基于仲裁的 SM9 标识加密算法

秦宝东[1),2),3)] 张博鑫[1)] 白 雪[1)]

[1)](西安邮电大学网络空间安全学院 西安 710121)

[2)](西安电子科技大学综合业务网理论及关键技术国家重点实验室 西安 710071)

[3)](保密通信重点实验室 成都 610041)

摘 要 SM9-IBE 是我国于 2016 年发布的标识加密算法行业标准.标识加密算法以用户的标识(如邮件地址、身份证号等)作为公钥,从而降低系统管理用户密钥的复杂性.然而,标识加密算法的密钥撤销和更新问题却变得更加困难.此外,SM9 算法的结构特殊使得已有技术无法完全适用于该算法.为此,本文提出一种基于仲裁的 SM9 标识加密算法,可快速实现对用户访问权限的撤销和更新操作.该算法引入一个可信第三方(即仲裁者)用于管理用户的部分私钥,使得用户必须借助仲裁者的帮助才能访问 SM9 密文,同时仲裁者无法从用户密文中获取任何有用信息.在安全性方面,该算法被证明在随机预言机模型下满足适应性选择标识和可复制选择密文攻击安全性.在效率方面,该算法不仅保持了原始 SM9 算法的效率优势,而且使用户的解密时间缩短了将近 10 倍.

关键词 标识加密算法;SM9;仲裁者;密钥撤销;密钥更新

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2022.00412

Mediated SM9 Identity-Based Encryption Algorithm

QIN Bao-Dong[1),2),3)] ZHANG Bo-Xin[1)] BAI Xue[1)]

[1)](*School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an* 710121)

[2)](*The State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an* 710071)

[3)](*Science and Technology on Communication Security Laboratory, Chengdu* 610041)

Abstract SM9-IBE is an industry standard for identity-based encryption (IBE) algorithms issued by China in 2016. In IBE, any string, e.g., email address and identity number, can be viewed as the user's public key, thereby reducing the complexity of management of user keys. As in the traditional PKI-based setting, revocation capability is very necessary for IBE setting once a user's secret key is exposed or expired. Though, there exists an efficient way to revoke users in the traditional PKI settings, it becomes more difficult in the IBE setting. Moreover, there are only a few ways to revoke users in the IBE setting, and those revocation technologies mainly rely on a homomorphic property between the IBE master secret key and users' identity-based secret keys. However, this method is not fully applicable to SM9 algorithm due to its special algebraic structure of users' secret keys. In this paper, we propose mediated SM9 IBE algorithm (shorted as SM9-mIBE), which can directly revoke and update user's access rights. The algorithm introduces a trusted third party (also called mediator) to manage a part of the user's secret key, so that the user must rely on the mediator's help to decrypt the SM9 ciphertexts. Meanwhile, the mediator cannot obtain any useful information from user's ciphertexts. To achieve such revocation, the key generation center (KGC) generates a blind SM9 user key and sends it to the mediator. While the blind factor is given to the user as his secret key. When the mediator decrypts a ciphertext, the

收稿日期:2020-12-08;在线发布日期:2021-09-15.本课题得到国家自然科学基金面上项目(61872292)和装备预研重点实验室基金(6142103190101)资助.**秦宝东**,博士,教授,中国计算机学会(CCF)会员,主要研究领域为公钥密码学基础理论及应用.E-mail:qinbaodong@xupt.edu.cn.**张博鑫**,硕士研究生,主要研究方向为公钥密码学.**白 雪**,硕士研究生,主要研究方向为公钥密码学.

result will hide the real message by a factor and only a valid user can remove this factor from the result to obtain the original message. To revoke a user from the system, the KGC only needs to request the mediator only to remove the corresponding blind SM9 user key from the mediator key list. In terms of security, the new algorithm is proven to be secure against adaptive identity and replayable chosen-ciphertext attacks in random oracle model if the gap bilinear collision attack problem is hard. To evaluate its efficiency, we compare it with the original SM9 IBE scheme and a previous related scheme proposed by Sun et al. in both theory and practice. In theory, our SM9-mIBE scheme keeps the system key generation algorithm and encryption algorithm as in original SM9-IBE scheme. To revoke a user's secret key, Sun et al.'s method needs to update all ciphertexts of this user, while ours only needs to update a blind SM9 secret key of this user. In practice, we implement these three schemes using JPBC library with a Type-F elliptic curve. The results showed that our SM9-mIBE scheme not only maintains the efficiency of the original SM9 encryption algorithm, but also improves the user's decryption speed nearly 10 times. Compared with Sun et al.'s scheme, at the user side, our decryption algorithm is also 10 times faster than that of Sun et al.'s scheme. However, on the third part server side, our SM9-mIBE scheme takes more time than Sun et al.'s scheme.

Keywords　identity-based encryption; SM9; mediator; key revocation; key update

1　引　言

1984 年，Shamir[1] 提出了标识密码（Identity-Based Cryptosystem，简称 IBC）的概念. 它是一种特殊的公钥密码系统，允许用户以任意标识，如邮箱地址、身份证号等，作为用户公钥对消息进行加密或签名验证. 用户的私钥由系统密钥和身份标识唯一确定，从而简化了传统公钥密码体制中基于 PKI 的复杂密钥管理方式. 直到 2001 年，Boneh 和 Franklin[2] 以及 Cocks[3] 才分别给出首个标识加密算法（Identity-Based Encryption，简称 IBE）的构造. 经过近 20 年的发展，学术界从算法的效率、功能和安全性等角度提出了大量的 IBE 算法，如文献[4-7].

SM9[8] 是我国采用的一种标识密码标准，由国家密码管理局于 2016 年 3 月 28 日发布，其相关标准为“GM/T 0044.1-2016 SM9 标识密码算法”. SM9 系列算法包含密钥交换、签名和加密三种，其中 SM9 标识签名算法已于 2018 年 11 月正式成为国际标准（ISO/IEC 14888-3:2018①）. 2019 年 10 月，SM9 标识加密算法（ISO/IEC 18033-5:2015 补篇 1 SM9-IBE②）和 SM9 密钥协商协议（ISO/IEC 11770-3:2015 补篇 2 SM9-KA③）也分别进入国际标准 PDAM 阶段. 随着 SM9 算法在密码技术和网络空间安全领域占据愈来愈重要的国际地位，近年来我国学者对 SM9 算法进行了较多的研究，如文献[9-13].

尽管标识密码系统简化了用户公钥的管理环节，但是标准的标识密码算法，包括 SM9 并没有提供用户密钥撤销机制. 一旦用户密钥泄露，则原有的用户标识将无法继续使用，否则恶意攻击者可以利用泄露的密钥访问用户的所有密文. 近年来，IBC 密码系统的密钥撤销问题备受国内外学者的关注并取得了一定的研究成果. 这些方法可以归纳为以下几种：

（1）针对一些具体 IBC 算法[6-7]，基于二叉树、子集差分等密钥存储和更新方法的间接密钥撤销技术. 2008 年，Boldyreva 等人[14] 提出第一个高效的可撤销 IBE 方案（简称 RIBE）. 该方案利用完全二叉树存储方法使得密钥更新复杂度仅为 $\mathcal{O}(R\cdot\log N/R)$，其中 N 和 R 分别表示系统用户数量和被撤销用户数量. 近年来，在 Boldyreva 等人的工作基础上，学者们从安全模型、参数大小、密钥更新方法等方面提出了一些新的 RIBE 方案[15-20]. 此外，这些 IBE 方案的密钥撤销技术还被推广到其他密码学原语中，如可撤销的层次 IBE 方案（简称 RHIBE）[21-23]. 特别地，2013 年，Seo 和 Emura[17] 针对实际应用中解密密钥可能泄露的问题，提出一种抵抗解密密钥泄露攻击（DKER）的 RIBE 方案. 在随后的工作中，DKER

① https://www.iso.org/standard/76382.html
② https://www.iso.org/standard/78751.html
③ https://www.iso.org/standard/78750.html

安全性已成为构造 R(H)IBE 方案的基本安全性要求[15,18,23-25]. 然而,上述构造通常依赖具体结构的 IBC 算法. 特别地,基于双线性配对的 RIBE 方案需要具有所谓的密钥可随机化的性质. 目前,这种方法还无法应用于具有特殊代数结构的 SM9 系列算法中.

(2)基于通用构造方法的密钥撤销技术[2,26]. 尽管该方法具有通用性,但是文献[2]中的密钥更新复杂性较高,达到 $\mathcal{O}(N-R)$. 文献[26]虽然降低了密钥更新复杂性,但是 IBE 密文从 $\mathcal{O}(1)$ 扩展为 $\mathcal{O}(\log N)$,或者依赖基于身份的广播加密. 将这些通用方法应用于 SM9 对系统效率影响较大并不现实.

(3)服务器辅助的密钥撤销技术[27-30]. 该方法主要借助第三方服务器来实现对用户密钥的撤销功能. 一些工作中的第三方服务器主要起到用户部分密钥合成和密文转化的功能,如文献[28-30];另一些工作中的第三方服务器主要起到分发用户更新密钥的作用,如文献[27]. 这些方法主要利用到系统主密钥和用户密钥之间具有的同态性质,即将系统主密钥拆分成两部分,一部分用于生成用户身份标识对应的密钥,另一部分用于生成时间标识对应的密钥. 两种密钥组合可以合成不同时间周期内的完整用户密钥. 然而,SM9 的主密钥和用户密钥之间并不存在这种良好的代数性质.

到目前为止,几乎还没有完全针对 SM9 密钥撤销机制的研究. 2020 年,Sun 等人[11]首次研究了 SM9-IBE 算法的密钥撤销机制,提出一种服务器辅助的用户密钥撤销和更新技术. 在该技术中,服务器将接收到的数条密文进行二次加密后存储在服务器上,使得用户只能通过服务器的协助才能完全解密密文. 因此,该方法需要数据发送者和服务器之间建立一条安全信道用于传递原始密文,否则用户可以直接解密原始密文,从而失去对用户解密能力的撤销功能. 此外,该方法通过更新用户所有密文来实现对以前密钥的撤销,从而使得服务器的工作负担随密文数量线性增长.

