中国科学: 信息科学 2023 年 第 53 卷 第 3 期: 454–469
SCIENTIA SINICA Informationis
论文

身份基加密机制的挑战后泄露容忍性

周彦伟[1,2,3,4], 王兆隆[1,2], 乔子芮[1*], 杨波[1*], 顾纯祥[3], 夏喆[5], 张明武[4,6]

1. 陕西师范大学计算机科学学院, 西安 710062
2. 密码科学技术全国重点实验室, 北京 100878
3. 河南省网络密码技术重点实验室, 郑州 450040
4. 广西密码学与信息安全重点实验室, 桂林 541004
5. 武汉理工大学计算机科学与技术学院, 武汉 430070
6. 湖北工业大学计算机学院, 武汉 430068
* 通信作者. E-mail: qzr_snnu@163.com, byang@snnu.edu.cn

收稿日期: 2022–04–12; 修回日期: 2022–06–15; 接受日期: 2022–07–22; 网络出版日期: 2023–03–13

国家重点研发计划 (批准号: 2017YFB0802000)、国家自然科学基金 (批准号: U2001205, 62272287)、广西密码学与信息安全重点实验室开放课题 (批准号: GCIS202108) 和河南省网络密码技术重点实验室开放课题 (批准号: LNCT2021 A04) 资助项目

摘要 现有的多数抗泄露加密机制的研究均假设敌手的泄露是来自收到挑战密文之前, 并禁止敌手在挑战后进行泄露操作. 然而, 在现实中敌手往往是接触到密文数据后会通过各种手段获取相应密钥的泄露信息, 因此挑战后的泄露容忍性更符合实际环境的真实应用需求. 针对上述不足, 本文将对身份基加密 (identity-based encryption, IBE) 机制的挑战后泄露容忍性展开研究, 提出 IBE 机制熵泄露容忍性的属性要求和安全性定义; 并在状态分离模型中联合熵泄露容忍的 IBE 机制和二源提取器设计抗挑战后泄露攻击的 IBE 机制, 对上述构造在选择明文攻击下的安全性进行了形式化证明. 此外, 为了获得具有更优安全性的 IBE 机制, 在上述构造的基础上, 通过增加消息验证码设计选择密文攻击安全的挑战后泄露容忍的 IBE 机制.

关键词 挑战后泄露容忍性, 身份基哈希证明系统, 身份基加密机制, 熵泄露容忍性

1 引言

在现有密码机制的研究中, 普遍认为参与者的密钥、随机数等内部秘密信息对外界敌手而言是无法获知的; 然而, 随着冷启动、边信道攻击等泄露攻击方式的兴起, 为外部攻击者提供了获知用户秘密信息的有效方法, 现实环境中泄露攻击的存在导致在传统安全模型下被证明安全的密码原语不再拥有其所声称的安全性. 因此, 具备泄露容忍性的密码机制拥有更强的实用性. 近年来, 为缩小科学理论研究与实际应用需求间的差距, 多个抵抗泄露攻击的密码原语 [1~3] 相继被提出, 泄露容忍性要求即使敌

引用格式: 周彦伟, 王兆隆, 乔子芮, 等. 身份基加密机制的挑战后泄露容忍性. 中国科学: 信息科学, 2023, 53: 454–469, doi: 10.1360/SSI-2022-0148
Zhou Y W, Wang Z L, Qiao Z R, et al. After-the-fact leakage resilience in identity-based encryption (in Chinese). Sci Sin Inform, 2023, 53: 454–469, doi: 10.1360/SSI-2022-0148

手获得秘密信息的泄露, 密码原语依然保持其原有的安全性. 在抗泄露密码原语的安全模型中敌手能够选取任意高效可计算的泄露函数, 获得秘密状态在泄露函数作用下的函数值, 该函数值就是敌手获得的泄露信息.

在加密机制中, 敌手通过泄露函数来获得相应秘密信息的泄露, 由于敌手在获得挑战密文之后能够将消息编码到泄露函数中, 通过泄露询问协助其在安全性游戏中获胜, 因此现有加密机制 [1~3] 的抗泄露性研究中要求泄露信息是在敌手接触到挑战密文之前获得的, 即挑战密文生成之后不允许泄露的产生. 例如, 在公钥加密 (public-key encryption, PKE) 机制的选择明文攻击 (chosen-plaintext attacks, CPA) 安全性游戏和选择密文攻击 (chosen-ciphertext attacks, CCA) 安全性游戏中, 敌手在挑战阶段将向挑战者发送两个等长的明文消息 M_0 和 M_1, 获得挑战者返回的挑战密文 C_β (C_β 是关于消息 M_β ($\beta \leftarrow_R \{0,1\}$) 的加密密文), 一旦允许敌手在挑战之后进行泄露询问, 那么敌手将挑战密文 C_β 编码到泄露函数中, 通过进行泄露询问协助其输出对随机值 β 的正确猜测 β', 在这种情况下很难达到标准安全模型下的 CPA 或 CCA 安全性. 因此, 现有多数 PKE、身份基加密 (identity-based encryption, IBE) 机制的研究均禁止敌手获得挑战密文后再进行泄露询问.

在现实环境中, 攻击者往往是先接触到加密的密文数据后才会通过各种手段获取相应密钥的泄露信息, 现有抵抗挑战前泄露攻击的加密机制 [1~3] 无法保证上述攻击下加密数据的保密性. 为使抗泄露的密码原语在现实环境中更加实用, 需研究加密机制挑战后的泄露容忍性, 允许敌手获得挑战密文之后继续进行泄露询问, 在此前提下依然保持加密机制的原始安全性, 使抗泄露密码学的理论研究更具有实用性. 针对上述研究目标, 本文以 IBE 机制为研究对象, 探讨其挑战后泄露容忍性的实现方法.

1.1 相关工作

Halevi 和 Lin [4] 对 PKE 机制挑战后的泄露容忍性进行了研究, 证明了 Naor 和 Segev [5] 提出的抗泄露 PKE 机制的通用构造具有熵泄露容忍性; 在状态分离模型下提出了具有挑战后泄露容忍性的 PKE 机制, 然而上述构造仅具有 CPA 安全性. 为了获得 PKE 机制挑战后泄露容忍的 CCA 安全性, Zhang 等 [6] 基于双加密 (double encryption) 技术提出了状态分离模型下的混合加密框架, 对上述构造的 CCA 安全性进行了证明; 此外, 文献 [6] 还对 IBE 机制的挑战后泄露容忍性进行了简单探索, 但相应的方案仅具有 CPA 安全性, 且需维护底层 IBE 机制的两套初始化系统, 使得相应构造的公开参数较长, 导致该机制的存储效率较低, 实用性较差. 现有 CCA 安全的 PKE 机制通常基于非交互式零知识 (non interactive zero knowledge proofs, NIZK) 系统实现挑战后的泄露容忍性, 导致相应构造的计算效率较低, 方案的实例化难度大. 针对上述不足, Zhao 等 [7] 首先在状态分离模型下基于抗泄露 PKE 机制给出了抗泄露损耗陷门函数的具体构造, 然后进一步在不使用 NIZK 的前提下实现了 PKE 机制的挑战后泄露容忍性, 该方案以较高的计算效率达到了 CCA 安全性. 文献 [8] 将挑战后的泄露容忍性和抵抗篡改攻击的需求结合起来, 设计了同时抵抗上述两种攻击的 CCA 安全的 PKE 机制, 解决了文献 [9] 所提出的公开问题. 文献 [10] 提出了一个通用转换方法, 能够基于任意泄露容忍的 CPA 安全的 PKE 机制构造出具有挑战后泄露容忍 CCA 安全性的 PKE 机制, 且新构造与底层抗泄露 PKE 具有相同的泄露率. 在状态分离模型中, 基于任意 CCA 安全的 PKE 机制, 文献 [11] 设计了具有多挑战的 CCA 安全性, 且抵抗挑战后泄露容忍的 PKE 机制. 此外, 文献 [12,13] 对抵抗挑战后泄露攻击的密钥协商协议进行了研究, 对相关构造在相应的安全性模型下进行了形式化证明.

近年来, 多个抗泄露 IBE 机制的具体构造 [14~16] 相继被研究者提出, 然而, 上述构造主要考虑了对挑战前泄露攻击的抵抗, 未涉及挑战后泄露容忍性的相关研究. 为实现 IBE 机制对挑战后泄露攻击的抵抗, 文献 [17] 提出了抵抗挑战后辅助泄露的 IBE 机制, 基于合数阶双线性群下的相关困难性假设

对方案的安全性进行了证明, 但分析发现该方案仅具有 CPA 安全性, 且计算和存储效率较低.

