H_2\left(e\left(P_{\text{pub2}},\left[r_{1,\beta}\right]P_2\right)^{r_{2,\beta}}\right)}{H_2\left(e\left(\left[r_{1,\beta}\right]\left[r_{2,\beta}\right]\left[t_{1,\beta}\right]P_1,\left[t'_{2,\beta}\right]P_2\right)\right)}=\\&[r_{2,\beta}]H_4(M_{\beta}).\end{aligned}$$

第2层计算的正确性分析如下所示，带入第1层中计算的中间结果:

$$\begin{aligned}e(C_{5,\alpha},X_{\beta})=&e([r_{2,\alpha}]P_2,[r_{2,\beta}]H_4(M_{\beta}))=\\&e(P_2,H_4(M_{\beta}))^{r_{2,\alpha}r_{2,\beta}},\\e(C_{5,\beta},X_{\alpha})=&e([r_{2,\beta}]P_2,[r_{2,\alpha}]H_4(M_{\alpha}))=\\&e(P_2,H_4(M_{\alpha}))^{r_{2,\alpha}r_{2,\beta}}.\end{aligned}$$

若$M_{\alpha}=M_{\beta}$，则等式测试的结果成立.

4.2 SM9-IBEET安全性证明

定理1. 假定嵌入的BDH困难问题是不可破解的猜想成立，则表示本文提出的支持等式测试的SM9算法是IBE-IND-CCA安全的.

证明. 假设存在无法获取目标用户陷门且不能任意执行等式测试的Ⅰ型敌手$\mathcal{A}_1$，其攻击目的是破坏所提方案的语义安全，也即在安全游戏中对挑战密文进行区分. 如果敌手$\mathcal{A}_1$可以成功破坏本文方案，则存在挑战者C能够以不可忽略的优势解决BDH困难问题. 给定$(P_1,P_2,[a]P_1,[a]P_2,[b]P_1,[c]P_1)$，其中$a,b,c\in\mathbb{Z}_N^*$，$C$的目标是计算出$e(P_1,P_2)^{abc}$. C与$\mathcal{A}_1$的挑战过程有7个.

1）初始化.

C随机选取$\kappa\in\{1,2,\cdots,q_{H_1}\}$，$N_{\kappa}\in\mathbb{Z}_N^*$以及$\tau_1,\tau_2,\cdots,\tau_{\kappa-1},\tau_{\kappa+1},\cdots,\tau_N\in\mathbb{Z}_N^*$，$q_{H_1}$代表的是查询随机预言机$\mathcal{H}_1$的次数，对$i=1,2,\cdots,\kappa-1,\kappa+1,\cdots,N$，计算$N_i=N_{\kappa}-\tau_i$并保存. P_1,P_2分别为群G_1,G_2的生成元，通过使用文献[30]的方法，C随机选取$\gamma\in[1,N-1]$，对$i\in\{1,2,\cdots,N\}\backslash\{\kappa\}$，$C$可以获得$N-1$个数值对$(\tau_i,(1/\gamma-\tau_i))$，计算$P_{\text{pub1}}=(\gamma-N_{\kappa})P_1$，令$P_{\text{pub2}}=aP_1$. 得到公共参数$params=\langle G_1,G_2,G_T,e,P_1,P_2,P_{\text{pub1}},P_{\text{pub2}},MAC\rangle$，并将其发送给$\mathcal{A}_1$.

C保存表格$\mathcal{L}_{\mathcal{H}_1},\mathcal{L}_{\mathcal{H}_2}\mathcal{L}_{\mathcal{H}_3},\mathcal{L}_{\mathcal{H}_4},\mathcal{L}_{\mathcal{H}_5},\mathcal{L}_{\mathcal{KDF}}$，初始化内容为空，用来模拟随机预言机$\langle\mathcal{H}_1,\mathcal{H}_2,\mathcal{H}_3,\mathcal{H}_4,\mathcal{H}_5,\mathcal{KDF}\rangle$. 设置空列表$\mathcal{L}_K$来保存公钥查询的结果.

2）敌手$\mathcal{A}_1$向C提出6个询问.

① $\mathcal{H}_1$-*query*. 在任何时刻$\mathcal{A}_1$可以询问随机预言机$\mathcal{H}_1$，为了回答这些询问，C保存表格$\mathcal{L}_{\mathcal{H}_1}$用来存取元组$\langle ID_i,N_i\rangle$，当接收到身份为$ID_i$的$\mathcal{H}_1$查询时，$C$查找$ID_i$对应的数值$N_i$，用$H_1(ID_i)=N_i$返回给$\mathcal{A}_1$，并将$\langle ID_i,N_i\rangle$添加到$\mathcal{L}_{\mathcal{H}_1}$中.

② $\mathcal{H}_2$-*query*. 在任何时刻$\mathcal{A}_1$可以询问随机预言机$\mathcal{H}_2$，为了回答这些询问，C保存表格$\mathcal{L}_{\mathcal{H}_2}$用来存取元组$\langle\sigma,\phi\rangle$，$C$按2个步骤回应:

i）如果询问的ID_i已经出现在$\mathcal{L}_{\mathcal{H}_2}$的元组$\langle\sigma,\phi\rangle$中，$C$用$\phi$来回复.

ii）否则，C随机选取$\sigma\in G_T,\phi\in G_1$，并将元组$\langle\sigma,\phi\rangle$插入$\mathcal{L}_{\mathcal{H}_2}$中，然后用$\phi$来回复$\mathcal{A}_1$.

③ $\mathcal{H}_3$-*query*. 在任何时刻$\mathcal{A}_1$可以询问随机预言机$\mathcal{H}_3$，为了回答这些询问，C保存表格$\mathcal{L}_{\mathcal{H}_3}$用来存取元组$\langle C_1,h_3\rangle$，如果询问的$C_1$存在$\mathcal{L}_{\mathcal{H}_3}$中，返回$h_3$给$\mathcal{A}_1$；否则，$C$随机选取$h_3\in\{0,1\}^*$并添加表项$\langle C_1,h_3\rangle$到$\mathcal{L}_{\mathcal{H}_3}$中，并返回$h_3$给$\mathcal{A}_1$.