本文工作:2003 年,Ding 等人[31]提出利用仲裁服务器实现基于 RSA 的标识加密算法. 尽管该算法后来被证明存在安全缺陷[32],但是该算法提供了一种快速的用户密钥撤销方法. 该方法将用户密钥(而非系统主密钥)拆分成两部分,一部分存储在仲裁服务器上,而另一部分由用户持有,使得用户需要借助仲裁服务器才能完成解密工作. 由于仲裁服务器仅持有用户部分密钥信息,因此无法从用户密文中获取任何明文信息. 受此思想的启发,本文提出一种基于仲裁的 SM9 标识加密算法(简称 SM9-mIBE),它利用仲裁服务器实现对用户密钥的撤销和更新机制. 在该算法中,密钥中心将盲化后的 SM9-IBE 用户密钥(称之为仲裁私钥)发送给仲裁服务器,而把盲化因子(称之为用户私钥)发送给用户. 仲裁服务器对用户密文的解密结果(称之为部分解密结果)都会含有一个盲化因子,只有合法用户才能将盲化因子从仲裁服务器的解密结果中去掉,从而恢复出原始消息. 与 Sun 等人的方法相比,该 SM9-mIBE 不需要数据发送者和服务器之间建立一条安全信道,也不需要更新用户的所有密文,仲裁服务器只需要维护一个仲裁私钥列表:当用户密钥泄露时,仲裁服务器将相应的仲裁私钥从列表中删除即可撤销该用户的密文访问权限;或者密钥中心将新的盲化因子和仲裁私钥分别发送给用户和仲裁服务器即可实现对用户密钥的更新操作.

内容安排:本文第 2 节回顾 SM9-IBE 算法的定义、困难问题假设及相关的基础知识;第 3 节介绍基于仲裁的 SM9-IBE 算法的形式化定义、安全模型及具体构造;第 4 节给出算法的安全性分析及证明;第 5 节分析和比较相关算法的性能;最后是本文工作的总结.

2 基础知识

2.1 符号说明

在本文中,"$A\|B$"表示两个比特串 A 和 B 的级联,"$|A|$"表示 A 的比特长度. 若 S 是一个集合,则 $s\leftarrow S$ 表示从集合 S 中均匀随机选取一个元素 s. 若 S 是一个算法,则 $s\leftarrow S$ 表示算法的输出结果.

在 SM9 算法中,

(1) H_v 表示一个输入为任意比特串,输出为 v 比特的密码杂凑函数.

(2) $\mathrm{H2RF}_1(H_v,Z,p)$ 表示一个输入为杂凑函数 H_v,比特串 Z 和整数 p,输出为整数 $h\in[1,p-1]$ 的密码函数.

(3) $\mathrm{KDF}(H_v,Z,l)$ 表示一个输入为杂凑函数 H_v,比特串 Z 和整数 l,输出为比特串 $K\in\{0,1\}^l$ 的密码函数.

2.2 双线性配对

令 $\mathcal{P}(1^\lambda)$ 表示一个双线性配对群生成算法,输入为安全参数 1^λ,输出为 $(e,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,p,g,h)$,其

中$\mathbb{G}_1$、$\mathbb{G}_2$和$\mathbb{G}_T$是阶为素数 p 的循环群，$e:\mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_T$是一个双线性映射，$g$ 和 h 分别是$\mathbb{G}_1$和$\mathbb{G}_2$的生成元. 一个双线性映射应满足以下性质：

(1) 双线性性. 对于任意的$\alpha,\beta\in\mathbb{Z}_p$，有$e(g^\alpha,h^\beta)=e(g,h)^{\alpha\beta}$；

(2) 非退化性. 若 g 和 h 不是单位元，则 $e(g,h)$ 也非单位元；

(3) 可计算性. 对于任意 g 和 h，存在多项式时间算法计算 $e(g,h)$.

2.3　困难问题假设

本节主要介绍文献[33]提出的 BCAA1 问题(Bilinear Collision Attack problem)在提供不同判定性问题预言机时的困难性问题假设. 判定性问题包括 DIDH 问题(Decisional Inverse Diffie-Hellman problem)和 DBIDH 问题(Decisional Bilinear Inverse Diffie-Hellman problem). 在假设中，$e:\mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_T$表示一个双线性映射，$g$ 和 h 分别是$\mathbb{G}_1$和$\mathbb{G}_2$的生成元，群$\mathbb{G}_1$、$\mathbb{G}_2$和$\mathbb{G}_T$的阶为 p. 令 $g_T=e(g,h)$.

假设 1.　DIDH 假设. 对于随机选取的元素 α，$\beta,\gamma\in\mathbb{Z}_p$，区分下列两个四元组

$$(g_T,g_T^\alpha,g_T^\beta,g_T^{\beta/\alpha})\text{和}(g_T,g_T^\alpha,g_T^\beta,g_T^\gamma)$$

在计算上是困难的.

实际上，判定$(g_T,g_T^\alpha,g_T^\beta,g_T^\gamma)$是否构成 DIDH 对等价于判定$(g_T,g_T^\alpha,g_T^\gamma,g_T^\beta)$是否构成 DDH (Decisional Diffie-Hellman)对. 因此，$\mathbb{G}_T$上的 DIDH 问题和传统的 DDH 问题是等价的.

假设 2.　DBIDH 假设[8]. 对于随机选取的元素 $\alpha,\beta,\gamma\in\mathbb{Z}_p$，区分下列两个五元组

$$(g,h,g^\alpha,g^\beta,e(g,h)^{\beta/\alpha})\text{和}(g,h,g^\alpha,g^\beta,e(g,h)^\gamma)$$

在计算上是困难的.

假设 3.　Gap-τ-BCAA1 假设[8]. 对于任意正整数 τ 和随机选取的元素$\alpha,z_0,z_1,\cdots,z_\tau\in\mathbb{Z}_p$，给定下列元素

$$(g,h,g^\alpha,z_0,(z_1,h^{\frac{\alpha}{\alpha+z_1}}),\cdots,(z_\tau,h^{\frac{\alpha}{\alpha+z_\tau}}))$$

以及一个求解 DIDH 问题的有效算法，计算$e(g,h)^{\frac{\alpha}{\alpha+z_0}}$是困难的.

注：上述 Gap-τ-BCAA1 假设与原始 SM9 方案基于的 Gap-τ-BCAA1 假设略有不同. 在 SM9 方案的安全性证明中，为了判定群$\mathbb{G}_1$上不同密文(即 C_1)解密结果的一致性，具体见文献[8]. 该假设仅需要访问 DBIDH 问题的判定预言机. 在本中，由于密文通过仲裁服务器部分解密后，被转化成群$\mathbb{G}_T$中的元素 C_1'(见 3.2 节)，为了在安全性证明中判定密文解密结果的一致性，本文方案基于的 Gap-τ-BCAA1 假设需要访问群$\mathbb{G}_T$中的 DIDH 问题的判定预言机.

2.4　SM9-IBE 算法

SM9 标识加密算法(简称 SM9-IBE 算法)是一种混合加密算法，由基于标识的 SM9 密钥封装算法和数据封装算法组合而成. 数据封装算法包含两种形式：一种形式是一次一密＋消息认证码；另一种形式是分组加密.

下面简要回顾第一种形式的 SM9-IBE 算法.

(1) 系统参数. 密钥中心选择一个双线性配对群$\mathcal{G}=(e,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,p,g,h)$，随机选择 $s\in\mathbb{Z}_p$，计算 $w=g^s$和 $u=e(g,h)^s$. 则系统主公钥和系统主私钥分别为

$$mpk=(\mathcal{G},w,u,H,hid),$$
$$msk=s,$$

其中，$H(id)$是哈希函数 $\mathrm{H2RF}_1(H_v,id,p)$和$hid=3$.

(2) 用户密钥提取. 对于标识为 id 的用户，密钥中心计算用户密钥为

$$sk_{id}=h^{\frac{s}{s+H(id)}}.$$

(3) 加密. 对于消息 M 和身份标识 id，随机选择$r\in\mathbb{Z}_p$，计算密文

$$C_1=(w\cdot g^{H(id)})^r,$$
$$C_2=K_1\oplus M,$$
$$C_3=H_v(C_2\|K_2),$$

其中

$$t=u^r,$$
$$K_1\|K_2=\mathrm{KDF}(H_v,C_1\|t\|id,|M|+v).$$

(4) 解密. 对于密文(C_1,C_2,C_3)，首先利用私钥 sk_{id} 计算 $t'=e(C_1,sk_{id})$，然后利用 KDF 计算出

$$K_1\|K_2=\mathrm{KDF}(H_v,C_1\|t'\|id,|M|+v).$$

若 $H_v(C_2\|K_2)\neq C_3$，则返回$\perp$；否则，返回消息 $M=K_1\oplus C_2$.

ID-CCA 安全性. 适应性选择标识和选择密文攻击安全性(Adaptive Identity and Chosen-Ciphertext Attacks security，简称 ID-CCA 安全性)，除了平凡的询问外，允许攻击者选择任意标识并获取相应的私钥，以及任意密文并获取解密结果. 文献[8]证明，在随机预言机模型下，SM9-IBE 算法的 ID-CCA 安全性可以归约到求解 Gap-q_H-BCAA1 问题，其中 q_H+1 表示攻击者查询哈希函数 H 的次数. 详细内容可查看文献[8]定理 4.

3 基于仲裁的 SM9 标识加密算法

3.1 mIBE 的定义及安全模型

图 1 给出了基于仲裁的标识加密算法(简称 mIBE)的系统模型.它包含以下五个实体:

(1)密钥中心(KGC).主要负责生成系统主公钥 mpk 和主私钥 msk、标识 id 的用户私钥 sk_{user} 和仲裁私钥 sk_{sem} 及用户撤销列表 RL.

(2)数据发送者(DS).主要负责计算消息 M 的原始密文 C.

(3)存储服务器(SS).主要负责存储用户的密文.

(4)可信仲裁者(SEM).主要负责计算未撤销用户的原始密文 C 的部分解密结果 C'.

(5)数据接收者(DR).主要工作是从部分解密的密文 C' 中计算出原始消息 M.