由上述现状分析可知, 公钥加密机制的挑战后泄露容忍性自文献 [4] 提出以来, 截至目前相关的研究 [4~13] 较少, 主要集中在 PKE 机制 [4~11] 和密钥协商协议 [12,13] 等领域, 特别是缺乏对 IBE 机制挑战后泄露容忍性的研究工作, 因此本文将对 IBE 机制抵抗挑战后泄露攻击的能力展开研究.

1.2 本文的思路

挑战后泄露容忍的实质是要求挑战密文具有足够的熵, 使得在挑战后的泄露询问中, 敌手获得的关于秘密信息的泄露不影响挑战密文中明文消息的随机性. 为实现对挑战后泄露攻击的抵抗, 本文首先提出 IBE 机制熵泄露容忍性的概念, 给出具体的性质要求和安全定义. 在熵泄露容忍的 IBE 机制中, 即使敌手根据挑战密文设计相应的泄漏函数, 但是它无法从泄露信息中获知比其长度更多的明文信息; 也就是说, 敌手即便拥有挑战后泄露攻击的能力, 只要挑战密文具有足够的最小熵, 就能确保挑战后泄露询问被敌手执行结束后明文消息依然具有一定的最小熵, 意味着该消息具有一定的随机性. 特别地, IBE 机制的熵泄露容忍性将传统抗泄露 IBE 机制关注密钥的剩余熵问题扩展到对明文消息剩余熵的讨论. 在此基础上, 基于熵泄露容忍的 IBE 机制, 联合二源提取器和消息验证码, 在状态分离模型中开展关于 IBE 机制挑战后泄露容忍的 CPA 安全性和 CCA 安全性的相关研究工作.

(1) 抗挑战后泄露攻击的 CPA 安全性. 在状态分离模型中, 本文将用户的密钥划分为相互独立的两个状态, 允许敌手分别对两个状态进行多项式次挑战前和挑战后的泄露询问, 其中部分密钥的相互独立性确保相关泄露信息同样是相互独立的. 相较于文献 [6] 中建立两套底层 IBE 机制而言, 本文仅运行一套熵泄露容忍的 IBE 机制, 通过两个安全的哈希函数 H_1 和 H_2 将用户身份 id 映射出两个相互独立的新身份 id_1 和 id_2, 基于底层 IBE 机制的密钥生成算法输出上述新身份 id_1 和 id_2 所对应的密钥 d_1 和 d_2, 由身份 id_1 和 id_2 的相互独立性, 可知密钥 d_1 和 d_2 同样是相互独立的, 满足状态分离模型下 IBE 机制的性质要求. 此外, 基于均匀选取的随机数 x_1 和 x_2, 使用二源提取器的输出 $\text{2-Ext}\,(x_1, x_2)$ 对明文消息 M 进行隐藏, 即 $\text{2-Ext}\,(x_1, x_2) \oplus M$, 底层熵泄露容忍的 IBE 机制确保即使敌手获知 x_1 和 x_2 的对应密文和用户密钥的泄露信息, x_1 和 x_2 依然具有足够的最小熵; 也就是说, 即使敌手获得相应的挑战后泄露, $\text{2-Ext}\,(x_1, x_2)$ 依然能够很好地隐藏明文消息 M.

(2) 抗挑战后泄露攻击的 CCA 安全性. 使用两个随机数作为二源提取器的输入, 使用对应的输出加密相应的明文消息. 然后, 采用熵泄露容忍的 IBE 机制对明文加密操作所使用的两个随机数分别进行加密, 由其所具备的熵泄露容忍性保证在有挑战后泄露的情况下原始随机数依然具有一定的随机性, 随机数在有挑战后泄露攻击的情况下依然保持着足够的平均最少熵, 因此由二源提取器的安全性可知, 加密明文消息的密钥与均匀随机值是不可区分的, 确保了最终的密文消息能够抵抗挑战后的泄露攻击. 更具体地讲, 通过哈希函数 H_1 和 H_2 对身份 id 进行映射生成新的身份 id_1 和 id_2, 使用熵泄露容忍的 IBE 机制对均匀选取的随机数 x_1 和 x_2 进行加密, 分别生成 $c_1 = \mathrm{Enc}\,(\mathrm{id}_1, x_1)$ 和 $c_2 = \mathrm{Enc}\,(\mathrm{id}_1, x_1)$; 然后基于两个不同的二源提取器 2-Ext_1 和 2-Ext_2 输出两个相互独立的对称密钥 $k_1 = \text{2-Ext}_1\,(x_1, x_2)$ 和 $k_2 = \text{2-Ext}_2\,(x_1, x_2)$, 其中一个对称密钥 k_1 完成对明文消息 M 的加密生成相应的密文 $c_3 = k_1 \oplus M$, 另外一个对称密钥 k_2 作为消息验证码 (message authentication code, MAC) 的密钥, 通过将 k_2 输入消息验证码的标签生成算法产生消息 (c_1, c_2, c_3) 的相应标签 $\mathrm{Tag} = \mathrm{Tag}\,(k_2, (c_1, c_2, c_3))$, 实现密文的防扩展性; 除接收者之外的任何敌手, 要想获知对称密钥 k_1 和 k_2, 其必须掌握接收者的密钥 $\mathrm{sk}_{\mathrm{id}}$ 才能从相应的密文中解密获得, 由密钥 $\mathrm{sk}_{\mathrm{id}}$ 的安全性保证了本文构造的安全性. 特别地, 本文方案中, 用户的密钥 $\mathrm{sk}_{\mathrm{id}}$ 实际是对随机数 x_1 和 x_2 提供了保护, 且熵泄露容忍性确保私钥存在泄露的情况下相应的密文 c_1 和 c_2 具有足够的随机性, 进而保证解密 c_1 和 c_2 所得到的结果依然具有足够的平均最小熵, 确

保能够基于二源提取器 2-Ext_1 和 2-Ext_2 分别恢复出加密明文 M 的一次性密钥 k_1 和消息验证码的对称密钥 k_2.

1.3 本文的贡献

针对 IBE 机制挑战后泄露容忍性研究缺乏的现状, 本文对 IBE 机制挑战后泄露容忍的 CPA 和 CCA 安全性分别进行了研究. 本文的主要工作分为以下三个方面, 详细叙述如下.

(1) 提出 IBE 机制熵泄露容忍性的安全属性要求及游戏模型, 并证明了基于身份基哈希证明系统 (identity-based hash proof system, IB-HPS) 和强随机性提取器所构造的 CPA 安全的 IBE 机制 [18] 具备本文所定义的熵泄露容忍性.

(2) 在状态分离模型中, 基于熵泄露容忍的 IBE 机制和二源提取器实现 IBE 机制的挑战后泄露容忍性, 并基于底层 IBE 机制的熵泄露容忍性和二源提取器的安全性证明上述通用构造具有抗挑战后泄露攻击的 CPA 安全性; 即使敌手拥有获得挑战后泄露信息的能力, 底层 IBE 机制的熵泄露容忍性能够确保用于隐藏明文消息的对称密钥依然具有足够的最小熵.

(3) 对于 IBE 机制而言, CCA 安全性是性能更优的安全属性, 因此本文基于熵泄露容忍的 IBE 机制、二源提取器和消息验证码等密码原语提出具有 CCA 安全性的抗挑战后泄露容忍的 IBE 机制.

特别地, 可更新的哈希证明系统 (updatable identity-based hash proof system, U-IB-HPS) 是对 IB-HPS 的性能扩展 [19], 它提供了对密钥的更新功能, 结合本文方法, 可基于 U-IB-HPS 获得抗密钥连续泄露攻击的挑战后泄露容忍 CCA 安全的 IBE 机制.

2 基础知识

文献 [18,20,21] 详细介绍了 MAC, IBE, IB-HPS 和 U-IB-HPS 等基础密码原语的形式化定义和安全属性. 由于篇幅所限且内容相似, 本文不再赘述上述内容.

令 $\mathsf{SD}(A,B)=\frac{1}{2}\sum_{w\in\Omega}|\Pr[A=w]-\Pr[B=w]|$ 是有限域 Ω 上任意两个随机变量 A 和 B 间的统计距离; $H_\infty(X)=-\log(\mathrm{Max}_x\Pr[X=x])$ 是 X 的最小熵, 其中 $\mathrm{Max}_x\Pr[X=x]$ 表示概率 $\Pr[X=x]$ 的最大值; $\tilde{H}_\infty(X|Y)=-\log\left(\mathrm{E}_{y\leftarrow Y}\left[2^{-H_\infty(X|Y=y)}\right]\right)$ 是已知 Y 时变量 X 的平均最小熵, 其中 E 是数学期望 [22]. 若 S 是集合, 用 $s\leftarrow_R S$ 表示从 S 中均匀随机地选取 s. $a\leftarrow\mathcal{F}(b)$ 表示算法 $\mathcal{F}$ 的输入为 b, 对应的输出为 a. $\mathrm{View}_{\mathcal{A}}(\Pi)$ 表示敌手 $\mathcal{A}$ 执行密码原语 Π 时的视图. 此外, 用 κ 表示安全参数.