④ $\mathcal{H}_4$-*query*. 在任何时刻$\mathcal{A}_1$可以询问随机预言机$\mathcal{H}_4$，为了回答这些询问，C保存表格$\mathcal{L}_{\mathcal{H}_4}$用来存取元组$\langle M,h_4\rangle$，如果询问的$M$存在$\mathcal{L}_{\mathcal{H}_4}$中，返回$h_4$给$\mathcal{A}_1$；否则，$C$随机选取$h_4\in G_1$并添加表项$\langle M,h_4\rangle$到$\mathcal{L}_{\mathcal{H}_4}$中，返回$h_4$给$\mathcal{A}_1$.

⑤ $\mathcal{H}_5$-*query*. 在任何时刻$\mathcal{A}_1$可以询问随机预言

机$\mathcal{H}_5$, 为了回答这些询问, C保存表格$\mathcal{L}_{\mathcal{H}_5}$用来存取元组$\langle w,h_5\rangle$, 如果询问的$w$存在$\mathcal{L}_{\mathcal{H}_5}$中, 返回$h_5$给$\mathcal{A}_1$; 否则, C随机选取$h_5\in\{0,1\}^*$并添加表项$\langle w,h_5\rangle$到$\mathcal{L}_{\mathcal{H}_5}$中, 返回$h_5$给$\mathcal{A}_1$.

⑥ $\mathcal{KDF}$-*query*. 在任何时刻$\mathcal{A}_1$可以询问随机预言机$\mathcal{KDF}$, 为了回答这些询问, C保存表格$\mathcal{L}_{\mathcal{KDF}}$用来存取元组$\langle Z,K\rangle$, 如果询问的$Z$存在于$\mathcal{L}_{\mathcal{KDF}}$中, 返回$K$给$\mathcal{A}_1$; 否则, C随机选取$K\in\{0,1\}^*$并添加表项$\langle Z,K\rangle$到$\mathcal{L}_{\mathcal{KDF}}$中, 返回$K$给$\mathcal{A}_1$.

3) 公钥查询. 当接受到身份为ID_i的公钥询问时, C按照如下方式回应: 检查列表$\mathcal{L}_{\mathcal{H}_1}$, 如果$i=\kappa$, C放弃, 否则得到$H_1(ID_i)=N_i$; 计算用户公钥$Q_i=[H_1(ID_i)]P_1+P_{\text{pub1}}=N_iP_1+(\gamma-N_\kappa)P_1=(\gamma-\tau_i)P_1$, 并将$Q_i$发送给$\mathcal{A}_1$.

4) 私钥查询. 当接受到身份为ID_i的私钥询问时, C按照如下方式回应: 检查列表$\mathcal{L}_{\mathcal{H}_1}$, 如果$i=\kappa$, C放弃; 否则得到$H_1(ID_i)=N_i$, 计算用户私钥为

$$d_{i,1}=[s(H_1(ID_i)+s)^{-1}]P_2=[(\gamma-N_\kappa)/(\gamma-\tau_i)]P_2=[1-(N_k-\tau_i)/(\gamma-\tau_i)]P_2,$$

$$d_{i,2}=[s'(H_1(ID_i)+s)^{-1}]P_2=[a/(\gamma-\tau_i)]P_2.$$

并将$d_{i,1}$和$d_{i,2}$发送给$\mathcal{A}_1$.

5) 解密查询. 当接受到身份为ID_i以及密文$C=(C_1,C_2,C_3,C_4,C_5,C_6)$的解密询问时, 如果$i\neq\kappa$, C计算$d_{i,1}=[(\gamma-N_\kappa)/(\gamma-\tau_i)]P_2$; 如果$i=\kappa$, 计算$d_{\kappa,1}=[1-N_\kappa/\gamma]P_2$. 计算$w'=e(C_1,d_{i,1})$, 查找$C_1$在$\mathcal{L}_{\mathcal{H}_3}$对应的表项$\langle C_1,h_3\rangle$, w'在$\mathcal{L}_{\mathcal{H}_5}$对应的表项$\langle w',h_5\rangle$, 得到$Z=h_3\|h_5\|ID_i$, 查找$Z$在$\mathcal{L}_{\mathcal{KDF}}$中对应的表项$\langle Z,K\rangle$, 拆分$K=K_1'\|K_2'$. 计算$M'=C_2\oplus K_1'$, 验证$C_3=MAC(K_2',C_2)$是否成立, 如果成立, 则返回明文; 如果不成立, 则C输出$\perp$.

6) 挑战. 敌手$\mathcal{A}_1$将身份ID^*以及消息M_0^*,M_1^*(二者长度相同)发送给C, 如果$ID^*\neq ID_\kappa$, C放弃游戏; 如果$ID^*=ID_\kappa$, C随机选取$\rho\in\{0,1\}$, $C_2^*\in\{0,1\}^*$, $C_3^*\in\{0,1\}^*$, $C_4^*\in G_1$, $C_6^*\in G_1$, 并计算$C_1^*=(\gamma-\tau_i)bP_1$, $C_5^*=cP_1$后将$C^*=(C_1^*,C_2^*,C_3^*,C_4^*,C_5^*,C_6^*)$发送给$\mathcal{A}_1$作为挑战密文.

7) 猜测. $\mathcal{A}_1$输出$\rho'\in\{0,1\}$. C从$\mathcal{L}_{\mathcal{H}_2}$随机选取一个元组$\langle\sigma^*,\phi^*\rangle$, 并输出$\sigma^*=e(P_1,P_2)^{abc}$作为BDH实例的解. 证毕.

定理 2. 假定嵌入的BDH困难问题是不可破解的猜想成立, 则表示我们提出的支持等式测试的国密SM9算法方案是IBE-OW-CCA安全的.

证明. 假定存在可以获取目标用户陷门且可以执行等式测试的Ⅱ型敌手$\mathcal{A}_2$, 其攻击目的是为了破坏所提方案的机密性, 也即揭示挑战密文对应的消息. 如果敌手$\mathcal{A}_2$可以成功破坏所提方案, 则存在挑战者C能够以不可忽略的优势解决BDH困难问题. 给定$(P_1,P_2,[a]P_1,[a]P_2,[b]P_1,[c]P_1)$, 其中$a,b,c\in\mathbb{Z}_N^*$, C的目标是计算出$e(P_1,P_2)^{abc}$. C与$\mathcal{A}_2$的挑战过程有8个.