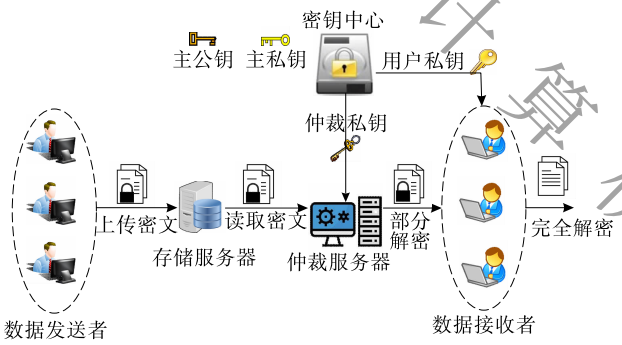


图 1 mIBE 系统模型

定义 1. mIBE 算法.一个 mIBE 算法包含以下七个(概率)多项式时间算法:

(1)系统参数生成算法 Setup(1^{λ}).该算法由密钥中心执行.它输入一个安全参数 λ,输出系统的主公钥 mpk 和主私钥 msk.此外,密钥中心维护一个初始化为空集的用户撤销列表 RL.

(2)用户注册算法 Register(id).该算法由用户和密钥中心交互执行.用户将自己的身份标识 id 发送给密钥中心进行合法性验证.若验证未通过,则密钥中心终止算法的执行.否则,利用主私钥 msk 计算仲裁私钥(id,sk_{sem})和用户私钥(id,sk_{user}),并分别发送给仲裁者和用户.仲裁者将密钥添加到自己的私钥列表 KL 中.

(3)加密算法 Enc(mpk,id,M).该算法由数据发送者执行.利用系统主公钥 mpk 和用户的身份标识 id,数据发送者计算消息 M 的原始密文 C 并将其发送给存储服务器.

(4)服务器解密算法 SDec(id,KL,C).该算法由可信仲裁者执行.当用户请求解密标识为 id 的密文时,仲裁者首先查找私钥列表 KL 中是否存在 id 的仲裁私钥 sk_{sem}.若不存在,则终止算法;否则,利用私钥 sk_{sem} 计算部分解密结果 C' 并将其发送给用户 id.

(5)用户解密算法 UDec(id,sk_{user},C').该算法由数据接收者执行.利用私钥 sk_{user} 从部分解密密文 C' 中计算出原始消息 M.

(6)用户撤销算法 Revoke(id,RL).该算法由密钥中心执行.当身份标识为 id 的用户被撤销时,密钥中心将 id 添加到用户撤销列表 RL 中并通知仲裁者将标识 id 的仲裁私钥从服务器中删除.

(7)密钥更新算法 Update(id).该算法由密钥中心交互执行.若身份标识为 id 的用户密钥被泄露,则密钥中心重新执行 Register(id)算法并将更新后的仲裁私钥(id,sk_{sem})和用户私钥(id,sk_{user})分别发送给仲裁服务器和用户.

mIBE 算法的安全模型:在下面的安全模型中,我们假设仲裁者是诚实且好奇的,即仲裁者会按照算法要求执行任务,但不会主动将自己的私钥泄露给其他用户及攻击者.同时仲裁者可能会设法从用户的密文中获取一些有用信息.我们将攻击者分成类型 I 和类型 II 以下两种:

类型 I. 攻击者可以获取任意标识 id 的仲裁私钥 sk_{sem} 和部分标识 id(不包括挑战标识 id^*)的用户私钥 sk_{user}.该类型攻击者刻画了仲裁者的好奇行为及与部分恶意用户的共谋行为,也说明该类型攻击涵盖了仲裁服务器的密钥泄露攻击.

类型 II. 攻击者可以获取任意标识 id 的用户私钥 sk_{user} 和部分标识 id(不包括挑战标识 id^*)的仲裁私钥 sk_{sem}.该类型攻击者刻画了被撤销用户的好奇行为.

对于攻击类型 II,若攻击者是被系统注销的用户,则系统需要将该用户对应的仲裁私钥从服务器的私钥列表 KL 中删除并将该用户的身份标识 id 添加到用户撤销列表 RL 中.该类撤销称之为用户撤销.若攻击者是外部用户并获取了某合法用户的私钥,此时用户的身份标识 id 依旧有效,不需要将用户身份标识 id 添加到用户撤销列表 RL 中,但是需要执行密钥更新算法 Register(id),为用户更新仲裁私钥和用户私钥,服务器需要更新 KL 中用户标识为 id 的仲裁私钥,而非用户撤销时所进行的删

除用户对应仲裁私钥的操作. 该类撤销称之为密钥撤销.

下面给出 mIBE 算法抗适应性选择标识和可复制选择密文攻击(Adaptive Identity and Replayable Chosen-Ciphertext Attacks，简称 ID-RCCA)的游戏定义.

初始化阶段. 挑战者运行系统参数生成算法 Setup(1^λ)获取系统主公钥 mpk 和主私钥 msk，并建立一个空的用户撤销列表 RL. 最后，挑战者将主公钥 mpk 发送给攻击者.

询问阶段 1. 挑战者建立两个空的列表 T_1 和 T_2 攻击者可以进行以下询问：

(1) 仲裁私钥查询. 当攻击者要查询标识 id 的仲裁私钥时，挑战者首先查找列表 T_1 中是否包含形如(id，sk_{sem})的元素. 若存在，则将 sk_{sem} 返回给攻击者；若不存在，挑战者运行 Register(id)算法得到仲裁私钥 sk_{sem} 和用户私钥 sk_{user} 并将仲裁私钥 sk_{sem} 发送给攻击者. 最后，挑战者将(id，sk_{sem})和(id，sk_{user})分别添加到列表 T_1 和 T_2 中.

(2) 用户私钥询问. 当攻击者要查询标识 id 的用户私钥时，挑战者类似仲裁私钥查询进行回答.

(3) 服务器解密查询. 当攻击者查询(id，C)的服务器解密结果时，挑战者执行 SDec(id，T_1，C)算法并将计算结果发送给攻击者.

(4) 用户解密查询. 当攻击者查询 C' 的用户解密结果时，挑战者首先查询列表 T_2 中是否存在包含用户私钥 sk_{user} 的元素. 若不存在，则直接返回终止符号 $\perp$；否则，挑战者执行算法 UDec(id，sk_{id}，C')并将计算结果发送给攻击者.

挑战. 攻击者提交一个挑战标识 id^* 和两个等长的消息 M_0 和 M_1. 挑战者随机选择一个比特 $b\in\{0,1\}$，计算密文 $C^*=\mathrm{Enc}(mpk,id^*,M_b)$ 并发送给攻击者.

询问阶段 2. 攻击者可以重复阶段 1 进行各种查询，但是有如下限制：

(1) 若攻击者询问了挑战标识 id^* 的仲裁私钥，则攻击者不能询问挑战标识 id^* 的用户私钥.

(2) 若攻击者询问了挑战标识 id^* 的用户私钥，则攻击者不能询问挑战标识 id^* 的仲裁私钥，并且该仲裁私钥必须被撤销或更新.

(3) 当用户解密查询结果为挑战消息 M_0 或 M_1 时，挑战者返回终止符号 $\perp$.

猜测. 最终，攻击者输出一比特 b' 作为对 b 的猜测结果.

定义 2.　ID-RCCA 安全性. 如果对于任意多项式时间攻击者 $\mathcal{A}$，在上面的 ID-RCCA 游戏中成功的优势 $Adv^{\mathrm{RCCA}}_{\mathcal{A},\mathrm{mIBE}}(\lambda)=\left|\Pr[b'=b]-\frac{1}{2}\right|$ 是可忽略的，则算法 mIBE 满足 ID-RCCA 安全性.

3.2　算法构造

基于仲裁的 SM9 标识加密算法(简称 SM9-mIBE)的具体构造如下：

(1) 系统参数生成算法 Setup(1^λ). 在输入安全参数 λ 后，该算法将执行以下操作：

① 密钥中心选择一个双线性配对群

$$\mathcal{G}=(e,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,p,g,h);$$

② 选择一个随机元素 $s\in\mathbb{Z}_p$ 并计算 $w=g^s$、$u=e(w,h)$. 令 $H(id)=\mathrm{H2RF}_1(H_v,id,p)$、$hid=3$ 及用户撤销列表 $RL=\varnothing$.

③ 密钥中心输出系统主公钥 mpk 和主私钥 msk，其中，

$$mpk=(\mathcal{G},w,u,H,hid)\ msk=s.$$

(2) 用户注册算法 Register(id)：在输入用户标识 id 后，该算法生成用户对应的用户私钥 sk_{user} 和仲裁私钥 sk_{sem}.

① 密钥中心首先随机选取一个元素 $a\in\mathbb{Z}_p$ 作为盲化因子，即用户私钥 $sk_{user}=a$.

② 计算仲裁私钥 $sk_{sem}=h^{\frac{s}{a(s+H(id))}}$.

③ 将(id，sk_{sem})和(id，sk_{user})分别发送给仲裁者和用户.

(3) 加密算法 Enc(mpk，id，M). 在输入主公钥 mpk，身份标识 id 和消息 M 后，该算法将执行以下操作：

① 数据拥有者首先选择一个随机元素 $r\in\mathbb{Z}_p$，并计算

$$Q=w\cdot g^{H(id)},$$

$$C_1=Q^r=g^{r(s+H(id))},$$

$$t=u^r.$$

② 利用密钥派生函数 KDF，计算

$$K_1\|K_2=\mathrm{KDF}(H_v,t\|id,|M|+v),$$

其中，K_1 和 K_2 的长度分别为 $|M|$ 比特和 v 比特.

③ 用户计算

$$C_2=K_1\oplus M,$$

$$C_3=H_v(C_2\|K_2).$$

④ 用户输出密文 $C=(C_1,C_2,C_3)$.

(4) 服务器解密算法 SDec(id，KL，C)：在输入

身份标识 id，仲裁私钥列表 KL 以及原始密文 C 后，该算法将执行以下操作：

① 仲裁者将密文拆分成 $C=(C_1,C_2,C_3)$，获得原始密文的第一部分 C_1.

② 仲裁者从私钥列表 KL 中获取标识为 id 的仲裁私钥 sk_{sem}. 若不存在，返回 $\perp$；否则，计算 $C_1'=e(C_1,sk_{sem})$.

③ 仲裁者将部分解密结果 $C'=(C_1',C_2,C_3)$ 返回给用户.

(5) 用户解密算法 UDec(id,sk_{user},C'). 在输入身份标识 id，用户私钥 sk_{user} 以及部分解密结果 C' 后，该算法将执行以下操作：

① 数据接收者将部分解密结果 C' 拆分

$$C'=(C_1',C_2,C_3).$$

② 利用用户私钥 sk_{user} 计算：

$$t'=(C_1')^{sk_{user}},$$

$$K_1\|K_2=\mathrm{KDF}(H_v,t'\|id,|M|+v),$$

$$C_3'=H_v(C_2\|K_2),$$

其中 K_1 和 K_2 分别为 $K_1\|K_2$ 的最左侧的 $|M|$ 比特以及最右侧的 v 比特.