2.1 相关引理

引理1 已知随机变量 A, B 和 Z, 其中 B 满足 $|B|\leqslant 2^l$ ($|B|$ 表示 B 的取值个数), 则 $\tilde{H}_\infty(A|(B,Z))\geqslant\tilde{H}_\infty(A|Z)-l$.

引理2 令 X 是有限域 Ω 上的任意变量, U 是 Ω 中的均匀随机变量, Y 是任意的变量. 对于 $\varepsilon\in[0,1]$, 若有 $\mathsf{SD}((X,Y),(U,Y))\leqslant\varepsilon$ 成立, 那么 $\tilde{H}_\infty(X|Y)\geqslant-\log(\frac{1}{|\Omega|+\varepsilon})$.

2.2 强随机性提取器

定义1 令随机变量 X 和 Y 满足条件 $X\in\{0,1\}^{l_n}$ 和 $\tilde{H}_\infty(X|Y)\geqslant k$, 对于任意的变量 $S\leftarrow_R\{0,1\}^{l_t}$ 和 $Z\leftarrow_R\{0,1\}^{l_m}$, 若有 $\mathsf{SD}((\mathsf{Ext}(X,S),S,Y),(Z,S,Y))\leqslant\varepsilon$ 成立, 则称 $\mathsf{Ext}:\{0,1\}^{l_n}\times\{0,1\}^{l_t}\to\{0,1\}^{l_m}$ 是平均情况的 (k,ε)- 强随机性提取器, 其中 $S\leftarrow_R\{0,1\}^{l_t}$ 是可公开的随机种子.

2.3 二源提取器

相较于强随机性提取器需输入短且真实随机的种子 (该种子有时很难获得), 二源提取器在不需要随机种子的前提下, 能够从相互独立的两个随机源中提取出均匀随机变量.

定义2 令随机变量 $A \in \{0,1\}^{l_n}$ 和 $B \in \{0,1\}^{l_n}$ 的最小熵为 k, 若有 $\mathsf{SD}\left(\text{2-Ext}\left(A,B\right),U_m\right) \leqslant \varepsilon$ 成立, 则称函数 $\text{2-Ext}: \{0,1\}^{l_n} \times \{0,1\}^{l_n} \to \{0,1\}^{l_m}$ 是最差情况的 (k,ε)- 二源提取器, 其中 U_m 是 $\{0,1\}^{l_m}$ 上的均匀随机值.

定义3 对于任意的随机变量 Z, $A \in \{0,1\}^{l_n}$ 和 $B \in \{0,1\}^{l_n}$ 满足条件 $\tilde{H}_\infty\left(A|Z\right) \geqslant k$ 和 $\tilde{H}_\infty\left(B|Z\right) \geqslant k$, 若有 $\mathsf{SD}\left(\left(\text{2-Ext}\left(A,B\right),Z\right),\left(U_m,Z\right)\right) \leqslant \varepsilon$ 成立, 则称函数 $\text{2-Ext}: \{0,1\}^{l_n} \times \{0,1\}^{l_n} \to \{0,1\}^{l_m}$ 是平均情况的 (k,ε)- 二源提取器.

3 抗挑战后泄露的熵安全性

对于任意的敌手, 当其获得加密密文和相应的泄露信息 (即使泄露是在看到密文之后获得的) 后, 原始明文消息依然具有足够的最小熵, 那么相应的加密机制具备熵泄露容忍性. 本节将给出 IBE 机制熵泄露安全性的属性要求及具体的安全性定义.

3.1 IBE 机制的熵泄露容忍性

为了给出 IBE 机制 $\Pi = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ 熵泄露容忍性的具体定义, 本文将定义下述两个游戏: 真实游戏 Game_{rl} 和模拟游戏 Game_{sm}. 令 k 是消息 M 的最小熵, 为了方便起见, 本文假设 M 是 k 比特长的均匀随机字符串, 即 $M \leftarrow_R \{0,1\}^k$; l_{Pre} 和 l_{Post} 分别表示游戏中各部分的泄露量, 其中 l_{Pre} 表示收到挑战密文前关于密钥的泄露长度, l_{Post} 表示收到挑战密文后关于密钥的泄露长度; 所有的参数均由安全参数 κ 通过相应的函数生成.

真实游戏 Game_{rl}. 给定相应的参数 $(k, l_{\text{Pre}}, l_{\text{Post}})$, 真实游戏的两个参与者分别为挑战者 $\mathcal{C}$ 和敌手 $\mathcal{A}$, 具体的消息交互过程如下所述.

(1) 初始化. $\mathcal{C}$ 运行 $(\text{Params}, \text{msk}) \leftarrow \text{Setup}\left(1^\kappa\right)$, 生成 Params 和 msk, 并将 Params 发送给 $\mathcal{A}$.

(2) 挑战前的询问. 在收到挑战密文之前, $\mathcal{A}$ 能够适应性地执行密钥生成询问和挑战前泄露询问, 其中适应性询问是指 $\mathcal{A}$ 基于前期询问的应答结果提出后续的询问内容.

当 $\mathcal{C}$ 收到 $\mathcal{A}$ 提出的对身份 $\text{id} \in \mathcal{ID}$ (其中 $\mathcal{ID}$ 为相应 IBE 机制的身份空间) 的密钥生成询问时, 运行 $\text{sk}_{\text{id}} \leftarrow \text{KeyGen}\left(\text{id}, \text{msk}\right)$, 并将生成的密钥 sk_{id} 发送给 $\mathcal{A}$.

$\mathcal{A}$ 以泄露函数 $f_i^{\text{Pre}}: \{0,1\}^* \to \{0,1\}^{l_1}$ 作为输入, 向 $\mathcal{C}$ 发出关于身份 $\text{id} \in \mathcal{ID}$ 的挑战前泄露询问. $\mathcal{C}$ 首先运行 $\text{sk}_{\text{id}} \leftarrow \text{KeyGen}\left(\text{id}, \text{msk}\right)$ 生成 id 所对应的密钥 sk_{id}, 然后返回泄露信息 $f_i^{\text{Pre}}\left(\text{sk}_{\text{id}}\right)$ 给 $\mathcal{A}$. 特别地, 在 $\mathcal{A}$ 进行的多项式次挑战前泄露询问中关于相同 sk_{id} 的挑战前泄露总量不超过泄露参数 l_{Pre}.

(3) 挑战. $\mathcal{A}$ 向 $\mathcal{C}$ 输出挑战身份 $\text{id}^* \in \mathcal{ID}$, 并且 id^* 不能在任何密钥生成询问中出现. $\mathcal{C}$ 随机选取 $M^{\text{rl}} \in \{0,1\}^k$, 计算 $C^* = \text{Enc}\left(\text{id}^*, M^{\text{rl}}\right)$, 最后将 C^* 发送给 $\mathcal{A}$, 其中 M^{rl} 表示真实游戏的明文.

(4) 挑战后的询问. 收到挑战密文 C^* 之后, $\mathcal{A}$ 可适应性地执行密钥生成询问和挑战后泄露询问.

$\mathcal{C}$ 收到 $\mathcal{A}$ 对除 id^* 之外的任意身份 $\text{id} \in \mathcal{ID}$ (其中 $\text{id} \neq \text{id}^*$) 的密钥生成询问后, 将采用与挑战前密钥生成询问相同的方式进行应答.

$\mathcal{A}$ 以泄露函数 $f_i^{\text{Post}}:\{0,1\}^*\to\{0,1\}^{l_2}$ 作为输入, 向 $\mathcal{C}$ 发出关于 id 的挑战后泄露询问. $\mathcal{C}$ 运行 $\text{sk}_{\text{id}}\leftarrow\text{KeyGen}(\text{id},\text{msk})$ 生成 id 所对应的 sk_{id} 后, 返回泄露信息 $f_i^{\text{Post}}(\text{sk}_{\text{id}})$ 给 $\mathcal{A}$.

特别地, 在 $\mathcal{A}$ 进行的多项式次挑战后泄露询问中关于相同 sk_{id} 的挑战后泄露总量不超过泄露参数 l_{Post}.