1) 初始化. C随机选取$\kappa\in\{1,2,\cdots,q_{H_1}\}$, $N_\kappa\in\mathbb{Z}_N^*$以及$\tau_1,\tau_2,\cdots,\tau_{\kappa-1},\tau_{\kappa+1},\cdots,\tau_N\in\mathbb{Z}_N^*$, q_{H_1}代表的是查询随机预言机$\mathcal{H}_1$的次数, 对$i=1,2,\cdots,\kappa-1,\kappa+1,\cdots,N$, 计算$N_i=N_\kappa-\tau_i$并保存. P_1,P_2分别为群G_1,G_2的生成元, 通过使用文献[30]的方法, C随机选取$\gamma\in[1,N-1]$, 对$i\in\{1,2,\cdots,N\}\backslash\{\kappa\}$, C可以获得$N-1$个数值对$(\tau_i,(1/\gamma-\tau_i))$, 计算$P_{\text{pub1}}=(\gamma-N_\kappa)P_1$, 令$P_{\text{pub2}}=aP_1$. 得到公共参数$params=\langle G_1,G_2,G_T,e,P_1,P_2,P_{\text{pub1}},P_{\text{pub2}},MAC\rangle$, 并将其发送给$\mathcal{A}_2$.

C保存表格$\mathcal{L}_{\mathcal{H}_1},\mathcal{L}_{\mathcal{H}_2}\mathcal{L}_{\mathcal{H}_3},\mathcal{L}_{\mathcal{H}_4},\mathcal{L}_{\mathcal{H}_5},\mathcal{L}_{\mathcal{KDF}}$, 初始化内容为空, 用来模拟随机预言机$\langle\mathcal{H}_1,\mathcal{H}_2,\mathcal{H}_3,\mathcal{H}_4,\mathcal{H}_5,\mathcal{KDF}\rangle$. 设置空列表$\mathcal{L}_K$来保存公钥查询的结果.

2) 敌手$\mathcal{A}_2$向C提出6点询问.

① $\mathcal{H}_1$-*query*. 在任何时刻$\mathcal{A}_2$可以询问随机预言机$\mathcal{H}_1$, 为了回答这些询问, C保存表格$\mathcal{L}_{\mathcal{H}_1}$用来存取元组$\langle ID_i,N_i\rangle$, 当接收到身份为$ID_i$的$\mathcal{H}_1$查询时, C查找ID_i对应的数值N_i, 用$H_1(ID_i)=N_i$返回给$\mathcal{A}_2$, 并将$\langle ID_i,N_i\rangle$添加到$\mathcal{L}_{\mathcal{H}_1}$中.

② $\mathcal{H}_2$-*query*. 在任何时刻$\mathcal{A}_2$可以询问随机预言机$\mathcal{H}_2$, 为了回答这些询问, C保存表格$\mathcal{L}_{\mathcal{H}_2}$用来存取元组$\langle\sigma,\phi\rangle$, C按2个步骤回应:

i) 如果询问的ID_i已经出现在$\mathcal{L}_{\mathcal{H}_2}$的元组$\langle\sigma,\phi\rangle$中, C用ϕ来回复.

ii) 否则, C随机选取$\sigma\in G_T,\phi\in G_1$, 并将元组$\langle\sigma,\phi\rangle$插入$\mathcal{L}_{\mathcal{H}_2}$中, 然后用$\phi$来回复$\mathcal{A}_2$.

③ $\mathcal{H}_3$-*query*. 在任何时刻$\mathcal{A}_2$可以询问随机预言机$\mathcal{H}_3$, 为了回答这些询问, C保存表格$\mathcal{L}_{\mathcal{H}_3}$用来存取元组$\langle C_1,h_3\rangle$, 如果询问的$C_1$存在$\mathcal{L}_{\mathcal{H}_3}$中, 返回$h_3$给$\mathcal{A}_2$; 否则, C随机选取$h_3\in\{0,1\}^*$并添加表项$\langle C_1,h_3\rangle$到$\mathcal{L}_{\mathcal{H}_3}$中, 并返回$h_3$给$\mathcal{A}_2$.

④ $\mathcal{H}_4$-*query*. 在任何时刻$\mathcal{A}_2$可以询问随机预言机$\mathcal{H}_4$, 为了回答这些询问, C保存表格$\mathcal{L}_{\mathcal{H}_4}$用来存取元组$\langle M,h_4\rangle$, 如果询问的$M$存在$\mathcal{L}_{\mathcal{H}_4}$中, 返回$h_4$给$\mathcal{A}_2$; 否则, C随机选取$h_4\in G_1$并添加表项$\langle M,h_4\rangle$到$\mathcal{L}_{\mathcal{H}_4}$中, 返回$h_4$给$\mathcal{A}_2$.

⑤ $\mathcal{H}_5$-*query*. 在任何时刻$\mathcal{A}_2$可以询问随机预言机$\mathcal{H}_5$, 为了回答这些询问, C保存表格$\mathcal{L}_{\mathcal{H}_5}$用来存取元组$\langle w,h_5\rangle$, 如果询问的$w$存在$\mathcal{L}_{\mathcal{H}_5}$中, 返回$h_5$给$\mathcal{A}_2$;

否则，C随机选取$h_5\in\{0,1\}^*$并添加表项$\langle w,h_5\rangle$到$\mathcal{L}_{\mathcal{H}_5}$中，返回$h_5$给$\mathcal{A}_2$.

⑥ $\mathcal{KDF}$-*query*. 在任何时刻$\mathcal{A}_2$可以询问随机预言机$\mathcal{KDF}$，为了回答这些询问，C保存表格$\mathcal{L}_{\mathcal{KDF}}$用来存取元组$\langle Z,K\rangle$，如果询问的$Z$存在于$\mathcal{L}_{\mathcal{KDF}}$中，返回$K$给$\mathcal{A}_2$；否则，$C$随机选取$K\in\{0,1\}^*$并添加表项$\langle Z,K\rangle$到$\mathcal{L}_{\mathcal{KDF}}$中，返回$K$给$\mathcal{A}_2$.

3）公钥查询. 当接受到身份为ID_i的公钥询问时，C按照如下方式回应：检查列表$\mathcal{L}_{\mathcal{H}_1}$，如果$i=\kappa$，$C$放弃；否则得到$H_1(ID_i)=N_i$，计算用户公钥

$$Q_i=[H_1(ID_i)]P_1+P_{\text{pub1}}=N_iP_1+(\gamma-N_\kappa)P_1=(\gamma-\tau_i)P_1,$$

并将Q_i发送给$\mathcal{A}_2$.