③ 若 $C_3'\neq C_3$，则返回 $\perp$；否则，返回如下消息

$$M=K_1\oplus C_2.$$

(6) 用户撤销算法 Revoke(id,RL). 当身份标识为 id 的用户被撤销时，密钥中心将 id 添加到用户撤销列表 RL 中并通知仲裁者将标识 id 的仲裁私钥从服务器中删除.

(7) 密钥更新算法 Update(id). 若身份标识为 id 的用户需要更新密钥，则密钥中心重新执行算法 Register(id) 并将新的仲裁私钥 (id,sk_{sem}) 和用户私钥 (id,sk_{user}) 分别发送给仲裁服务器和用户.

正确性分析. 由于 $sk_{sem}=h^{\frac{s}{a(s+H(id))}}$、$sk_{user}=a$，所以

$$\begin{aligned}(C_1')^{sk_{user}}&=e(g^{r(s+H(id))},h^{\frac{s}{a(s+H(id))}})^a\\&=e(g^r,h^{\frac{s}{a}})^a\\&=e(g,h)^{rs}=t.\end{aligned}$$

因此，用户可以计算出正确的中间密钥 K_1 和 K_2，从而恢复出原始消息 M.

4 SM9-mIBE 的安全性分析

定理 1. 假设攻击者询问哈希函数 H 和 H_v 的次数分别为 q_H 和 q_{H_v}，询问用户解密的次数为 q_D. 若 Gap-q_H-BCAA1 问题是困难的且 H、H_v 和 KDF 被看作是随机预言机，则 SM9-mIBE 方案是 ID-RCCA 安全的. 具体有：

$$Adv^{\mathrm{RCCA}}_{\mathcal{A},\mathrm{SM9\text{-}mIBE}}(\lambda)\leq 2(q_H+1)Adv^{\mathrm{Gap\text{-}}q_H\text{-}\mathrm{BCCA1}}_{\mathcal{B}}(\lambda)+\frac{q_{H_v}+q_D}{2^v}.$$

证明. 我们通过定义一系列不可区分游戏的形式，证明定理 1 的结论. 令 $\mathcal{A}$ 表示一个攻击 SM9-mIBE 方案 ID-RCCA 安全性的概率多项式攻击者. 在游戏 Game_i 中，令 S_i 表示事件 $b'=b$. 则游戏 Game_0、Game_1 和 Game_2 定义如下：

游戏 Game_0. 该游戏和原始 ID-RCCA 安全性的游戏定义一致. 特别地，对于挑战标识 id^* 和两个长度相等的挑战消息 M_0 和 M_1，挑战者 $\mathcal{B}$ 随机选择一个比特 $b\in\{0,1\}$ 和一个随机元素 $r^*\in\mathbb{Z}_p$，然后计算

$$C_1^*=g^{r^*(s+H(id^*))},$$

$$C_2^*=K_1^*\oplus M_b,$$

$$C_3^*=H_v(C_2^*\|K_2^*),$$

其中 $t^*=u^{r^*}$，$K_1^*\|K_2^*=\mathrm{KDF}(H_v,t^*\|id^*,|M_b|+v)$ 且 K_1^* 和 K_2^* 的长度分别为 $|M_b|$ 比特和 v 比特. 则挑战密文记作密文 $C^*=(C_1^*,C_2^*,C_3^*)$. $\mathcal{A}$ 输出一比特 b' 作为对 b 的猜测.

根据 ID-RCCA 安全性的定义，则有：

$$Adv^{\mathrm{RCCA}}_{\mathcal{A},\mathrm{SM9\text{-}mIBE}}(\lambda)=\left|\Pr[S_0]-\frac{1}{2}\right|.$$

游戏 Game_1. 与游戏 Game_0 相比，该游戏仅将挑战密文中 $K_1^*\|K_2^*$ 的计算方式替换为随机选取，其他操作与游戏 Game_0 保持一致.

引理 1. 在随机预言机模型下，基于 Gap-q_H-BCAA1 假设，游戏 Game_0 和 Game_1 是不可区分的. 具体有：

$$|\Pr[S_0]-\Pr[S_1]|\leq 2(q_H+1)Adv^{\mathrm{Gap\text{-}}q_H\text{-}\mathrm{BCCA1}}_{\mathcal{B}}(\lambda),$$

其中 q_H 是攻击者查询哈希函数 H 的次数，$\mathcal{B}$ 是一个求解 Gap-q_H-BCAA1 问题的算法.

游戏 Game_2. 与游戏 Game_1 相比，若攻击者的用户解密询问 $(id,(C_1',C_2,C_3))$ 满足 $id=id^*$ 且 $C_1'=e(C_1^*,sk_{sem}^*)$，其中 sk_{sem}^* 是挑战标识 id^* 对应的仲裁私钥，则直接返回 $\perp$，其他操作与游戏 Game_1 保持一致.

引理 2. 在随机预言机模型下，游戏 Game_1 和 Game_2 是不可区分的. 具体有：

$$|\Pr[S_1]-\Pr[S_2]|\leq\frac{q_{H_v}+q_D}{2^v},$$

其中 q_{H_v} 和 q_D 分别表示攻击者查询哈希函数 H_v 和用户解密询问的次数.

游戏 Game$_3$. 与游戏 Game$_2$ 相比，该游戏将挑战密文中 C_2^* 的计算方式由 $K_1^* \oplus M_b$ 替换为随机选取，其他操作与游戏 Game$_2$ 保持一致.

在游戏 Game$_2$ 和 Game$_3$ 中，K_1^* 都是独立且仅在挑战密文中使用. 对于任意消息 M_b，$C_2^* = K_1^* \oplus M_b$ 和 K_1^* 的分布是一样，所以 $\Pr[S_2] = \Pr[S_3]$.

在游戏 Game$_3$ 中，显然我们有 $\Pr[S_3] = 1/2$. 结合后面的引理 1 和引理 2，我们可以得到定理 1.

定理 1 证毕.

下面，我们分别证明引理 1 和引理 2.

证明(引理 1).　给定 Gap-q_H-BCAA1 问题的一个实例 $(g, h, g^s, z_0, (z_1, h^{\frac{s}{s+z_1}}), \cdots, (z_{q_H}, h^{\frac{s}{s+z_{q_H}}}))$，其中 $z_i \in \mathbb{Z}_p$ 随机选取且互不相同，以及 DIDH 预言机 $\mathcal{O}_{\mathrm{DIDH}}$，$\mathcal{B}$ 将借助 $\mathcal{A}$ 攻击 SM9-mIBE 方案的 ID-RCCA 优势计算 Gap-q_H-BCAA1 问题的解，即 $e(g,h)^{\frac{s}{s+z_0}}$. 假设攻击者 $\mathcal{A}$ 查询哈希函数 H 的次数最多为 q_H+1 次(包括攻击者通过仲裁私钥或用户私钥查询时访问 H 的情况).

$\mathcal{B}$ 通过如下方式模拟 $\mathcal{A}$ 交互的 ID-RCCA 游戏环境.

初始化阶段. $\mathcal{B}$ 利用 Gap-q_H-BCAA1 问题的实例，建立 SM9-mIBE 的主公钥 $mpk = (\mathcal{G}, w, u, H, hid)$，其中 $\mathcal{G} = (e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g, h)$ 是双线性配对群，$w = g^s$ 且 s 未知，$u = e(w, h)$. $\mathcal{B}$ 将 mpk 发送给 $\mathcal{A}$.

在进行询问回答之前，$\mathcal{B}$ 随机猜测攻击者 $\mathcal{A}$ 的类型. 根据不同类型的攻击者，$\mathcal{B}$ 模拟 ID-RCCA 的内部随机数选取方式不同，但是 $\mathcal{A}$ 看到的外部环境一样. 由于两种类型的攻击者是不可能同时出现的，从而保证 $\mathcal{B}$ 以至少 1/2 的概率成功猜测出攻击者 $\mathcal{A}$ 的类型.

询问阶段 1. $\mathcal{B}$ 随机猜测 $1 \le i^* \le q_H + 1$，作为挑战标识 id^* 出现的位置. 当 $\mathcal{A}$ 是类型 I 的攻击者时，$\mathcal{B}$ 回答 $\mathcal{A}$ 询问的方式如下：

(1) Hash 查询 $\mathcal{O}_H(id_i)$. $\mathcal{B}$ 建立一个形如 (id_i, z_i, θ_i) 的哈希列表 $\mathcal{L}_H$. 当 $\mathcal{A}$ 询问标识 id_i 的哈希值时，$\mathcal{B}$ 首先检查 $\mathcal{L}_H$ 是否存在形如 (id_i, z_i, θ_i) 的三元组.

1) 若存在，则 $\mathcal{B}$ 将 z_i 返回给 $\mathcal{A}$.

2) 若不存在，

2.1) 如果 id_i 是第 i^* 次询问，则 $\mathcal{B}$ 将三元组 $(id_{i^*}, z_0, \perp)$ 存储到列表中，并将 z_0 返回给 $\mathcal{A}$.

2.2) 否则，$\mathcal{B}$ 从 Gap-q_H-BCAA1 问题实例中选择 z_i 作为查询结果返回给 $\mathcal{B}$，并将 $(id_{i^*}, z_i, h^{\frac{s}{s+z_i}})$ 添加至列表中.

(2) KDF 查询 $\mathcal{O}_{\mathrm{KDF}}(t_i, id_i, l_i)$. $\mathcal{B}$ 建立一个形如 (t_i, id_i, K_i) 的列表 $\mathcal{L}_{\mathrm{KDF}}$. 当 $\mathcal{A}$ 询问 (t_i, id_i, l_i) 的 KDF 输出结果时，$\mathcal{B}$ 按如下方式返回相应的 K_i 给 $\mathcal{A}$.

1) $\mathcal{B}$ 首先查询列表 $\mathcal{L}_{\mathrm{KDF}}$ 中是否包含元素 (t_i, id_i, K_i). 若存在，当 $|K_i| \ge l_i$ 时，返回 K_i 的前 l_i 比特. 否则，随机选取 $l_i - |K_i|$ 长的比特串 K，将 $K_i \| K$ 返回给 $\mathcal{A}$，并将 (t_i, id_i, K_i) 替换为 $(t_i, id_i, K_i \| K)$.