随机变量 $\text{View}_{\mathcal{A}}^{\text{rl}}(\Pi)=(\text{randomness},\text{Params},\text{sk}_{\text{id}},\text{id}^*,f_{\text{Pre}}(\text{sk}_{\text{id}^*}),C^*,f_{\text{Post}}(\text{sk}_{\text{id}^*}))$ 表示在真实游戏中 $\mathcal{A}$ 的视图; M^{rl} 表示真实游戏中挑战者随机选取的明文信息, $(M^{\text{rl}},\text{View}_{\mathcal{A}}^{\text{rl}}(\Pi))$ 表示在真实游戏中消息和敌手视图的联合分布. 特别地, 由于 $\mathcal{A}$ 能够获得除 id^* 之外任意身份 id 的密钥 sk_{id}, 因此在 $\text{View}_{\mathcal{A}}^{\text{rl}}(\Pi)$ 中仅包含了 sk_{id^*} 的泄露信息.

模拟游戏 Game_{sm}. 在该游戏中, 使用模拟者 Simu 代替上述游戏 Game_{rl} 的挑战者, Simu 从相应的明文空间中均匀随机地选取 $M^{\text{sm}}\in\{0,1\}^k$, 并为消息 M^{sm} 模拟与敌手 $\mathcal{A}$ 的游戏交互过程. 给定相应的参数 $(k,l_{\text{Pre}},l_{\text{Post}})$, 模拟游戏中具体的消息交互过程如下所述.

(1) 初始化. Simu 运行 $(\text{Params},\text{msk})\leftarrow\text{Setup}(1^\kappa)$, 产生 Params 和 msk, 并将 Params 发送给 $\mathcal{A}$.

(2) 挑战前的询问. 收到挑战密文之前, $\mathcal{A}$ 可适应性地执行密钥生成询问和挑战前泄露询问. $\mathcal{A}$ 发出对 $\text{id}\in\mathcal{ID}$ 的密钥生成询问. Simu 运行 $\text{sk}_{\text{id}}\leftarrow\text{KeyGen}(\text{id},\text{msk})$, 发送 sk_{id} 给 $\mathcal{A}$.

$\mathcal{A}$ 以泄露函数 $f_i^{\text{Pre}}:\{0,1\}^*\to\{0,1\}^{\lambda_i}$ 作为输入, 向 Simu 发出关于 id 的挑战前泄露询问. Simu 运行 $\text{sk}_{\text{id}}\leftarrow\text{KeyGen}(\text{id},\text{msk})$ 生成对应的密钥 sk_{id}, 并将相应的挑战前泄露信息 $f_i^{\text{Pre}}(\text{sk}_{\text{id}})$ 发送给 $\mathcal{A}$. 类似地, 在 $\mathcal{A}$ 进行的多项式次挑战前泄露询问中关于相同 sk_{id} 的挑战前泄露总量不超过泄露参数 l_{Pre}.

(3) 挑战. $\mathcal{A}$ 向 $\mathcal{C}$ 输出挑战身份 $\text{id}^*\in\mathcal{ID}$, 其中 id^* 不能在任何密钥生成询问中出现. Simu 随机选取 $M^{\text{sm}}\in\{0,1\}^k$, 计算 $C^*=\text{Enc}(\text{id}^*,M^{\text{sm}})$, 并将 C^* 发送给 $\mathcal{A}$, 其中 M^{sm} 表示模拟游戏的明文.

(4) 挑战后的询问. 收到挑战密文 C^* 之后, $\mathcal{A}$ 可适应性地执行密钥生成询问和挑战后泄露询问.

$\mathcal{A}$ 能对除 id^* 之外的任意 $\text{id}\in\mathcal{ID}$ (其中 $\text{id}\neq\text{id}^*$) 进行密钥生成询问, Simu 采用与挑战前密钥生成询问相同的方式进行回应.

$\mathcal{A}$ 以泄露函数 $f_i^{\text{Post}}:\{0,1\}^*\to\{0,1\}^{\lambda_i}$ 作为输入, 向 Simu 发出关于 id 的挑战后泄露询问. Simu 运行 $\text{sk}_{\text{id}}\leftarrow\text{KeyGen}(\text{id},\text{msk})$ 生成对应的密钥 sk_{id}, 并返回相应的挑战后泄露信息 $f_i^{\text{Post}}(\text{sk}_{\text{id}})$ 给 $\mathcal{A}$. 类似地, 在 $\mathcal{A}$ 进行的多项式次挑战后泄露询问中关于相同 sk_{id} 的挑战后泄露总量不超过泄露参数 l_{Post}.

随机变量 $\text{View}_{\mathcal{A}}^{\text{sm}}(\text{Simu})=(\text{randomness},\text{Params},\text{id}^*,\text{sk}_{\text{id}},f_{\text{Pre}}(\text{sk}_{\text{id}^*}),C^*,f_{\text{Post}}(\text{sk}_{\text{id}^*}))$ 表示在模拟游戏中敌手 $\mathcal{A}$ 的视图.

定义4 (IBE 机制的熵泄露容忍性) 对于相应的泄露参数 k, l_{Pre} 和 l_{Post}, 若存在一个模拟者 Simu, 使得任意的敌手 $\mathcal{A}$ 满足下述两个条件, 那么对应的 IBE 机制是熵泄露容忍的.

(1) 敌手 $\mathcal{A}$ 的真实视图 $(M^{\text{rl}},\text{View}_{\mathcal{A}}^{\text{rl}}(\Pi))$ 和模拟视图 $(M^{\text{sm}},\text{View}_{\mathcal{A}}^{\text{sm}}(\text{Simu}))$ 是不可区分的;

(2) 给出 $\text{View}_{\mathcal{A}}^{\text{sm}}(\text{Simu})$ 的前提下, 消息 M^{sm} 的平均最小熵为 $\tilde{H}_\infty(M^{\text{sm}}|\text{View}_{\mathcal{A}}^{\text{sm}}(\text{Simu}))\geqslant k-l_{\text{Post}}-\omega(\log\kappa)$, 其中 $\omega(\log\kappa)$ 表示计算中所产生的额外泄露量. 特别地, 上述关系式能够确保在敌手获得挑战后泄露的前提下 M^{sm} 具有足够的最小熵.

敌手获得密钥的挑战后泄露量至多为 l_{Post}, 其通过挑战后泄露获得消息 M^{sm} 的信息至多为 l_{Post}. 挑战前的泄露 l_{Pre}, 并不影响消息 M^{sm} 的最小熵, M^{sm} 最小熵的改变仅与挑战后的泄露 l_{Post} 有关. 除了挑战之后的泄露 l_{Post} 和额外消耗的 $\omega(\log\kappa)$ 比特外, 消息 M^{sm} 保持了其最初的熵; 由于模拟游戏和真实游戏是不可区分的, 因此在真实游戏中明文消息 M^{rl} 保持了与 M^{sm} 相同的最小熵, 则真实游戏中敌手获得挑战后泄露的前提下明文消息 M^{rl} 依然具有足够的最小熵.

3.2 熵泄露容忍的 IBE 机制

本小节将展示基于 IB-HPS 所构造的 IBE 机制事实上是熵泄露容忍的, 能够抵抗有界的挑战后泄露攻击. 在该构造中, 加密算法采用 IB-HPS 的有效密文封装算法进行采样, 并使用对应的封装密钥来隐藏消息. 为了显示 IBE 机制的熵泄露容忍性, 模拟者采用无效密文封装算法进行采样, 与加密算法进行相同的操作. 由 IB-HPS 的有效封装密文和无效封装密文的不可区分性确保上述真实游戏和模拟游戏是不可区分的. 此外, 由 IB-HPS 的平滑性可知, 无效密文对应的封装密钥具有较高的最小熵, 因此消息被很好地隐藏, 那么消息就具有较高的平均最小熵, 能够抵抗挑战后的泄露攻击. 综上所述, 基于 IB-HPS 构造的 IBE 机制中, 即使敌手能够获得一定数量的挑战后泄露信息, 由于封装密钥具有较高的平均最小熵, 确保密文是足够随机的, 则该机制具有抵抗挑战后泄露攻击的能力.

(1) 具体构造

令 $\Pi = (\mathrm{Setup}, \mathrm{KeyGen}, \mathrm{Encap}, \mathrm{Encap}^*, \mathrm{Decap})$ 是平滑的 IB-HPS, 且封装密钥空间为 $\{0,1\}^{l_\kappa}$; $\mathsf{Ext} : \{0,1\}^{l_t} \times \{0,1\}^{l_\kappa} \to \{0,1\}^{l_\eta}$ 是平均情况的 $(l_\varepsilon, \varepsilon)$- 强随机性提取器, 其输入的平均最小熵为 l_ε, 且输出与 l_η 比特长的任意均匀随机字符串间的统计距离为 ε.