4）私钥查询. 当接受到身份为ID_i的私钥询问时，C按照如下方式回应：检查列表$\mathcal{L}_{\mathcal{H}_1}$，如果$i=\kappa$，$C$放弃；否则得到$H_1(ID_i)=N_i$，计算用户私钥

$$\begin{aligned}d_{i,1}=&[s(H_1(ID_i)+s)^{-1}]P_2=[(\gamma-N_\kappa)/(\gamma-\tau_i)]P_2=\\&[1-(N_k-\tau_i)/(\gamma-\tau_i)]P_2,\\d_{i,2}=&[s'(H_1(ID_i)+s)^{-1}]P_2=[a/(\gamma-\tau_i)]P_2,\end{aligned}$$

并将$d_{i,1}$和$d_{i,2}$发送给$\mathcal{A}_2$.

5）陷门查询. 当接受到身份为ID_i的陷门询问时，如果$i=\kappa$，C放弃；否则计算

$$d_{i,2}=[s'(H_1(ID_i)+s)^{-1}]P_2=[a/(\gamma-\tau_i)]P_2,$$

并将$d_{i,1}$和$d_{i,2}$发送给$\mathcal{A}_2$.

6）解密查询. 当接受到身份为ID_i以及密文$C=(C_1,C_2,C_3,C_4,C_5,C_6)$的解密询问时，$C$按照如下方式回应：如果$i\neq\kappa$，$C$计算$d_{i,1}=[(\gamma-N_\kappa)/(\gamma-\tau_i)]P_2$；如果$i=\kappa$，计算$d_{\kappa,1}=[1-N_\kappa/\gamma]P_2$、计算$w'=e(C_1,d_{i,1})$，查找$C_1$在$\mathcal{L}_{\mathcal{H}_3}$对应的表项$\langle C_1,h_3\rangle$，$w'$在$\mathcal{L}_{\mathcal{H}_5}$对应的表项$\langle w',h_5\rangle$，得到$Z=h_3\|h_5\|ID_i$，查找$Z$在$\mathcal{L}_{\mathcal{KDF}}$中对应的表项$\langle Z,K\rangle$，拆分$K=K_1'\|K_2'$. 计算$M'=C_2\oplus K_1'$，验证$C_3=MAC(K_2',C_2)$是否成立，如果成立，则返回明文；如果不成立，则$C$输出$\perp$.

7）挑战. 敌手$\mathcal{A}_2$将身份ID^*以及消息$M^*\in\{0,1\}^*$发送给C，如果$ID^*\neq ID_\kappa$，C放弃游戏；如果$ID^*=ID_\kappa$，C随机选取$C_2^*\in\{0,1\}^*$，$C_3^*\in\{0,1\}^*$，$C_4^*\in G_1$，$C_6^*\in G_1$，并计算$C_1^*=(\gamma-\tau_i)bP_1$，$C_5^*=cP_1$后将$C^*=(C_1^*,C_2^*,C_3^*,C_4^*,C_5^*,C_6^*)$发送给$\mathcal{A}_2$作为挑战密文.

8）猜测. $\mathcal{A}_2$输出$M'\in M^*$. C从$\mathcal{L}_{\mathcal{H}_2}$随机选取一个元组$\langle\sigma^*,\phi^*\rangle$，并输出$\sigma^*=e(P_1,P_2)^{abc}$作为BDH实例的解. 证毕.

4.3 ASA安全性

ASA安全性要求SM9-IBEET-CRF可以抵抗Ⅲ型敌手$\mathcal{A}_3$的攻击. 其中敌手$\mathcal{A}_3$具备ASA能力，在保持算法功能不变的前提下，可以替换除了CRF重随机化以外的算法，然后对系统发起攻击.

定理 3. SM9-IBEET-CRF中的CRF具有维持功能性.

证明. 维持功能性要求CRF不影响原协议的功能与安全性，当数据上传者将密文与陷门经由CRF上传至云服务器后，可以得到与原协议相同的功能与安全强度.

数据上传者运行加密算法*Encrypt*生成密文$C=(C_1,C_2,C_3,C_4,C_5,C_6)$并发送给CRF.CRF重随机化密文后得到$C'=(C_1,C_2,C_3,C_4,C_5,[r_3]C_6)$，并将其上传至云服务器.

当数据接收者需要解密时，从云服务器上下载密文，此时不需要经过CRF. 解密过程的正确性通过用户私钥d以及密文消息$C_\alpha'=(C_1,C_2,C_3,C_4,C_5,[r_3]C_6)$来验证：

$$\begin{aligned}M'_\alpha=&C_{2,\alpha}\oplus KDF(H_3(C_{1,\alpha})\|H_5(e(C_{1,\alpha},d_{1,\beta}))\|ID_\beta,klen)=\\&C_{2,\alpha}\oplus KDF(H_3(C_{1,\alpha})\|H_5(w_{1,\alpha})\|ID_\beta,klen)=\\&C_{2,\alpha}\oplus K'_{1,\beta},\end{aligned}$$

其中$K'_{1,\beta}$为*KDF*函数结果左边的*mlen*比特. 接下来计算消息认证码函数：

$u=MAC(K'_{2,\beta},C_{2,\alpha})$（$K'_{2,\beta}$为*KDF*函数结果右边的$K_2_len$比特）.

若$u=C_{3,\alpha}$，则消息认证完整性的结果通过，解密结果正确，输出明文M'_α，解密过程不受影响.

当数据接收者需要执行等式测试时，运行*Trapdoor*算法生成陷门*td*并发送给CRF. CRF重随机化陷门后得到$td'=[r_4]td=[r_4t_2']P_2$，并将td'上传至云服务器. 接下来验证等式测试计算结果的正确性.