2) 若不存在，则 $\mathcal{B}$ 询问 $\mathcal{O}_H(id_i)$ 对应的哈希列表元素 (id_i, z_i, θ_i). 根据 θ_i 的取值，$\mathcal{B}$ 进行如下回答：

2.1) 若 $\theta_i = \perp$，则对于列表 $\mathcal{L}_D$(见用户解密查询)中的每个元素 (C'_{i1}, id_i, K_i)，$\mathcal{B}$ 计算 $A = (C'_{i1})^{1/\alpha}$，并通过预言机 $\mathcal{O}_{\mathrm{DIDH}}$ 判断 $(u, g_T^{s+z_0}, A, t_i)$ 是否构成 DIDH 对.

如果 $\mathcal{L}_D$ 中存在元素 (C'_{i1}, id_i, K_i) 使得 $\mathcal{O}_{\mathrm{DIDH}}$ 返回 1)，则 $\mathcal{B}$ 将相应的 K_i 取回并将 (t_i, id_i, K_i) 添加到 $\mathcal{L}_{\mathrm{KDF}}$ 列表中. 然后，$\mathcal{B}$ 按照查询列表 $\mathcal{L}_{\mathrm{KDF}}$ 中存在形如 (t_i, id_i, K_i) 元素的方式将相应的 K_i 返回给 $\mathcal{A}$.

否则，$\mathcal{B}$ 随机选取长度为 l_i 的比特串 K_i，将 K_i 返回给 $\mathcal{A}$，并将 (t_i, id_i, K_i) 添加至列表 $\mathcal{L}_{\mathrm{KDF}}$.

2.2) 若 $\theta_i \ne \perp$，$\mathcal{B}$ 随机选取长度为 l_i 的比特串 K_i，并将 (t_i, id_i, K_i) 添加至列表 $\mathcal{L}_{\mathrm{KDF}}$.

(3) 仲裁私钥查询 $\mathcal{O}_{\mathrm{SKey}}(id_i)$. $\mathcal{B}$ 建立两个初始为空集的列表 T_1 和 T_2. 当 $\mathcal{A}$ 要查询第 i 个标识 id_i 的仲裁私钥时，$\mathcal{B}$ 首先查找列表 T_1 中是否存在元素 (id_i, sk_{sem}).

1) 若存在，则将 sk_{sem} 返回给攻击者.

2) 若不存在，$\mathcal{B}$ 首先查询列表 $\mathcal{L}_H$. 如果 θ_i 不存在列表中，则 $\mathcal{B}$ 询问 $\mathcal{O}_H(id_i)$.

2.1) 若 $\theta_i \ne \perp$，$\mathcal{B}$ 随机选择 $a \in \mathbb{Z}_p$，令 $sk_{sem} = (\theta_i)^{\frac{1}{a}}$，$sk_{user} = a$. $\mathcal{B}$ 将 sk_{sem} 返回给 $\mathcal{A}$，并将 (id_i, sk_{sem}) 和 (id_i, sk_{user}) 分别添加至列表 T_1 和 T_2 中.

2.2) 若 $\theta_i = \perp$，则 $\mathcal{B}$ 随机选择一个元素 $\alpha \in \mathbb{Z}_p$，计算 $sk_{sem} = h^{\alpha}$. $\mathcal{B}$ 将 sk_{sem} 返回给 $\mathcal{A}$，并将 (id_i, sk_{sem}) 和 $(id_i, \perp)$ 分别添加至列表 T_1 和 T_2.

(4) 用户私钥询问 $\mathcal{O}_{\mathrm{UKey}}(id_i)$. 当 $\mathcal{A}$ 要查询第 i 个标识 id_i 的用户私钥时，$\mathcal{B}$ 先查找列表 T_2 中是否存在形如 (id_i, sk_{user}) 的元素. 如果不存在，$\mathcal{B}$ 按照仲裁私钥查询方式，建立标识 id_i 相应的仲裁私钥

(id_i, sk_{sem})和用户私钥(id_i, sk_{user}). 若 $sk_{user}=\perp$,则$\mathcal{B}$终止游戏(记作事件 E_1). 否则$\mathcal{B}$将 sk_{user}返回给$\mathcal{A}$.

(5)服务器解密查询$\mathcal{O}_{\mathrm{SDec}}(id_i, C_i)$. 由于$\mathcal{B}$可以模拟任意标识的仲裁私钥,故当$\mathcal{A}$查询$(id_i, C_i)$的服务器解密结果时,$\mathcal{B}$可以执行算法 $\mathrm{SDec}(id_i, T_1, C_i)$并将计算结果发送给$\mathcal{A}$.

(6)用户解密查询$\mathcal{O}_{\mathrm{UDec}}(id_i, C_i')$. $\mathcal{B}$维护一个形如(C_{i1}', id_i, K_i)的列表$\mathcal{L}_D$. 当$\mathcal{A}$查询 $C_i'=(C_{i1}', C_{i2}, C_{i3})$的用户解密结果时,$\mathcal{B}$首先查询列表 T_2 中是否存在标识 id_i的用户私钥 sk_{user}. 若不存在,则通过询问$\mathcal{O}_{\mathrm{UKey}}(id_i)$建立相应的用户私钥.

1)若 $sk_{user}\neq\perp$,则$\mathcal{B}$可以利用该密钥计算出 $t_i=(C_{i1}')^{*_{user}}$,通过询问$\mathcal{O}_{\mathrm{KDF}}(t_i, id_i, l_i)$得到中间密钥 K_i.

2)若 $sk_{user}=\perp$,表示$\mathcal{A}$查询挑战标识的解密结果. 首先,$\mathcal{B}$查询列表$\mathcal{L}_D$中是否存在索引为(C_{i1}', id_i)的元素.

2.1)若存在,则$\mathcal{B}$选取相应的 K_i并取其前$|C_2|+v$ 比特或者将其扩展至$|C_2|+v$ 比特.

2.2)若不存在,则对于列表$\mathcal{L}_{\mathrm{KDF}}$中的每个元素$(t_i, id_i, K_i)$,$\mathcal{B}$计算 $A=(C_{i1}')^{1/\alpha}$,并通过预言机$\mathcal{O}_{\mathrm{DIDH}}$判断$(u, g_T^{s+z_0}, A, t_i)$是否构成 DIDH 对. 如果$\mathcal{O}_{\mathrm{DIDH}}$返回 1),则$\mathcal{B}$将相应的 K_i取回并将(C_{i1}', id_i, K_i)添加至列表$\mathcal{L}_D$;否则,$\mathcal{B}$随机选取一个长为$|C_{i2}|+v$ 比特的 K_i,并将(C_{i1}', id_i, K_i)添加至列表$\mathcal{L}_D$.

3)最后,$\mathcal{B}$利用中间密钥 K_i从(C_{i2}, C_{i3})中计算出消息 M. 若 M 等于挑战消息M_0或M_1,则返回$\perp$;否则,返回消息 M.

挑战. $\mathcal{A}$提交一个挑战标识 id^*和两个等长的消息 M_0和 M_1时,若$\mathcal{B}$从未进行id^*的哈希询问,则$\mathcal{B}$首先询问$\mathcal{O}_H(id^*)$并获取元素(id^*, z^*, θ^*). 若 $\theta^*\neq\perp$,则$\mathcal{B}$终止游戏(记作事件 E_2). 否则,$\mathcal{B}$随机选择一个比特 $b\in\{0,1\}$. 接下来,$\mathcal{B}$随机选择 $K^*=K_1^*\|K_2^*\in\{0,1\}^{|M_b|+v}$以及 $y\in\mathbb{Z}_p$,计算 $C_1^*=g^y$,$C_2^*=K_1^*\oplus M_b$和 $C_3^*=H_v(C_2^*\|K_2^*)$. 最后,$\mathcal{B}$将挑战密文 $C^*=(C_1^*, C_2^*, C_3^*)$发送给$\mathcal{A}$.

询问阶段 2. $\mathcal{B}$按照阶段 1 的方式回答$\mathcal{A}$的各种询问. 但是,$\mathcal{A}$不能查询挑战标识 id^*的用户私钥.

猜测. 最终,$\mathcal{A}$输出一比特 b'. $\mathcal{B}$按如下方式回答 Gap-q_H-BCAA1 问题的解:

(1)对于$\mathcal{L}_{\mathrm{KDF}}$中的每个元素$(t_j, id^*, K_j)$,$\mathcal{B}$利用预言机$\mathcal{O}_{\mathrm{DIDH}}$判断$(u, g_T^{s+z_0}, A, t_j)$是否构成 DIDH 对. 若$\mathcal{O}_{\mathrm{DIDH}}$输出 1,则$\mathcal{B}$返回$(t_j)^{1/y}$作为 Gap-$q_H$-BCAA1 问题的解.

(2)如果$\mathcal{L}_{\mathrm{KDF}}$中不存在上述元素,则$\mathcal{B}$输出$\perp$(记作事件 E_3).

当$\mathcal{A}$是类型 II 的攻击者时,$\mathcal{B}$模拟$\mathcal{A}$询问的方式与前面的类似. 不同之处在于,对于$\mathcal{B}$随机猜测 $1\leq i^*\leq q_H+1$ 位置的标识对应的用户私钥已知(随机选取),但是相应的仲裁私钥未知. 若$\mathcal{A}$询问挑战标识的仲裁私钥解密,由于该私钥已被撤销,所以$\mathcal{B}$直接返回$\perp$. 否则,$\mathcal{B}$知道其他仲裁私钥和所有的用户私钥,从而可以回答攻击者的解密查询.

引理 3. 若算法$\mathcal{B}$在上述游戏中不终止,即事件 E_1和事件 E_2都不发生,则$\mathcal{A}$交互的环境与游戏 Game_1定义的环境完全一致.