熵泄露容忍的 IBE 机制 $\Pi_1 = (\mathrm{Setup}_1, \mathrm{KeyGen}_1, \mathrm{Enc}_1, \mathrm{Dec}_1)$ 由下述算法组成:

(i) $(\mathrm{Params}, \mathrm{msk}) \leftarrow \mathrm{Setup}_1(1^\kappa)$

输出 $\mathrm{Params} = (\mathrm{mpk}, \mathsf{Ext})$ 和 msk, 其中 $(\mathrm{mpk}, \mathrm{msk}) \leftarrow \mathrm{Setup}(1^\kappa)$.

(ii) $\mathrm{sk}_{\mathrm{id}} \leftarrow \mathrm{KeyGen}_1(\mathrm{msk}, \mathrm{id})$

输出 $\mathrm{sk}_{\mathrm{id}} = d_{\mathrm{id}}$, 其中 $d_{\mathrm{id}} \leftarrow \mathrm{KeyGen}(\mathrm{msk}, \mathrm{id})$.

(iii) $C \leftarrow \mathrm{Enc}_1(\mathrm{id}, M)$, 其中 $M \in \{0,1\}^{l_\eta}$

随机选取 $S \leftarrow \{0,1\}^{l_t}$, 并计算 $(c_1, k) \leftarrow \mathrm{Encap}(\mathrm{id})$ 和 $c_2 \leftarrow \mathsf{Ext}(k, S) \oplus M$. 输出 $C = (c_1, c_2, S)$.

(iv) $M \leftarrow \mathrm{Dec}_1(\mathrm{sk}_{\mathrm{id}}, C)$

计算 $k \leftarrow \mathrm{Decap}(\mathrm{sk}_{\mathrm{id}}, c_1)$ 和 $M \leftarrow \mathsf{Ext}(k, S) \oplus c_2$. 最后, 输出 M 作为相应密文 C 的解密结果.

特别地, 上述通用构造的正确性可由底层 IB-HPS 和强随机性提取器的正确性获得. 此外, IBE 机制的构造中仅使用了底层 IB-HPS 的有效密文封装算法, 无效密文封装算法将在安全性证明中使用.

(2) 安全性证明

定理1 若 $\Pi = (\mathrm{Setup}, \mathrm{KeyGen}, \mathrm{Encap}, \mathrm{Encap}^*, \mathrm{Decap})$ 是具有平滑性的 IB-HPS; $\mathsf{Ext} : \{0,1\}^{l_t} \times \{0,1\}^{l_\kappa} \to \{0,1\}^{l_\eta}$ 是平均情况的 $(l_\varepsilon, \varepsilon)$- 强随机性提取器, 那么上述 IBE 机制的通用构造具有熵泄露容忍性, 其中挑战前和挑战后的泄露参数 l_{Pre} 和 l_{Post} 满足关系 $l_{\mathrm{Pre}} \leqslant \log(\frac{1}{l_k+\varepsilon}) - l_\varepsilon$ 和 $l_{\mathrm{Post}} \leqslant l_\eta - \omega(\log\kappa)$.

证明 由熵泄露容忍性的定义可知, 需将真实游戏转换为模拟游戏分析 IBE 机制的熵泄露容忍性. 真实游戏和模拟游戏的具体构造如下.

真实游戏 $\mathrm{Game}_{\mathrm{rl}}$: 该游戏由挑战者 $\mathcal{C}$ 和敌手 $\mathcal{A}$ 执行, 其中 $\mathcal{C}$ 运行 $(\mathrm{Params}, \mathrm{msk}) \leftarrow \mathrm{Setup}_1(1^\kappa)$ 建立 IBE 机制的系统环境. 挑战前 $\mathcal{A}$ 能够进行密钥生成和挑战前泄露询问, 由于 $\mathcal{C}$ 掌握 msk, 则其能够应答 $\mathcal{A}$ 所提交的相关询问. 收到 $\mathcal{A}$ 递交的挑战身份 id^* 后, $\mathcal{C}$ 通过下述计算生成挑战密文 $C^* = (c_1^*, c_2^*, S^*)$.

随机选取 $M^{\mathrm{rl}} \leftarrow_R \{0,1\}^{l_\kappa}$ 和 $S^* \leftarrow_R \{0,1\}^{l_t}$, 计算 $(c_1^*, k^*) \leftarrow \mathrm{Encap}(\mathrm{id}^*)$ 和 $c_2^* \leftarrow \mathsf{Ext}(k^*, S^*) \oplus M^{\mathrm{rl}}$.

挑战后 $\mathcal{A}$ 能够继续进行密钥生成 (除 id^* 外) 和挑战后泄露询问, 那么 $\mathcal{C}$ 用与挑战前同样的方式应答 $\mathcal{A}$ 所提交的相关询问.

游戏 $\mathrm{Game}_{\mathrm{mi}}$: 该游戏与原始游戏 $\mathrm{Game}_{\mathrm{rl}}$ 相类似, 但该游戏中 $\mathcal{C}$ 使用 id^* 对应的密钥 $\mathrm{sk}_{\mathrm{id}^*}$ 完成挑战密文 $C^* = (c_1^*, c_2^*, S^*)$ 的生成. 收到 $\mathcal{A}$ 递交的 id^* 后, $\mathcal{C}$ 通过下述计算生成相应的 $C^* = (c_1^*, c_2^*, S^*)$.

(i) 计算 $\mathrm{sk}_{\mathrm{id}^*} \leftarrow \mathrm{KeyGen}_1(\mathrm{msk}, \mathrm{id}^*)$.

(ii) 随机选取 $M^{\mathrm{rl}} \leftarrow_R \{0,1\}^{l_\kappa}$ 和 $S^* \leftarrow_R \{0,1\}^{l_t}$, 并计算 $(c_1^*, k^*) \leftarrow \mathrm{Encap}(\mathrm{id}^*)$, $\tilde{k}^* \leftarrow \mathrm{Decap}(\mathrm{sk}_{\mathrm{id}^*}, c_1^*)$ 和 $c_2^* \leftarrow \mathsf{Ext}(\tilde{k}^*, S^*) \oplus M^{\mathrm{rl}}$.

$\mathrm{Game}_{\mathrm{mi}}$ 使用 $\mathrm{sk}_{\mathrm{id}^*}$ 对 c_1^* 进行解封装, 并用相应的结果 $\tilde{k}^*$ 对 M^{rl} 进行隐藏, 由底层 IB-HPS 的正确性可知 $\mathrm{Game}_{\mathrm{rl}}$ 与 $\mathrm{Game}_{\mathrm{mi}}$ 是不可区分的, 因此有 $\mathsf{SD}((M^{\mathrm{rl}}, \mathrm{View}_{\mathcal{A}}^{\mathrm{rl}}(\Pi)), (M^{\mathrm{rl}}, \mathrm{View}_{\mathcal{A}}^{mi}(\Pi))) \leqslant \mathrm{negl}(\kappa)$, 其中 $\mathrm{negl}(\kappa)$ 表示在安全参数 κ 上可忽略的值.

模拟游戏 $\mathrm{Game}_{\mathrm{sm}}$: 该游戏由模拟者 Simu 和敌手 $\mathcal{A}$ 执行, 使用底层 IB-HPS 的无效密文封装算法完成 $C^* = (c_1^*, c_2^*, S^*)$ 的生成, 其中对于挑战前和挑战后的密钥生成询问与泄露询问, Simu 使用与 $\mathrm{Game}_{\mathrm{mi}}$ 相类似的方法进行应答. 对于随机消息 $M^{\mathrm{sm}} \leftarrow_R \{0,1\}^{l_\kappa}$, Simu 基于下述计算生成 $C^* = (c_1^*, c_2^*, S^*)$.

(i) 计算 $\mathrm{sk}_{\mathrm{id}^*} \leftarrow \mathrm{KeyGen}_1(\mathrm{msk}, \mathrm{id}^*)$.

(ii) 随机选取 $M^{\mathrm{sm}} \leftarrow_R \{0,1\}^{l_\kappa}$ 和 $S^* \leftarrow_R \{0,1\}^{l_t}$, 计算 $c_1^* \leftarrow \mathrm{Encap}^*(\mathrm{id}^*)$, $\tilde{k}^* \leftarrow \mathrm{Decap}(\mathrm{sk}_{\mathrm{id}^*}, c_1^*)$ 和 $c_2^* \leftarrow \mathsf{Ext}(\tilde{k}^*, S^*) \oplus M^{\mathrm{sm}}$.