1）第1层计算的安全性采用用户的陷门以及部分密文验证：

$$\begin{aligned}X_\alpha=&\frac{C_{4,\alpha}}{H_2\left(e\left(C'_{6,\alpha},td'_\alpha\right)\right)}=\\&\frac{[r_{2,\alpha}]H_4(M)H_2\left(e\left(P_{\text{pub2}},[r_{1,\alpha}]P_2\right)^{r_{2,\alpha}}\right)}{H_2\left(e\left([r_{1,\alpha}][r_{2,\alpha}][r_{3,\alpha}][t_{1,\alpha}]P_1,[r_{4,\alpha}][t'_{2,\alpha}]P_2\right)\right)}=\\&[r_{2,\alpha}]H_4(M_\alpha).\end{aligned}$$

$$\begin{aligned}X_\beta=&\frac{C_{4,\beta}}{H_2\left(e\left(C'_{6,\beta},td'_\beta\right)\right)}=\\&\frac{[r_{2,\beta}]H_4(M)H_2\left(e\left(P_{\text{pub2}},[r_{1,\beta}]P_2\right)^{r_{2,\beta}}\right)}{H_2\left(e\left([r_{1,\beta}][r_{2,\beta}][r_{3,\beta}][t_{1,\beta}]P_1,[r_{4,\beta}][t'_{2,\beta}]P_2\right)\right)}=\\&[r_{2,\beta}]H_4(M_\beta).\end{aligned}$$

2）第2层计算的正确性分析带入第1层中计算的中间结果：

$e(C_{5,\alpha}, X_\beta) = e([r_{2,\alpha}]P_2, [r_{2,\beta}]H_4(M_\beta)) = e(P_2, H_4(M_\beta))^{r_{2,\alpha} r_{2,\beta}}$,

$e(C_{5,\beta}, X_\alpha) = e([r_{2,\beta}]P_2, [r_{2,\alpha}]H_4(M_\alpha)) = e(P_2, H_4(M_\alpha))^{r_{2,\alpha} r_{2,\beta}}$.

若 $M_\alpha = M_\beta$, 则等式测试的结果成立. 证毕.

定理 4. SM9-IBEET-CRF 中的 CRF 具有弱保留安全性和预防泄露性.

证明. 假定存在可以替换除 CRF 重随机化以外算法的Ⅲ型敌手$\mathcal{A}_3$, 其攻击目的是破坏所提方案的机密性, 也即通过篡改原始算法来泄露隐私信息. 如果$\mathcal{A}_3$可以成功破坏所提方案, 使用篡改算法 *Setup**, *KeyExtract**, *Encrypt**, *Decrypt**, *Trapdoor**, *Test**来替换原始算法. 我们则将通过 SM9-IBEET-CRF 的安全性游戏和原 SM9-IBEET 安全性游戏的不可区分性, 来证明 CRF 满足弱保留安全性和预防泄露性. 本文考虑 3 种安全游戏:

1) 安全游戏 Game 1. 与第 2 节中定义的 Game 3 相同.

2) 安全游戏 Game 2. 与 Game 1 的其他部分都相同, 除了在陷门查询阶段, C用于生成陷门的算法为 *Trapdoor*, 而不是先执行 *Trapdoor**算法再执行 *ReTrapdoor* 算法.

3) 安全游戏 Game 3. 与 Game 2 的其他部分都相同, 除了挑战阶段, C用于生成密文的算法为 *Encrypt*, 而不是先执行 *Encrypt**算法再执行 *ReEncrypt* 算法. 事实上, Game 3 就是原基础方案 SM9-IBEET 的安全性游戏.

现在我们证明, Game 1 和 Game 2, Game 2 和 Game 3 分别具有不可区分性.

Game 1 和 Game 2 之间, 对于任何篡改算法 *Trapdoor**, 其生成的陷门 td'在经由 CRF 的重随机化算法 *ReTrapdoor* 后, 由于数据的可延展性, td'会重新分布, 被映射到与原始 *Trapdoor* 相同的输出空间中. 也就是说, 即使敌手篡改了 *Trapdoor* 算法的实现, 它也难以区分 td'是由 *Trapdoor* 算法产生, 还是由先执行 *Trapdoor**算法再执行 *ReTrapdoor* 产生. 因此, Game 1 和 Game 2 之间具有不可区分性.

Game 2 和 Game 3 之间, 对于任何篡改算法 *Encrypt**, 其生成的密文 C'在经由 CRF 的重随机化算法 *ReEncrypt* 后, 由于数据的可延展性, C'会重新分布, 被映射到与原始密文相同的输出空间中. 也就是说, 即使敌手篡改了 *Encrypt* 算法的实现, 它也难以区分 C'由 *Encrypt* 算法产生, 还是是由先执行 *Encrypt**算法再执行 *ReEncrypt* 产生. 因此, Game 2 和 Game 3 之间具有不可区分性.

综上所述, Game 1 与 Game 3 具有不可区分性, SM9-IBEET-CRF 满足与原方案相同的 IBE-IND-CCA 安全性与 IBE-OW-CCA 安全性. 这种选择密文攻击下的不可区分性表明云服务器与用户之间的 CRF 具有弱保留安全性, Game 1, Game 2, Game 3 之间的不可区分性证明 CRF 有弱抵抗渗透性. 证毕.

5 方案对比

在本节中主要从计算开销、通信开销、安全性等方面对本文方案(SM9-IBEET-CRF)与其他等式测试的公钥加密方案和支持关键词检索的公钥加密文献 [4, 6, 11, 13, 15, 17] 进行比较. 其中, 文献 [4] 为具有前向安全性的公钥可搜索加密方案(FS-PKSE); 文献 [6] 为带有关键字搜索的公钥认证加密方案(PAEKS); 文献 [11] 为具有灵活授权机制的公钥加密等式测试方案(PKEET-FA); 文献 [13] 为支持等式测试的无证书公钥加密方案(CLE-PKEET); 文献 [15] 为支持等式测试的异构签密方案(HSC-ET); 文献 [17] 为支持等式测试标识加密方案(IBEET).

为评估方案性能, 将本文方案与其他方案在相同的环境下逐一对比, 该环境配置的处理器为 Intel® Core™ i7-8750H, 内存为 16 GB(RAM), 在 VMware 软件的虚拟机上运行, 在 PBC(pairing-based cryptography library)库中实现双线性对的接口, 实现了双线性对公钥密码体制的有效仿真, 达到了 1 024 b RSA 安全.

使用 SM9 定义 256 b 的 BN 曲线, 椭圆曲线方程为 $y^2 = x^3 + b$来生成映射 $e: G_1 \times G_2 \to G_T$, 嵌入次数为 12, 根据 SM9 的参数配置 PBC 库中对应的算法, 进行多次模拟后取平均值, 与之前的文献进行了对比, 其中涉及的符号定义和密码算法的执行时间定义分别如表 1 和表 2 所示.