证明(引理 3). 首先,因为 Gap-q_H-BCAA1 问题实例$(g, h, g^s, z_0, (z_1, h^{\frac{s}{s+z_1}}), \cdots, (z_{q_H}, h^{\frac{s}{s+z_{q_H}}}))$中的 $z_i\in\mathbb{Z}_p$是随机选取且互不相同,所以算法$\mathcal{B}$模拟的哈希函数查询$\mathcal{O}_H(id_i)$的结果是均匀独立分布的. 当密钥派生函数 KDF 也看作是随机预言机时,我们只需要保证对于不同查询,$\mathcal{O}_{\mathrm{KDF}}$返回的结果随机且独立,并且询问相同,查询结果一致. 若游戏不终止,则挑战标识 id^*对应的用户私钥为 $sk_{id^*}=\frac{s}{\alpha(s+h_0)}$(对于$\mathcal{B}$未知). 那么,对于 KDF 返回结果的一致性,可分为两种情况:

(1)当攻击者查询$\mathcal{O}_{\mathrm{KDF}}(t_i, id_i, l_i)$时,如果列表$\mathcal{L}_{\mathrm{KDF}}$中不存在元素$(t_i, id_i, K_i)$,则当 $id_i=id^*$时,$\mathcal{B}$要保证返回结果与任意用户解密查询 $C_j'=(C_{j1}', C_{j2}, C_{j3})$的用户解密查询结果一致. $\mathcal{B}$通过访问 DIDH 预言机来判断$(C_{j1}')^{sk_{id^*}}=t_i$是否成立. 如果等式成立,则$\mathcal{B}$返回用户解密查询列表$\mathcal{L}_D$中$(C_{j1}', id_j, K_j)$对应的 K_i.

(2)当攻击者进行挑战标识 id^*和 $C_i'=(C_{i1}', C_{i2}, C_{i3})$的用户解密查询时,如果列表$\mathcal{L}_D$中不存在元素$(C_{i1}', id_i, K_i)$,尽管$\mathcal{B}$无法通过计算 $t_i=(C_{i1}')^{sk_{id^*}}$来询问$\mathcal{O}_{\mathrm{KDF}}(t_i, id^*, l_i)$. 但是,通过 DIDH 预言机,$\mathcal{B}$可以判定列表$\mathcal{L}_{\mathrm{KDF}}$中是否存在元素$(t_j, id_j, K_j)$满足 $t_i=t_j$,从而保证 KDF 查询的一致性.

对于其他查询,$\mathcal{B}$模拟的结果和游戏 Game_1中的定义完全一致. 因此,若$\mathcal{B}$在上述游戏中不终止,则模拟的环境和游戏 Game_1一致. 引理 3 证毕.

令 S 表示$\mathcal{B}$模拟的游戏中的事件 $b'=b$. 根据引理 1，则有

$$S_1=S\wedge(\bar{E}_1\wedge\bar{E}_2).$$

当事件 E_3 发生时，根据随机预言机的性质，游戏 Game_0 和游戏 Game_1 等价，从而有

$$S_1\wedge E_3=S_0\wedge E_3.$$

根据差分引理[34]，有

$$|\Pr[S_1]-\Pr[S_0]|\leq\Pr[\bar{E}_3].$$

根据事件 E_1 和 E_2 的定义，显然若 E_2 不发生，则 E_1 必然不发生. 由于$\mathcal{B}$成功猜测挑战标识出现位置 i^* 的概率至少为$\frac{1}{q_H+1}$，所以 $\Pr[\bar{E}_2]\geq\frac{1}{q_H+1}$. 结合$\mathcal{B}$正确猜测攻击类型的概率，$\mathcal{B}$解决 Gap-$q_H$-BCAA1 问题的概率则为

$$\Pr[\mathcal{B}\text{成功}]=\frac{1}{2}\Pr[\bar{E}_2\wedge\bar{E}_3]\geq\frac{|\Pr[S_1]-\Pr[S_0]|}{2(q_H+1)}.$$

由上式可得：

$$|\Pr[S_1]-\Pr[S_0]|\leq 2(q_H+1)Adv_{\mathcal{B}}^{\text{Gap-}q_H\text{-BCCA1}}(\lambda).$$

引理 1 证毕.

证明(引理 2).　我们通过构造一个概率多项式算法$\mathcal{C}$模拟攻击者$\mathcal{A}$在游戏 Game_2 或 Game_1 中的环境来证明两个游戏不可区分.

系统初始化及各种查询. 算法$\mathcal{C}$按照游戏 Game_1 中的方式生成系统参数. 故$\mathcal{C}$知道系统的主公钥 mpk 和主私钥 msk. 利用主私钥，$\mathcal{C}$可以按照游戏 Game_1 中的方式回答任意标识 id 的仲裁私钥 sk_{sem} 和用户私钥 sk_{user} 查询. 此外，$\mathcal{C}$将哈希函数 H 和密钥派生函数 KDF 看作随机预言机来回答攻击者的询问. 除此之外，$\mathcal{C}$按如下方式回答 H_v 的哈希查询和解密查询.

(1) Hash 查询 $\mathcal{O}_{H_v}(C_{2i},K_{2i})$. 对于哈希函数 H_v，$\mathcal{C}$按照如下方式回答$\mathcal{A}$的查询：$\mathcal{C}$维护一个形如 (C_{2i},K_{2i},C_{3i}) 的列表 $\mathcal{L}_{H_v}$. 当$\mathcal{A}$询问 (C_{2i},K_{2i}) 的哈希值时，$\mathcal{C}$查询列表 $\mathcal{L}_{H_v}$. 如果存在，则返回相应的结果 C_{3i}；若不存在，则$\mathcal{C}$随机选择 $C_{3i}\in\{0,1\}^v$，并将 (C_{2i},K_{2i},C_{3i}) 添加至列表 $\mathcal{L}_{H_v}$.

(2) 用户解密查询 $\mathcal{O}_{\text{UDec}}(id_i,C_i')$. 对于解密查询，由于$\mathcal{C}$知道任意标识的仲裁私钥和用户私钥. 因此，$\mathcal{C}$可以按照游戏 Game_1 中的方式回答攻击者的任意解密查询(服务器解密和用户解密). 但是，当$\mathcal{A}$查询标识 id^* 的密文 $C_i'=(C_{i1}',C_{i2},C_{i3})$ 的用户解密服务时，如果 $C_{i1}'=e(C_1^*,sk_{sem}^*)$，其中 sk_{sem}^* 表示挑战标识 id^* 对应的仲裁私钥，则$\mathcal{C}$直接返回 $\perp$ 给$\mathcal{A}$.

挑战. 当攻击者提交挑战标识 id^* 和两个等长消息 M_0 和 M_1 时，$\mathcal{C}$随机选择 $b\in\{0,1\}$，$r^*\in\mathbb{Z}_p$，以及长度分别为 $|M_b|$ 比特和 v 比特的随机比特串 K_1^* 和 K_2^*. 然后，$\mathcal{C}$计算

$$C_1^*=g^{r^*(s+H(id^*))},$$
$$C_2^*=K_1^*\oplus M_b,$$
$$C_3^*=H_v(C_2^*\|K_2^*),$$

$\mathcal{C}$将 $C^*=(C_1^*,C_2^*,C_3^*)$ 作为挑战密文返回给攻击者.

猜测. 最终，$\mathcal{A}$输出一比特 b' 作为对 b 的猜测.

分析. 显然，$\mathcal{C}$模拟的上述环境和攻击者在游戏 Game_2 看到的环境是一致的.

在挑战密文中，我们隐含假设 (u^{r^*},id^*) 的 KDF 询问结果为 $K_1^*\|K_2^*$. 在游戏 Game_1 中，如果挑战标识 id^* 的密文 $C_i'=(C_{i1}',C_{i2},C_{i3})$ 满足 $C_{i1}'=e(C_1^*,sk_{sem}^*)$，则$\mathcal{C}$应使用 K_1^* 和 K_2^* 从 (C_{i2},C_{i3}) 中解密出消息 M 并返回给$\mathcal{A}$. 令事件 F 表示在上述用户解密查询时 $M\neq\perp$. 显然，当事件 F 不发生时，$\mathcal{C}$模拟的环境和游戏 Game_1 完全一样. 下面分析事件 F 不发法生的可能情形：

(1) 当 $(C_{i2},C_{i3})=(C_2^*,C_3^*)$ 时，由于解密结果 $M\in\{M_0,M_1\}$，根据游戏规则，在 Game_1 和 Game_2 中，$\mathcal{C}$都返回 $\perp$.

(2) 当 $C_{i2}=C_2^*$ 且 $C_{i3}\neq C_3^*$ 时，根据哈希函数 H_v 的一致性，则有 $H_v(C_2^*\|K_2^*)=C_3^*\neq C_{i3}$. 故在 Game_1 和 Game_2 中，$\mathcal{C}$都返回 $\perp$.

(3) 当 $C_{i2}\neq C_2^*$ 时，由于 K_2^* 对于$\mathcal{A}$在信息论意义下是未知的，因此，在$\mathcal{A}$提交的哈希查询 (C_{2i},K_{2i}) 中，存在 $K_{2i}=K_2^*$ 的概率最多为$\frac{q_{H_v}}{2^v}$，其中 q_{H_v} 是攻击者进行哈希函数 H_v 查询的次数. 当$\mathcal{A}$从未查询 $\mathcal{O}_{H_v}(C_{i2},K_2^*)$ 时，$H_v(C_{i2}\|K_2^*)$ 的值是独立随机选取的. 故在 q_D 次解密查询中，存在 $H_v(C_{i2}\|K_2^*)=C_{i2}$ 的概率不超过$\frac{q_D}{2^v}$，即

$$\Pr[H_v(C_{i2}\|K_2^*)=C_{i2}]\leq\frac{q_D}{2^v}.$$

综上可得：

$$\Pr[F]\leq\frac{q_{H_v}}{2^v}+\frac{q_D}{2^v}.$$

因此，

$$|\Pr[S_1]-\Pr[S_2]|\leq\frac{q_{H_v}+q_D}{2^v}.$$

引理 2 证毕.

前向和后向安全性. 除了密钥撤销机制，前向安全性和后向安全性也是缓解密钥泄露危害的主要方法之一. 在本方案的安全模型中，假设用户的仲裁私钥和用户私钥不会同时泄露. 因此，部分密钥泄露不会危害其他时刻密钥的安全性，从而使得方案具有前向安全性和后向安全性. 例如，当用户私钥泄露后，通过更新仲裁私钥和用户私钥，或撤销仲裁私钥的方式来撤销被泄露的用户私钥，使得泄露的用户私钥不再具有解密功能. 从而实现用户密钥的前向和后向安全性. 若本方案的仲裁私钥和用户私钥同时泄露，则攻击者可以恢复出 SM9-IBE 用户的完整密钥，从而解密该用户的任意密文. 此时，该方案既不具备前向安全性也不具备后向安全性.