$\mathrm{Game}_{\mathrm{sm}}$ 使用 Encap^* 代替 $\mathrm{Game}_{\mathrm{mi}}$ 中的 Encap 完成 c_1^* 的生成. 由有效密文与无效密文的不可区分性可知 $\mathrm{Game}_{\mathrm{mi}}$ 与 $\mathrm{Game}_{\mathrm{sm}}$ 是不可区分的, 则有 $\mathsf{SD}\left(\left(M^{\mathrm{rl}}, \mathrm{View}_{\mathcal{A}}^{mi}(\Pi)\right), \left(M^{\mathrm{sm}}, \mathrm{View}_{\mathcal{A}}^{\mathrm{sm}}(\mathrm{Simu})\right)\right) \leqslant \mathrm{negl}(\kappa)$.

因此 $\mathsf{SD}((M^{\mathrm{rl}}, \mathrm{View}_{\mathcal{A}}^{\mathrm{rl}}(\Pi)), (M^{\mathrm{sm}}, \mathrm{View}_{\mathcal{A}}^{\mathrm{sm}}(\mathrm{Simu}))) \leqslant \mathrm{negl}(\kappa)$, 则 Simu 生成了与 $\mathcal{C}$ 几乎等价的游戏. 由 IB-HPS 的平滑性可知, 在已知 id^* 和 c_1^* 的前提下, k^* 的原始最小熵是 $\log(\frac{1}{l_k+\varepsilon})$. 为满足强随机性提取器 Ext 的输入要求 (对于任意的随机变量, Ext 输入的平均最小熵为 l_ε), 敌手在已知挑战前泄露信息的前提下, 有关系 $\log(\frac{1}{l_k+\varepsilon}) - l_{\mathrm{Pre}} \geqslant l_\varepsilon \Rightarrow l_{\mathrm{Pre}} \leqslant \log(\frac{1}{l_k+\varepsilon}) - l_\varepsilon$ 成立.

即使 $\mathcal{A}$ 获知 id^*, c_1^*, 挑战前泄露信息 $|\mathrm{Inf}_{\mathrm{leak}}| \leqslant l_{\mathrm{Pre}}$ 和随机种子 S^*, 那么 Ext 的输出是均匀随机的. 由于 Ext 的输出与任意 l_η 比特的字符串是 ε 靠近的, 则明文消息到达挑战阶段时的最小熵为 $l_\eta - \omega(\log\kappa)$, 其中 $\omega(\log\kappa)$ 表示 Simu 在模拟计算过程中的额外开支. 由于挑战之后的泄露界为 l_{Post}, 那么到达挑战阶段时明文消息的最小熵至少为 $l_\eta - l_{\mathrm{Post}} - \omega(\log\kappa)$, 则有 $l_{\mathrm{Post}} \leqslant l_\eta - \omega(\log\kappa)$. 因此, 在 $\mathrm{Game}_{\mathrm{sm}}$ 中有关系 $\tilde{H}_\infty(M^{\mathrm{sm}} | \mathrm{View}_{\mathcal{A}}^{\mathrm{sm}}(\mathrm{Simu})) \geqslant l_\eta - l_{\mathrm{Post}} - \omega(\log\kappa)$ 成立.

综上所述, 对于挑战前及挑战后的泄露参数 $l_{\mathrm{Pre}} \leqslant \log(\frac{1}{l_k+\varepsilon}) - l_\lambda$ 和 $l_{\mathrm{Post}} \leqslant l_\eta - \omega(\log\kappa)$, 上述 IBE 的通用构造具有熵泄露容忍的安全性.

4 状态分离模型下抗挑战后泄露的 IBE 机制

为方便读者理解我们的构造思路, 本文首先在 4.1 小节给出状态分离模型下挑战后泄露容忍的 CPA 安全性的定义; 然后在 4.2 小节构造状态分离模型下抗挑战后泄露攻击的 CPA 安全的 IBE 机制; 最后在 4.3 小节基于上述构造提出 CCA 安全的抗挑战后泄露 IBE 机制.

4.1 挑战后泄露容忍的 CPA 安全性

在状态分离模型中, 将密码学机制中的秘密状态划分为相互独立的几部分, 敌手可分别从其选择的部分获得相应的泄露, 但不能通过一个全局函数作用于所有的秘密状态. 对于 IBE 机制而言, 将用户的密钥 $\mathrm{sk}_{\mathrm{id}}$ 划分为相互独立的多个部分.

定义5 当 $\Pi = (\mathrm{Setup}, \mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ 具备下述性质时, 称其是 2- 状态分离的 IBE 机制.

(1) 任意身份的密钥包含两部分 $\mathrm{sk}_{\mathrm{id}} = (d_1, d_2)$, 其中 d_1 和 d_2 是相互独立的.

(2) 算法 KeyGen 包含两个子程序 KeyGen_1 和 KeyGen_2, 其中对于 $i\in\{1,2\}$, KeyGen_i 的输出为部分密钥 d_i.

(3) 解密算法 Dec 同样包含两个子程序 Dec_1 和 Dec_2, 同时还包含一个拼接子程序 Comb. 对于 $i\in\{1,2\}$, Dec_i 输入密文 C 和相应的部分密钥 d_i, 输出部分解密结果 t_i; 拼接子程序 Comb 输入密文 C 和相应的部分解密结果 t_1 和 t_2, 输出最终的明文 M.

状态分离模型中, 各部分的信息泄露是相互独立的. l_{Pre} 和 l_{Post} 表示游戏中各部分的泄露量, 其中 l_{Pre} 表示收到挑战密文前的泄露量, l_{Post} 表示收到挑战密文后的泄露量.

给定挑战前和挑战后的相应泄露参数 $(l_{\mathrm{Pre}}, l_{\mathrm{Post}})$, 在状态分离模型下, IBE 机制抵抗挑战后泄露的 CPA 安全性游戏包括挑战者 $\mathcal{C}$ 和敌手 $\mathcal{A}$ 两个参与者, 具体的消息交互过程如下所述.

(1) 初始化. $\mathcal{C}$ 运行 $(\mathrm{Params}, \mathrm{msk})\leftarrow\mathrm{Setup}(1^{\kappa})$, 产生 Params 和 msk, 并将 Params 发送给 $\mathcal{A}$.

(2) 阶段 1 (训练). 在该阶段, $\mathcal{A}$ 可适应性地进行密钥生成询问和挑战前泄露询问.

$\mathcal{A}$ 发出对 $\mathrm{id}\in\mathcal{ID}$ 的密钥生成询问. 对于 $i\in\{1,2\}$, $\mathcal{C}$ 运行 $d_i\leftarrow\mathrm{KeyGen}_i(\mathrm{id}, \mathrm{msk})$, 返回相应的 $\mathrm{sk}_{\mathrm{id}}=(d_1,d_2)$ 发送给 $\mathcal{A}$.

$\mathcal{A}$ 以泄露函数 $f_{1,i}^{\mathrm{Pre}}:\{0,1\}^*\rightarrow\{0,1\}^{l_1}$ 和 $f_{2,i}^{\mathrm{Pre}}:\{0,1\}^*\rightarrow\{0,1\}^{l_2}$ 作为输入, 向 $\mathcal{C}$ 发出关于 id 的挑战前泄露询问. $\mathcal{C}$ 运行 $\mathrm{sk}_{\mathrm{id}}\leftarrow\mathrm{KeyGen}(\mathrm{id}, \mathrm{msk})$ 生成相应的 $\mathrm{sk}_{\mathrm{id}}=(d_1,d_2)$, 并把相应的泄露信息 $f_{1,i}^{\mathrm{Pre}}(d_1)$ 和 $f_{2,i}^{\mathrm{Pre}}(d_2)$ 发送给 $\mathcal{A}$. 特别地, 同一密钥 $\mathrm{sk}_{\mathrm{id}}=(d_1,d_2)$ 在挑战前所有泄露询问的长度和至多是 l_{Pre}; 否则挑战者将忽略相应的泄露询问.

(3) 挑战. $\mathcal{A}$ 输出两个等长的明文消息 $M_0, M_1\in\mathcal{M}$ 和一个挑战身份 $\mathrm{id}^*\in\mathcal{ID}$, 其中 id^* 不能在阶段 1 的任何密钥生成询问中出现. $\mathcal{C}$ 计算 $C_{\beta}^*=\mathrm{Enc}(\mathrm{id}^*, M_{\beta})$, 其中 $\beta\leftarrow_R\{0,1\}$, 并将 C_{β}^* 发送给 $\mathcal{A}$.

(4) 阶段 2 (训练). 在该阶段, $\mathcal{A}$ 可适应性地进行密钥生成询问和挑战后泄露询问.