为评估方案的通信开销, 考虑部署 2 类无线传感器节点平台 MICAz[31] 和 Tmote Sky[32]. 其中 MICAz 配置的微控制器为 ATmega128L, 内存为 4 KB(RAM). Tmote Sky 配置的微控制器为 MSP430, 内存为 10 KB(RAM). 采用 CC2420, 2.4 GHz IEEE 802.15.4 作为射频收发器标准, 在 TinyOS 系统运行. 使用文献 [33] 的方法, 在 2 类传感器节点架构体系上实现对公钥密码通信系统的有效仿真.

5.1 计算开销对比

我们首先比较了不同方案, 包括 Enc 加密操作、Dec 解密操作和 Test 等式测试操作的计算开销. 具体结果如表 3 所示.

图 3(a)表示在模拟环境下, 不同方案的加密时间随消息数量的变化, 虽然当消息数量增加到 100 时,

Table 1 Symbols Definition

表 1 符号定义

符号	含义
$\|G_1\|$	G_1 中元素的大小
$\|G_2\|$	G_2 中元素的大小
$\|G_T\|$	G_T 中元素的大小
$\|G\|$	对称配对 G 中元素的大小
$\|G_T'\|$	对称配对 G_T' 中元素的大小
$\|\mathbb{Z}_p\|$	$\mathbb{Z}_p$ 中元素的大小
$\|PK\|$	公钥长度
$\|CL\|$	密文长度
$\|TD\|$	陷门长度
T_p	1 次双线性配对运算
T_{x1}	1 次 G_1 或 G_2 上的幂运算
T_{x2}	1 次 G_T 上的幂运算

Table 2 Execution Time of Cryptographic Operation

表 2 密码操作的执行时间

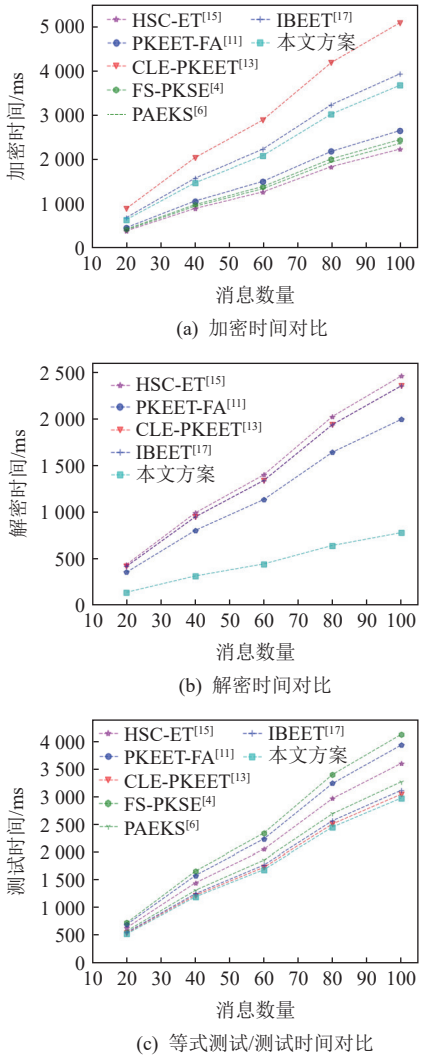
具体操作	计算时间/ms
T_P	7.595 4
T_{x1}	3.891 5
T_{x2}	1.235 7

Table 3 Computation Cost Comparison of Different Schemes

表 3 不同方案的计算开销对比

方案	加密	解密	等式测试/搜索
FS-PKSE[4]	$6T_{x1}$		$5T_p+3T_{x2}$
PAEKS[6]	T_p+4T_{x1}		$4T_p+T_{x2}$
PKEET-FA[11]	$6T_{x1}$	$5T_{x1}$	$2T_p+6T_{x1}$
CLE-PKEET[13]	$4T_p+5T_{x1}$	$2T_P+2T_{x1}$	$4T_p$
HSC-ET[15]	$5T_{x1}+2T_{x2}$	$3T_p+T_{x2}$	$4T_p+4T_{x2}$
IBEET[17]	$2T_p+6T_{x1}$	$2T_p+2T_{x1}$	$4T_p$
本文方案	$2T_p+5T_{x1}+2T_{x2}$	T_p	$4T_p$

本文方案的时间开销比文献 [4, 6, 11, 15] 大 1.58 倍左右，但很明显，与其他标识加密的等式测试的文献 [13, 17] 相比，本文方案的时间开销要小得多. 作为一种 IBE-ET 体制，本方案在加密时间上的开销是可以接受的. 与本文方案相比，文献 [11] 并没有实现 IBE 体制，在实际场景中，存在着密钥管理的问题；文献 [15] 异构等式测试，只能实现 PKI 端到 IBE 端的测试环境，具有一定的限制；文献 [4, 6] 实现的可搜索加密，不能对密文解密，只支持关键词搜索，无法搜索整段密文，搜索性有所降低. 本文方案的等式测试功能，可



(a) 加密时间对比

(b) 解密时间对比

(c) 等式测试/测试时间对比

Fig. 3 Computation overhead comparison for different schemes

图 3 不同方案的计算开销对比

以实现双向密文的任意测试；用户与测试者均采用标识加密的方法，避免了密钥管理的问题；与其他标识加密的方案相比，降低了加密开销.

从图 3(b)可以得出，本文方案在解密过程的计算时间远小于其他对比方案，具有解密开销上的优势.

从图 3(c)中得出，本文方案与文献 [13, 17] 在测试计算过程耗费的计算时间是相近的，文献 [4, 11, 15] 在测试场景下耗费的时间要大于其他方案. 文献 [4, 6] 实现的传统可搜索加密并不能支持整段密文的测试，可以看出，本文方案在等式测试过程中耗费的时间是合理且具有一定优势的.