解密权限代理. 通过更新仲裁私钥，本方案还具有一定的代理解密功能. 例如，当用户 Bob 出差时，由于在外部环境不方便使用私钥或可能增加私钥泄露的风险，可以将私钥授权给秘书 Alice 行使解密权限. 待 Bob 出差回来后，通过更新仲裁私钥和用户私钥，来撤销 Alice 的解密权限.

5 性能分析

下面从理论和实验两个角度比较 SM9-mIBE 算法与原始 SM9-IBE 算法[8]和 Sun 等人[11]的服务器辅助可撤销 SM9-IBE 算法(简称为 SM9-SR-IBE)的性能. 选择使用对称双线性配对，即 $e:\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$.

表 1 比较了三个算法的参数大小和算法主要运算次数. 在表中，N、R 和 n 分别表示系统用户的数量、被撤销用户的数量和用户的密文数量；“Exp.”和“Pairing”分别表示一次群元素的指数运算和配对运算. 在比较各算法所需运算操作次数时，忽略了除指数运算和配对运算之外的其他操作. 表中符号“—”表示该项不存在，“0”表示该项几乎不需要任何操作.

表 1 性能比较

	SM9-IBE 文献[8]	SM9-SR-IBE 文献[11]	SM9-mIBE 第 3.2 节
系统公钥	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$
系统私钥	$\|p\|$	$\|p\|$	$\|p\|$
服务器密钥	—	256 比特	$\|\mathbb{G}\|$
用户密钥	$\|\mathbb{G}\|$	$\|\mathbb{G}\|$	$\|p\|$
密文	$\|\mathbb{G}\|$	$\|\mathbb{G}\|$	$\|\mathbb{G}\|$
服务器密钥生成时间	—	0	1—Exp.
用户密钥生成时间	1—Exp.	1—Exp.	0
密钥/密文更新时间	—	nR—Exp.	$R\cdot$ Exp.
加密时间	3—Exp.	3—Exp.	3—Exp.
服务器解密时间	—	1—Exp.	1—Pairing
用户解密时间	1—Pairing	1—Pairing	1—Exp.

通过表 1 的比较可以看出，SM9-SR-IBE 算法和 SM9-mIBE 算法保持了原始 SM9-IBE 算法的系统公钥和系统私钥的选取方式，以及加密算法的密文生成方式. 在 SM9-SR-IBE 算法中，服务器的私钥是 256 比特的伪随机函数，即 HMAC 的密钥长度，而在 SM9-mIBE 算法中，服务器的私钥实际上是盲化的 SM9-IBE 用户密钥，即群$\mathbb{G}$中的一个元素. 对于本地用户私钥，SM9-SR-IBE 算法和原始 SM9-IBE 算法的本地用户私钥是一致的，但是 SM9-mIBE 算法的本地用户私钥是 $\mathbb{Z}_p$ 中的一个元素. 当系统中有 R 个用户密钥暴露时，SM9-SR-IBE 算法需要更新这 R 个用户的所有密文，从而需要 nR 次群元素的指数运算，而本文算法仅需要 R 次操作. 对于用户解密算法，其他两个算法都需要一次双线性配对运算，而本文仅需要一次指数运算. 一般来讲，配对运算比指数运算耗时要多. 因此，本文算法比 Sun 等人的算法在用户解密和密钥更新等方面具有一定的优势.

通过编程实现，本文在同一环境下测试了各个平均执行时间. 测试环境如下：

CPU：2.4GHz Inteli5-9300H

内存：16.0GB

操作系统：Windows10 家庭中文版

本文主要依赖 JPBC 库对 SM9 算法进行实现，选取的椭圆曲线类型为 Type-F. 表 2 列出了各算法运行的平均时间.

表 2 平均执行时间比较

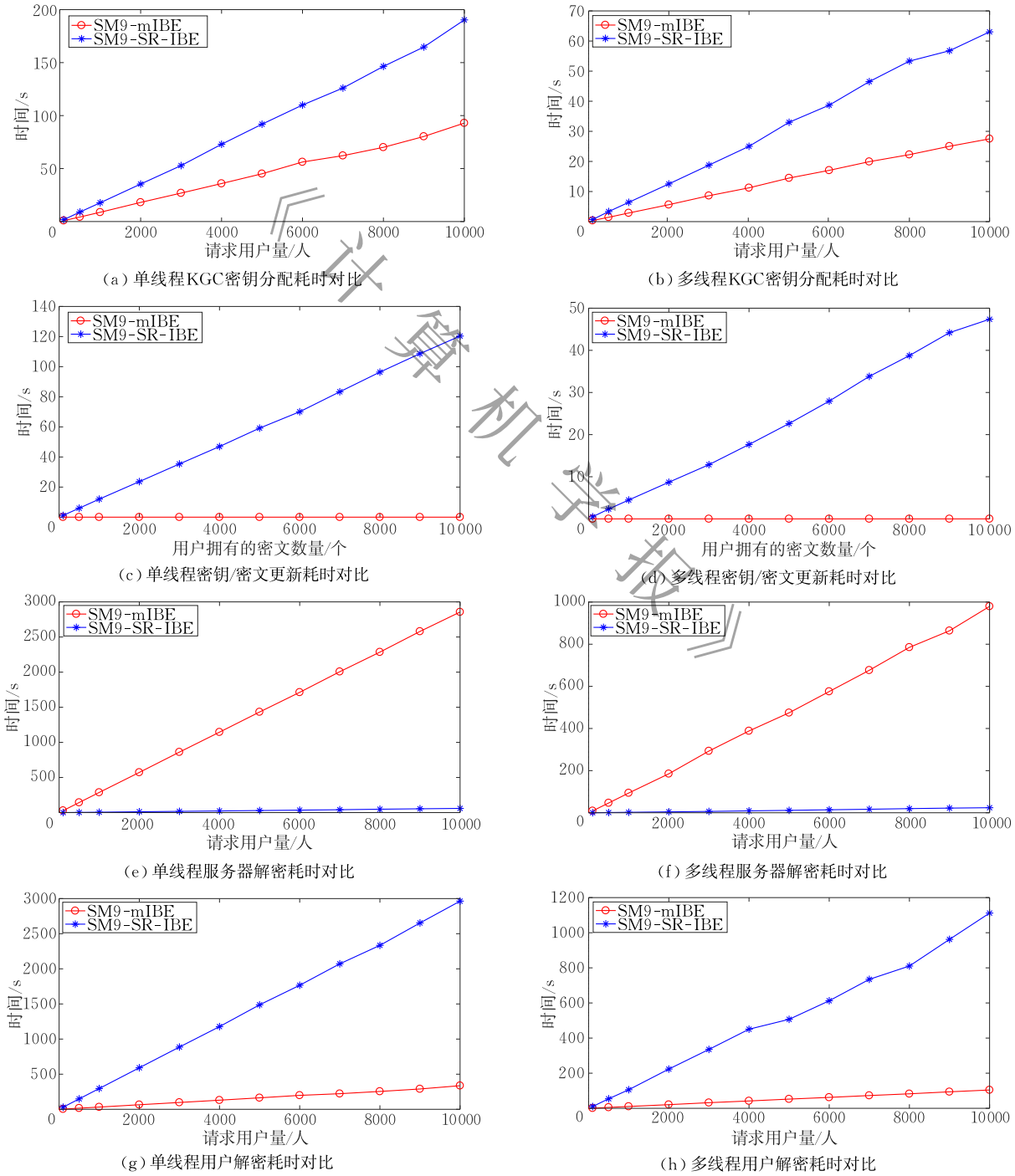
	SM9-IBE 文献[8]	SM9-SR-IBE 文献[11]	SM9-mIBE 第 3.2 节
服务器密钥	—	0	0.0088 s
用户密钥	0.0123 s	0.0178 s	0
密钥/密文更新	—	0.0118 s	0.0001 s
加密	0.3952 s	0.4091 s	0.3878 s
服务器解密	—	0.0061 s	0.2879 s
用户解密	0.3434 s	0.2994 s	0.0323 s

从表 2 可以看出，各算法的运行时间基本和原始 SM9-IBE 算法保持一致. 由于本文用户解密算法不需要配对运算，因此比其他两个算法快将近 10 倍.

为了更加清晰地描述 SM9-mIBE 与 SM9-SR-IBE 之间的差异，本文将两个算法根据具体功能划分为以下四个阶段：(1)KGC 密钥分配阶段：表 2 中的服务器密钥和用户密钥的生成阶段统称为 KGC

密钥分配阶段；(2) 密钥/密文更新阶段；(3) 服务器(部分)解密阶段；(4) 用户解密阶段. 本文将对两个算法的上述四个阶段分别进行测试和对比. 为了简单起见，在此基准测试中本文不考虑网络延迟和磁盘 I/O 时间，仅测量算法的处理时间. 本文将每个用户发送的邮件大小设置为 75 KB，并将 N 从 0 增加到 10 000，在第(1)，(3)，(4)阶段，N 表示请求的用户数量；在第(2)阶段，N 表示一个用户在服务器上拥有的密文数量. 因为在本方案中将 SM9-IBE 解密阶段分为服务器解密和用户解密两个阶段，方案虽然降低了用户解密的开销，但服务器解密阶段却给服务器带来了额外的消耗，为了更加充分利用服务器的计算能力，本文除在单线程的环境下进行测试以外，还调用了 Java 的线程池 ExecutorService 以支持多线程执行.

下面给出 SM9-mIBE 与 SM9-SR-IBE 这两个方案分别在这四个阶段处于单线程和多线程条件之下的耗时对比结果，如图 2 所示.



(a) 单线程KGC密钥分配耗时对比　(b) 多线程KGC密钥分配耗时对比

(c) 单线程密钥/密文更新耗时对比　(d) 多线程密钥/密文更新耗时对比

(e) 单线程服务器解密耗时对比　(f) 多线程服务器解密耗时对比

(g) 单线程用户解密耗时对比　(h) 多线程用户解密耗时对比

图 2　SM9-mIBE 与 SM9-SR-IBE 四个阶段耗时对比图

图 2(a)和图 2(b)分别给出了在单线程和多线程的环境下,两个算法在 KGC 密钥分配阶段所消耗的时间.从中可以看出,在此阶段,SM9-mIBE 相对于 SM9-SR-IBE 所消耗的时间略短.