$\mathcal{A}$ 能对除 id^* 之外的任意 $\mathrm{id}\in\mathcal{ID}$ (其中 $\mathrm{id}\neq\mathrm{id}^*$) 进行密钥生成询问, $\mathcal{C}$ 使用与阶段 1 中相同的方式进行应答.

$\mathcal{A}$ 以泄露函数 $f_{1,i}^{\mathrm{Post}}:\{0,1\}^*\rightarrow\{0,1\}^{l_1}$ 和 $f_{2,i}^{\mathrm{Post}}:\{0,1\}^*\rightarrow\{0,1\}^{l_2}$ 作为输入, 向 $\mathcal{C}$ 发出关于 id 的挑战后泄露询问. 则 $\mathcal{C}$ 运行 $\mathrm{sk}_{\mathrm{id}}\leftarrow\mathrm{KeyGen}(\mathrm{id}, \mathrm{msk})$ 生成相应的 $\mathrm{sk}_{\mathrm{id}}=(d_1,d_2)$, 并把对应的泄露信息 $f_{1,i}^{\mathrm{Post}}(d_1)$ 和 $f_{2,i}^{\mathrm{Post}}(d_2)$ 发送给 $\mathcal{A}$. 特别地, 同一密钥 $\mathrm{sk}_{\mathrm{id}}=(d_1,d_2)$ 在挑战后所有泄露询问的长度和至多是 l_{Post}; 否则挑战者将忽略相应的泄露询问.

(5) 猜测.

$\mathcal{A}$ 输出对 $\mathcal{C}$ 选取随机数 β 的猜测 $\beta'\in\{0,1\}$, 如果 $\beta'=\beta$, 则 $\mathcal{A}$ 攻击成功, 即 $\mathcal{A}$ 在该游戏中获胜, 且获胜的优势可定义为 $\mathrm{Adv}_{\mathrm{IBE,2\text{-}Post}}^{\mathrm{CPA}}(\kappa, l_{\mathrm{Pre}}, l_{\mathrm{Post}})=\left|\Pr[\beta'=\beta]-\frac{1}{2}\right|$.

定义6 (状态分离模型下 IBE 机制挑战后泄露容忍的 CPA 安全性) 对于任意的概率多项式时间敌手 $\mathcal{A}$, 若有 $\mathrm{Adv}_{\mathrm{IBE,2\text{-}Post}}^{\mathrm{CPA}}(\kappa, l_{\mathrm{Pre}}, l_{\mathrm{Post}})\leqslant\mathrm{negl}(\kappa)$, 那么在状态分离模型中, 对于挑战前和挑战后的泄露参数 l_{Pre} 和 l_{Post}, 相应的 IBE 机制具备挑战后泄露容忍的 CPA 安全性.

与传统语义安全性相比, 该模型中密钥有两个状态. 挑战前, 敌手可适应性地进行任意多次的泄露询问, 但同一密钥的所有挑战前泄露询问的泄露信息总和未超过泄露界 l_{Pre}; 挑战后, 敌手同样适应性地进行任意多次的泄露询问, 条件同样是同一密钥的所有挑战后泄露询问的泄露信息总和未超过泄露界 l_{Post}. 特别地, 对于挑战后泄露容忍的 CCA 安全性而言, 在上述游戏的基础上增加了解密询问, 敌手可就任意的身份密文对向挑战者提出解密询问, 但是挑战后不允许对挑战身份和挑战密文对进行解密询问.

4.2 状态分离模型下 CPA 安全的 IBE 机制

在状态分离模型下 IBE 机制的构造中, 密钥 $\mathrm{sk_{id}}=(d_1,d_2)$ 保持两个独立状态, 其中对于 $i=1,2$, d_i 是密钥生成子程序 KeyGen_2^i 为部分解密子程序 Dec_2^i 生成的部分密钥; 由于状态分离模型的限制, 敌手无法同时获得 d_1 和 d_2 的联合泄露, 并且在存在泄露的情况下, 两部分是相互独立的; 由于 d_1 和 d_2 具有足够的最小熵, 所以能够基于 d_1 和 d_2 使用二源提取器获得接近于均匀随机的字符串完成对消息的隐藏.

(1) 具体构造

令 $\Pi=(\mathrm{Setup},\mathrm{KeyGen},\mathrm{Enc},\mathrm{Dec})$ 是消息空间为 $\{0,1\}^{l_t}$ 和身份空间为 $\mathcal{ID}_1$ 的有熵泄露容忍性的 IBE 机制, 且相应的泄露参数为 l_{Pre} 和 l_{Post}; $H_1:\mathcal{ID}_2\to\mathcal{ID}_1$ 和 $H_2:\mathcal{ID}_2\to\mathcal{ID}_1$ 是两个安全的哈希函数; $\mathsf{2\text{-}Ext}:\{0,1\}^{l_t}\times\{0,1\}^{l_t}\to\{0,1\}^{l_u}$ 是平均情况的 (l_v,ε)- 二源提取器, 其中 $\varepsilon=2^{-l_u-\omega(\log\kappa)}$. 特别地, $\mathsf{2\text{-}Ext}$ 的输入是两个 l_t 比特长的独立字符串, 且其对应输出的平均最小熵至少为 l_v; 此外, $\mathsf{2\text{-}Ext}$ 的输出与长度为 l_u 的随机字符串间的统计距离至多是 ε.

状态分离模型下 CPA 安全的 IBE 的通用构造 $\Pi_2=(\mathrm{Setup}_2,\mathrm{KeyGen}_2,\mathrm{Enc}_2,\mathrm{Dec}_2)$ 由下述算法组成:

(i) $(\mathrm{Params},\mathrm{msk})\leftarrow\mathrm{Setup}_2(1^\kappa)$

输出 $\mathrm{Params}=\left(\mathrm{Params}',H_1,H_2,\mathsf{2\text{-}Ext}\right)$ 和 $\mathrm{msk}=\mathrm{msk}'$, 其中 $\left(\mathrm{Params}',\mathrm{msk}'\right)\leftarrow\mathrm{Setup}(1^\kappa)$.

(ii) $\mathrm{sk_{id}}\leftarrow\mathrm{KeyGen}_2(\mathrm{msk},\mathrm{id})$

对于 $i=1,2$, 运行密钥生成子程序 KeyGen_2^i: 计算 $\mathrm{id}_i=H_i(\mathrm{id})$ 和 $d_i\leftarrow\mathrm{KeyGen}(\mathrm{msk},\mathrm{id}_i)$. 最后输出 $\mathrm{sk_{id}}=(d_1,d_2)$.

(iii) $C\leftarrow\mathrm{Enc}_2(\mathrm{id},M)$, 其中 $M\in\{0,1\}^{l_u}$

随机选取 $x_1,x_2\leftarrow\{0,1\}^t$, 对于 $i=1,2$, 计算 $\mathrm{id}_i=H_i(\mathrm{id})$ 和 $c_i\leftarrow\mathrm{Enc}(\mathrm{id}_i,x_i)$. 然后计算 $c_3=\mathsf{2\text{-}Ext}(x_1,x_2)\oplus M$. 最后输出 $C=(c_1,c_2,c_3)$.

(iv) $M\leftarrow\mathrm{Dec}_2(\mathrm{sk_{id}},C)$, 其中 $\mathrm{sk_{id}}=(d_1,d_2)$ 和 $C=(c_1,c_2,c_3)$

部分解密子程序 Dec_2^1 使用部分密钥 d_1 解密密文元素 c_1, 输出相应的明文 $x_1\leftarrow\mathrm{Dec}(d_1,c_1)$. 部分解密子程序 Dec_2^2 使用部分密钥 d_2 解密密文元素 c_2, 输出相应的明文 $x_2\leftarrow\mathrm{Dec}(d_2,c_2)$. 拼接子程序 Comb 输入部分解密结果 x_1, x_2 和密文元素 c_3, 输出相应的消息 $M=\mathsf{2\text{-}Ext}(x_1,x_2)\oplus c_3$.

不失一般性, 上述解密算法可统一表示为: 对于 $i=1,2$, 计算 $x_i\leftarrow\mathrm{Dec}(d_i,c_i)$; 然后计算 $M=\mathsf{2\text{-}Ext}(x_1,x_2)\oplus c_3$, 最后输出 M 作为相应密文 C 的解密结果.