5.2 通信开销与功能对比

表 4 表示的是不同方案在通信开销与实际功能上的对比，可以看出本文方案与其他方案相比，具有更强的安全性与功能性. 在实际应用场景中，本文方

Table 4 Comparison of Communication Overhead and Function of Different Schemes

表 4 不同方案的通信开销与功能对比

方案	$\|PK\|$	$\|CL\|$	$\|TD\|$	等式测试	标识加密	抗渗透攻击	支持国密算法
FS-PKSE[4]	$4\|G\|$	$5\|G\|+\|\mathbb{Z}_p\|$	$5\|G\|+\|\mathbb{Z}_p\|$	×	×	×	×
PAEKS[6]	$\|G\|$	$2\|G\|$	$\|G\|$	×	×	×	×
PKEET-FA[11]	$3\|G\|$	$5\|G\|+\|\mathbb{Z}_p\|$	$\|\mathbb{Z}_p\|$	√	×	×	×
CLE-PKEET[13]	$2\|G\|$	$3\|G\|+\|\mathbb{Z}_p\|$	$\|G\|$	√	×	×	×
HSC-ET[15]	$\|G\|$	$3\|G\|+2\|\mathbb{Z}_p\|$	$\|G\|$	√	√	×	×
IBEET[17]	$2\|G\|$	$4\|G\|+\|\mathbb{Z}_p\|$	$\|G\|$	√	√	×	×
本文方案	$\|G_1\|$	$3\|G_1\|+\|G_2\|+2\|\mathbb{Z}_p\|$	$\|G_2\|$	√	√	√	√

注: “√”表示存在, “×”表示不存在.

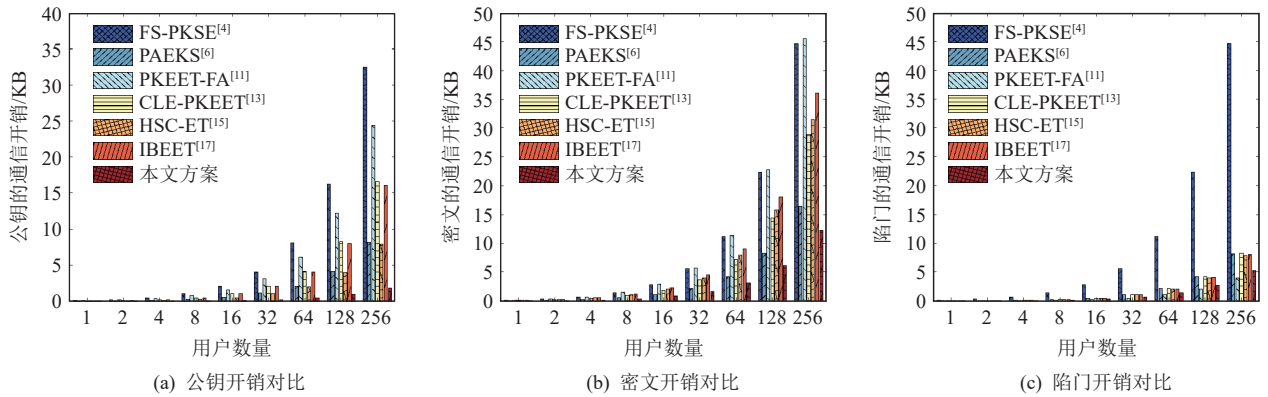
案实现的标识加密体制避免了证书管理的问题, 大大降低在通信过程中的开销. 同时, 支持等式测试的功能相比于其他可搜索加密文献 [4, 6] 具有更强的搜索性. CRF 的设置使得本文方案具备抵抗渗透攻击的能力, 这意味着在面对算法篡改攻击这类威胁时本文方案提供了更高的安全性. 值得注意的是, 本文方案还是唯一一个支持国密 SM9 算法的方案.

图 4 表示在模拟环境下, 不同方案随着用户数量增加下各种通信开销的对比. 从图 4(a)(b)可以看出, 本文方案享有最低的公钥通信开销与密文通信开销. 并且在图 4(c)中, 本文方案的陷门开销小于除了文献 [11] 以外的所有方案的开销. 这是由于本文方案不仅实现了标识加密体制, 在实际场景中避免了公钥证书的通信开销. 同时还是所有文献中唯一建立在非对称双线性配对基础上的方案, 这大大降低了在实际通信场景中的存储开销. 由此可见本文方案具有通信开销上的优势, 更适用于实际应用场景.

总的来说, 通过严格的实验仿真与性能对比证明, 本文方案在计算开销与通信开销上都具有一定的优势. 等式测试功能的引入, 使本文方案具有比可搜索加密方案更强的搜索性; 标识加密体制的拓展, 解决了密钥协商和证书管理的问题; 逆向防火墙的部署, 进一步提升了本文方案抵抗渗透攻击与篡改攻击的能力. 本文方案不仅解决了 SM9 密文难以搜索的问题, 还解决了目前支持等式测试的标识加密体制下计算与通信开销大、安全性弱的问题. 同时, 本文方案是国密 SM9 密码算法在云计算场景下的一次良好应用, 对于推动我国密码领域的安全研究也具有一定意义.

6 结 论

针对已有 IBEET 算法难抵抗渗透攻击的问题, 本文提出了一种支持等式测试并具有逆向防火墙的 SM9 标识密码方案 SM9-IBEET-CRF, 该方案可以运用于云服务器中加密数据的外包计算方案. 本文方案在用户与云服务器之间的上行信道分别部署了密码逆向防火墙; 形式化了本文方案的系统模型和定义, 并考虑 3 种不同的对手来定义安全模型; 然后在 BDH 假设下的随机预言机模型中证明了它的安全性; 最后通过严格的实验仿真和分析结果表明, 本文方案比已有方法在解密与通信开销方面具有一定的效率优势.



(a) 公钥开销对比 (b) 密文开销对比 (c) 陷门开销对比

Fig. 4 Comparison of communication overhead of different schemes

图 4 不同方案通信开销对比

作者贡献声明：熊虎提出了算法思路和实验方案；林烨负责完成实验并撰写论文；姚婷协助完成实验并修改论文.