图 2(c)和图 2(d)分别给出了在单线程和多线程的环境下,两个算法在密钥/密文更新阶段所耗费的时间.从图中可以看出,两者在该阶段的初始时期所消耗的时间相差无几.但是随着用户拥有密文数量的增长,SM9-SR-IBE 所消耗的时间呈线性增长,而 SM9-mIBE 耗时一直保持稳定.

图 2(e)和图 2(f)分别给出了在单线程和多线程的环境下,两个算法在服务器解密阶段所消耗的时间.由于 SM9-mIBE 在服务器解密的时候需要进行一次双线性对运算,而 SM9-SR-IBE 在该阶段只进行了一次指数运算.因此在该阶段,前者相对后者所消耗的时间较长.

图 2(g)和图 2(h)分别给出了在单线程和多线程环境下,两个算法在用户解密阶段所耗费的时间.因为 SM9-mIBE 在此阶段只进行了一次指数运算,而 SM9-SR-IBE 却需要进行一次双线性对运算.所以在此阶段,前者相对于后者有更快的效率,可以给予用户更高效的解密体验.

根据上述四个阶段的对比测试可以看出:一方面,SM9-mIBE 相比于 SM9-SR-IBE,在密钥/密文更新阶段不需要对密文进行更新,极大地节省了更新操作的时间;在用户解密阶段,极短的解密时间也能给用户带来良好的体验感.另一方面,虽然 SM9-mIBE 在服务器解密阶段相比于 SM9-SR-IBE 有一定的劣势,但这完全可以借助服务器强大的计算能力将这个时间尽可能缩短,控制在用户可接受的范围之内.

6 总 结

针对 SM9 标识加密算法存在的密钥撤销和更新问题,本文提出一种基于仲裁的 SM9 标识加密算法.通过辅助用户进行解密,仲裁服务器能够控制用户的访问权限,从而实现密钥撤销和更新机制.通过理论和实验分析表明,该方法比已有方法在密钥更新和本地用户解密等方面具有一定的效率优势.在后续工作中,存在以下几个问题值得进一步研究:(1)研究不依赖仲裁服务器的 SM9 标识加密算法的密钥撤销机制;(2)研究具有前向安全性和/或后向安全性的 SM9 标识加密算法.

参 考 文 献

[1] Shamir A. Identity-based cryptosystems and signature schemes //Proceedings of the Advances in Cryptology-Crypto'84. Santa Barbara, USA, 1984: 47-53

[2] Boneh D, Franklin M. Identity-based encryption from the Weil pairing//Proceedings of the Advances in Cryptology-Crypto 2001. Santa Barbara, USA, 2001: 213-229

[3] Cocks C. An identity-based encryption scheme based on quadratic residues//Proceedings of the IMA International Conference on Cryptography and Coding. Cirencester, UK, 2001: 360-363

[4] Hofheinz D, Jia D, Pan J. Identity-based encryption tightly secure under chosen-ciphertext attacks//Proceedings of the Advances in Cryptology-Asiacrypt2018. Brisbane, Australia, 2018: 190-220

[5] Langrehr R, Pan J. Tightly secure hierarchical identity-based encryption. Journal of Cryptology, 2020, 33(4): 1787-1821

[6] Sahai A, Waters B. Fuzzy identity-based encryption// Proceedings of the Advances in Cryptology-Eurocrypt 2005. Aarhus, Denmark, 2005: 457-473

[7] Waters B. Efficient identity-based encryption without random oracles//Proceedings of the Advances in Cryptology-Eurocrypt 2005. Aarhus, Denmark, 2005: 114-127

[8] Cheng Z H. The SM9 cryptographic schemes. IACR Cryptology ePrint Archive, 2017: 117-143

[9] Long Y, Xiong F. Collaborative generations of SM9 private key and digital signature using homomorphic encryption// Proceedings of the 2020 5th International Conference on Computer and Communication Systems (ICCCS). Shanghai (Virtual), China, 2020: 76-81

[10] Mu Y H, Xu H X, Li P L, et al. Secure two-party SM9 signing. Science China Information Sciences, 2020, 63(8): 239-241

[11] Sun S Z, Ma H, Zhang R, Xu W H. Server-aided immediate and robust user revocation mechanism for SM9. Cybersecurity, 2020, 3(1): 3-12

[12] Yang Ya-Tao, Cai Ju-Liang, Zhang Xiao-Wei, Yuan Zheng. Privacy preserving scheme in block chain with provably secure based on SM9 algorithm. Journal of Software, 2019, 30(6): 1692-1704(in Chinese)
(杨亚涛, 蔡居良, 张筱薇, 袁征. 基于 SM9 算法可证明安全的区块链隐私保护方案. 软件学报, 2019, 30(6): 1692-1704)

[13] Yao Ying-Ying, Chang Xiao-Lin, Zhen Ping. Decentralized identity authentication and key management scheme based on blockchain. Cyberspace Security, 2019, 10(6): 33-39 (in Chinese)
(姚英英, 常晓林, 甄平. 基于区块链的去中心化身份认证及密钥管理方案. 网络空间安全, 2019, 10(6): 33-39)

[14] Boldyreva A, Goyal V, Kumar V. Identity-based encryption with efficient revocation//Proceedings of the 15th ACM Conference on Computer and Communications Security. Alexandria, USA, 2008: 417-426

[15] Lee K, Lee D H, Park J H. Efficient revocable identity-based encryption via subset difference methods. Designs, Codes and Cryptography, 2017, 85(1): 39-76

[16] Libert B, Vergnaud D. Adaptive-ID secure revocable identity-based encryption//Proceedings of the Cryptographers' Track at the RSA Conference. San Francisco, USA, 2009: 1-15

[17] Seo J H, Emura K. Revocable identity-based cryptosystem revisited: Security models and constructions. IEEE Transactions on Information Forensics and Security, 2014, 9(7): 1193-1205

[18] Ishida Y, Shikata J, Watanabe Y. CCA-secure revocable identity-based encryption schemes with decryption key exposure resistance. International Journal of Applied Cryptography, 2017, 3(3): 288-311

[19] Watanabe Y, Emura K, Seo J H. New revocable IBE in prime-order groups: Adaptively secure, decryption key exposure resistant, and with short public parameters//Proceedings of the Cryptographers' Track at the RSA Conference. San Francisco, USA, 2017: 432-449

[20] Zhou Yan-Wei, Yang Bo, Xia Zhe, et al. Revocable identity-based encryption scheme with leakage-resilience. Chinese Journal of Computers, 2020, 43(8): 1534-1554(in Chinese)
(周彦伟，杨波，夏喆等. 抵抗泄露攻击的可撤销 IBE 机制. 计算机学报，2020，43(8)：1534-1554)

[21] Seo J H, Emura K. Revocable hierarchical identity-based encryption. Theoretical Computer Science, 2014, 542: 44-62

[22] Emura K, Seo J H, Youn T. Semi-generic transformation of revocable hierarchical identity-based encryption and its DBDH instantiation. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2016, 99-A(1): 83-91

[23] Katsumata S, Matsuda T, Takayasu A. Lattice-based revocable (hierarchical) IBE with decryption key exposure resistance//Proceedings of Public Key Cryptography(2). Beijing, China, 2019: 441-471

[24] Mao X P, Lai J Z, Chen K F, et al. Efficient revocable identity-based encryption from multilinear maps. Security and Communication Networks, 2015, 8(18): 3511-3522

[25] Seo J H, Emura K. Revocable hierarchical identity-based encryption via history free approach. Theoretical Computer Science, 2016, 615: 45-60

[26] Ma X C, Lin D D. Generic constructions of revocable identity-based encryption//Proceedings of the International Conference on Information Security and Cryptology. Seoul, South Korea, 2019: 381-396

[27] Li J, Li J W, Chen X F, et al. Identity-based encryption with outsourced revocation in cloud computing. IEEE Transactions on computers, 2013, 64(2): 425-437

[28] Nguyen K, Wang H, Zhang J. Server-aided revocable identity-based encryption from lattices//Proceedings of the International Conference on Cryptology and Network Security. Milan, Italy, 2016: 107-123

[29] Qin B D, Deng R H, Li Y J, et al. Server-aided revocable identity-based encryption//Proceedings of the European Symposium on Research in Computer Security. Vienna, Austria, 2015: 286-304

[30] Qin B D, Liu X M, Wei Z, et al. Space efficient revocable IBE for mobile devices in cloud computing. Science China Information Sciences, 2020, 63(3): 282-284

[31] Ding X H, Tsudik G. Simple identity-based cryptography with mediated RSA//Proceedings of the Cryptographers' Track at the RSA Conference. San Francisco, USA, 2003: 193-210

[32] Elashry I, Mu Y, Susilo W. Identity-based mediated RSA revisited//Proceedings of the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. Melbourne, Australia, 2013: 728-735

[33] Chen L Q, Cheng Z H. Security proof of Sakai-Kasahar's identity-based encryption scheme//Proceedings of the Cryptography and Coding. Cirencester, UK, 2005: 442-459

[34] Shoup V. Sequences of games: A tool for taming complexity in security proofs. IACR Cryptology ePrint Archive, 2004: 332

QIN Bao-Dong, Ph. D., professor. His current research interests include basic theory and application of public key cryptography.

ZHANG Bo-Xin, M. S. candidate. His research interest is public key cryptography.

BAI Xue, M. S. candidate. Her research interest is public key cryptography.

Background

SM9 is an industry standard for identity-based encryption algorithms issued by China in 2016. Because SM9 has a different mathematical structure (exponent inversion) from the existing revocable IBE schemes. Existing revocation mechanisms do not apply to SM9 well. This paper belongs to the field of key revocation and updating mechanism of SM9. So far, there is almost no research on this area. In 2020, Sun et al, initially studied the key revocation mechanism of SM9 algorithm and proposed a server assisted user key revocation and update technology. However, this technology has two problems. On one hand, this method needs to establish a secure channel between the data sender and the server to transmit the original ciphertext. Otherwise, the user can decrypt the original ciphertext directly, and hence losing the revocation functionality of the user's decryption ability. On the other hand, this method revokes the previous key by updating all the ciphertext, which makes the workload of the server increase linearly with the number of ciphertexts. The scheme proposed in this paper alleviates these two issues and user's decryption speed is 10 times faster than Sun et al.'s scheme.

This work was supported in part by the National Natural Science Foundation of China (No. 61872292) and the Fund of Science and Technology on Communication Security Laboratory (No. 6142103190101).