参 考 文 献

[1] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing[J]. Communications of the ACM, 2010, 53(4): 50−58

[2] Wang Wei, Xu Peng, Liu Dongli, et al. Lightweighted secure searching over public-key ciphertexts for edge-cloud-assisted industrial IoT devices[J]. IEEE Transactions on Industrial Informatics, 2020, 16(6): 4221−4230

[3] Boneh D, Di Crescenzo G, Ostrovsky R, et al. Public key encryption with keyword search[C] //Proc of the 2004 Cryptology-EUROCRYPT. Berlin: Springer, 2004: 506−522

[4] Zeng Ming, Qian Haifeng, Chen Jie, et al. Forward secure public key encryption with keyword search for outsourced cloud storage[J]. IEEE Transactions on Cloud Computing, 2022, 10(1): 426−438

[5] Xu Peng, Susilo W, Wang Wei, et al. ROSE: Robust searchable encryption with forward and backward security[J]. IEEE Transactions on Information Forensics and Security, 2022, 17: 1115−1130

[6] Huang Qiong, Li Hongbo. An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks[J]. Information Sciences, 2017, 403: 1−14

[7] He Kun, Chen Jing, Zhou Qinxi, et al. Secure dynamic searchable symmetric encryption with constant client storage cost[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 1538−1549

[8] Xu Peng, Wu Qianhong, Wang Wei, et al. Generating searchable public-key ciphertexts with hidden structures for fast keyword search[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(9): 1993−2006

[9] Yang Guomin, Tan C, Huang Qiong, et al. Probabilistic public key encryption with equality test[C] //Proc of the 2010 Cryptology-CT-RSA: The Cryptographers'Track at the RSA Conf 2010. Berlin: Springer, 2010: 119−131

[10] Tang Qiang. Public key encryption supporting plaintext equality test and user - specified authorization[J]. Security and Communication Networks, 2012, 5(12): 1351−1362

[11] Ma Sha, Huang Qiong, Zhang Mingwu, et al. Efficient public key encryption with equality test supporting flexible authorization[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(3): 458−470

[12] Lee H T, Ling S, Seo J H, et al. Semi-generic construction of public key encryption and identity-based encryption with equality test[J]. Information Sciences, 2016, 373: 419−440

[13] Qu Haipeng, Yan Zhen, Lin Xijun, et al. Certificateless public key encryption with equality test[J]. Information Sciences, 2018, 462: 76−92

[14] Wang Yujue, Pang H, Deng R H, et al. Securing messaging services through efficient signcryption with designated equality test[J]. Information Sciences, 2019, 490: 146−165

[15] Xiong Hu, Zhao Yanan, Hou Yingzhe, et al. Heterogeneous signcryption with equality test for IIOT environment[J]. IEEE Internet of Things Journal, 2021, 8(21): 16142−16152

[16] Alrawais A, Alhothaily A, Cheng Xiuzhen, et al. SecureGuard: A certificate validation system in public key infrastructure[J]. IEEE Transactions on Vehicular Technology, 2018, 67(6): 5399−5408

[17] Ma Sha. Identity-based encryption with outsourced equality test in cloud computing[J]. Information Sciences, 2016, 328: 389−402

[18] Shamir A. Identity-based cryptosystems and signature schemes[C] // Proc of the 1984 Cryptology-CRYPTO. Berlin: Springer, 1984: 47−53

[19] Patsakis C, Charemis A, Papageorgiou A, et al. The market's response toward privacy and mass surveillance: The Snowden aftermath[J]. Computers & Security, 2018, 73: 194−206

[20] Bellare M, Paterson K G, Rogaway P. Security of symmetric encryption against mass surveillance[C/OL] //Proc of the 2014 Cryptology-CRYPTO. Berlin: Springer, 2014[2023-01-01].https://link.springer.com/chapter/10.1007/978-3-662-44371-2_1

[21] Mironov I, Stephens-Davidowitz N. Cryptographic reverse firewalls[C] //Proc of the 2015 Cryptology-EUROCRYPT. Berlin: Springer, 2015: 657−686

[22] Dodis Y, Mironov I, Stephens-Davidowitz N. Message transmission with reverse firewalls-secure communication on corrupted machines[G] //LNCS 9814: Proc of the 2016 Cryptology-CRYPTO. Berlin: Springer, 2016: 341–372

[23] Ma Hui, Zhang Rui, Yang Guomin. Concessive online/offline attribute based encryption with cryptographic reverse firewalls-secure and efficient fine-grained access control on corrupted machines[C] //Proc of the 2018 European Symp on Research in Computer Security. Berlin: Springer, 2018: 507−526

[24] Chen Rongmao, Mu Yi, Yang Guomin, et al. Cryptographic reverse firewall via malleable smooth projective Hash functions[C] //Proc of the 2016 Cryptology-ASIACRYPT. Berlin: Springer, 2016: 844−876

[25] Tang Qiang. Towards public key encryption scheme supporting equality test with fine-grained authorization[C] //Proc of the 16th Australasian Conf on Information Security and Privacy. Berlin: Springer, 2011: 389−406

[26] Tang Qiang. Public key encryption schemes supporting equality test with authorisation of different granularity[J]. International Journal of Applied Cryptography, 2012, 2(4): 304−321

[27] Zhou Yuyang, Guan Yuanfeng, Li Fagen, et al. Cryptographic reverse firewalls for identity-based encryption[G] // CCIS 1105: Proc of the 2nd Int Conf on Frontiers in Cyber Security. Berlin: Springer, 2019: 36–52

[28] Boneh D, Franklin M. Identity-based encryption from the weil pairing[C] //Proc of the 2001 Cryptology-CRYPTO. Berlin: Springer, 2001: 213–229

[29] Boyen X, Mei Qixiang, Waters B, Direct chosen ciphertext security from identity-based techniques[C] //Proc of the 12th ACM Conf on Computer and Communications Security. New York: ACM, 2005: 320–329

[30] Boneh D, Boyen X. Short signatures without random oracles[C] // Proc of the 2004 Cryptology-EUROCRYPT. Berlin: Springer, 2004: 56–73

[31] Crossbow. MICAz datasheet[DB/OL]. [2023-01-01]. http://bullseye.xbow.com:81/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf

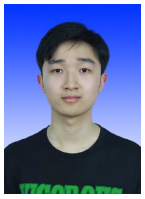
[32] Moteiv Corporation. Tmote Sky datasheet[DB/OL]. [2023-01-01]. www. moteiv. com/ products/docs/tmote-sky-datasheet. pdf

[33] Shim K A. BASIS: A practical multi-user broadcast authentication scheme in wireless sensor networks[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(7): 1545−1554



Xiong Hu, born in 1982. PhD, professor, PhD supervisor. His main research interests include cryptography and privacy-preserving computation.

熊　虎, 1982 年生. 博士, 教授, 博士生导师. 主要研究方向为密码学和隐私保护计算.



Lin Ye, born in 2000. Master candidate. His main research interests include public key cryptography and blockchain.

林　烨, 2000 年生. 硕士研究生. 主要研究方向为公钥密码、区块链.



Yao Ting, born in 1997. Master candidate. Her main research interest includes public key cryptography.

姚　婷, 1997 年生. 硕士研究生. 主要研究方向为公钥密码.