(2) 安全性证明

定理2 若 $\Pi=(\mathrm{Setup},\mathrm{KeyGen},\mathrm{Enc},\mathrm{Dec})$ 是挑战前和挑战后泄露参数分别为 l_{Pre} 和 l_{Post} 的熵泄露容忍的 IBE 机制; $\mathsf{2\text{-}Ext}$ 是平均情况的 (l_v,ε)- 二源提取器, 那么上述构造是状态分离模型下抗挑战后泄露的 CPA 安全的 IBE 机制, 且泄露参数满足 $l'_{\mathrm{Pre}}\leqslant l_{\mathrm{Pre}}$ 和 $l'_{\mathrm{Post}}\leqslant\min(l_{\mathrm{Post}}-l_u,l_t-l_v-\omega(\log\kappa))$.

证明 在状态分离模型下, 将通过游戏论证的方式对 IBE 机制挑战后泄露容忍的 CPA 安全性进行证明. 令事件 $\mathcal{E}_i$ 表示敌手 $\mathcal{A}$ 在游戏 Game_i 中获胜, 即有 $\Pr[\mathcal{E}_i]=\Pr[\mathcal{A}\text{ wins in Game}_i]$.

证明中, 与挑战密文相关的变量均标记为 "*". 此外, $\mathcal{C}$ 是挑战者, Simu 是熵泄露容忍 IBE 机制 Π 的模拟器, $\mathcal{A}$ 是语义安全性敌手, $\mathcal{A}^{\mathrm{ent}}$ 是攻击机制 Π 的泄露敌手, 且敌手 $\mathcal{A}^{\mathrm{ent}}$ 是敌手 $\mathcal{A}$ 的挑战者, 其中 $\mathcal{A}^{\mathrm{ent}}$ 与 $\mathcal{A}$ 间运行状态分离模型下挑战后泄露容忍的 CPA 安全性游戏, $\mathcal{A}^{\mathrm{ent}}$ 与 $\mathcal{C}$ 间运行熵泄露容忍性的真实游戏, $\mathcal{A}^{\mathrm{ent}}$ 与 Simu 间运行熵泄露容忍性的模拟游戏.

游戏 Game_1: 该游戏是状态分离模型下挑战后泄露容忍的原始 CPA 安全性游戏. 在该游戏中, 熵

敌手 $\mathcal{A}^{\text{ent}}$ 借助底层熵泄露容忍的 IBE 机制 Π 的挑战者 $\mathcal{C}$, 使用真实游戏中的方法生成挑战密文的元素 c_1^* 和 c_2^*, 并且使用真实的方法回答关于部分密钥 d_1 和 d_2 的泄露询问.

初始化阶段, $\mathcal{C}$ 运行 $(\text{Params}', \text{msk}') \leftarrow \text{Setup}(1^\kappa)$, 并发送 Params' 给 $\mathcal{A}^{\text{ent}}$; 然后 $\mathcal{A}^{\text{ent}}$ 发送相应的参数 $(\text{Params}', H_1, H_2, \text{2-Ext})$ 给 $\mathcal{A}$, 其中 2-Ext 是强随机性二源提取器, H_1 和 H_2 是两个安全的哈希函数.

当 $\mathcal{A}^{\text{ent}}$ 收到 $\mathcal{A}$ 关于身份 id 的密钥生成询问时, 首先计算 $\text{id}_1 = H_1(\text{id})$ 和 $\text{id}_2 = H_2(\text{id})$; 然后向 $\mathcal{C}$ 分别提出关于 id_1 和 id_2 的密钥生成询问, 并获得应答 d_1 和 d_2; 最后 $\mathcal{A}^{\text{ent}}$ 返回 $\text{sk}_{\text{id}} = (d_1, d_2)$ 给 $\mathcal{A}$.

当 $\mathcal{A}^{\text{ent}}$ 收到 $\mathcal{A}$ 关于 id 的挑战前泄露询问 $\left(f_{1,i}^{\text{Pre}}, f_{2,i}^{\text{Pre}}\right)$ 时, 首先计算 $\text{id}_1 = H_1(\text{id})$ 和 $\text{id}_2 = H_2(\text{id})$; 然后向 $\mathcal{C}$ 提出关于 $\left(\text{id}_1, f_{1,i}^{\text{Pre}}\right)$ 和 $\left(\text{id}_2, f_{2,i}^{\text{Pre}}\right)$ 的挑战前泄露询问, 并获得 $\mathcal{C}$ 返回的应答 $f_{1,i}^{\text{Pre}}(d_1)$ 和 $f_{2,i}^{\text{Pre}}(d_2)$, 其中 $d_1 = \text{KeyGen}\left(\text{msk}', \text{id}_1\right)$ 和 $d_2 = \text{KeyGen}\left(\text{msk}', \text{id}_2\right)$; 最后 $\mathcal{A}^{\text{ent}}$ 返回 $\left(f_{1,i}^{\text{Pre}}(d_1), f_{2,i}^{\text{Pre}}(d_2)\right)$ 给 $\mathcal{A}$.

挑战阶段, 当收到 $\mathcal{A}$ 提交的关于明文消息 $M_0, M_1 \in \{0,1\}^{l_u}$ (其中 $|M_0| = |M_1|$) 和挑战身份 id^* 的挑战询问时, $\mathcal{A}^{\text{ent}}$ 随机选取 $x_1, x_2 \in \{0,1\}^{l_t}$, 并向 $\mathcal{C}$ 提交关于 x_1 和 x_2 的加密询问, 获得 $\mathcal{C}$ 返回的相应应答 c_1^* 和 c_2^*; 然后自行计算 $r = \text{2-Ext}(x_1, x_2)$ 和 $c_3^* = r \oplus M_\beta$, 其中 $\beta \leftarrow \{0,1\}$; 最后发送 $C_\beta^* = (c_1^*, c_2^*, c_3^*)$ 给 $\mathcal{A}$.

$\mathcal{A}^{\text{ent}}$ 对 $\mathcal{A}$ 所提交的挑战后泄露询问的应答方式与挑战前泄露询问的应答方式一致, 为回答关于 id 的挑战后泄露询问 $\left(f_{1,i}^{\text{Post}}, f_{2,i}^{\text{Post}}\right)$, $\mathcal{A}^{\text{ent}}$ 计算 $\text{id}_1 = H_1(\text{id})$ 和 $\text{id}_2 = H_2(\text{id})$ 后分别将 $\left(\text{id}_1, f_{1,i}^{\text{Post}}\right)$ 和 $\left(\text{id}_2, f_{2,i}^{\text{Post}}\right)$ 发送给 $\mathcal{C}$, 获得相关应答 $f_{1,i}^{\text{Post}}(d_1)$ 和 $f_{2,i}^{\text{Post}}(d_2)$, 其中 $d_1 = \text{KeyGen}\left(\text{msk}', \text{id}_1\right)$ 和 $d_2 = \text{KeyGen}\left(\text{msk}', \text{id}_2\right)$; 最后 $\mathcal{A}^{\text{ent}}$ 将相应的泄露 $\left(f_{1,i}^{\text{Post}}(d_1), f_{2,i}^{\text{Post}}(d_2)\right)$ 返回给 $\mathcal{A}$. 在挑战后, $\mathcal{A}$ 可对除 id^* 之外的任意身份 $\text{id} \neq \text{id}^*$ 进行密钥生成询问, $\mathcal{A}^{\text{ent}}$ 使用与挑战前密钥生成询问相同的应答方式进行回答.

猜测阶段, $\mathcal{A}$ 输出对 $\mathcal{A}^{\text{ent}}$ 选取随机数 β 的猜测 $\beta' \in \{0,1\}$, 如果 $\beta' = \beta$, 则 $\mathcal{A}$ 攻击成功.

游戏 Game_2: 该游戏与 Game_1 相类似, 但挑战者使用真实游戏中的方法生成 $C_\beta^* = (c_1^*, c_2^*, c_3^*)$ 的元素 c_2 和回答关于部分密钥 d_2 的泄露询问; 使用模拟器 Simu 生成 $C_\beta^* = (c_1^*, c_2^*, c_3^*)$ 的元素 c_1 和回答关于部分密钥 d_1 的泄露询问.

初始化阶段, $\mathcal{C}$ 运行 $\left(\text{Params}', \text{msk}'\right) \leftarrow \text{Setup}(1^\kappa)$, 并发送 Params' 给敌手 $\mathcal{A}^{\text{ent}}$; 然后敌手 $\mathcal{A}^{\text{ent}}$ 发送 $\left(\text{Params}', H_1, H_2, \text{2-Ext}\right)$ 给 $\mathcal{A}$, 其中 2-Ext 是强随机性二源提取器, H_1 和 H_2 是两个安全的哈希函数.

当 $\mathcal{A}^{\text{ent}}$ 收到 $\mathcal{A}$ 关于 id 的密钥生成询问时, 首先计算 $\text{id}_1 = H_1(\text{id})$ 和 $\text{id}_2 = H_2(\text{id})$; 然后向 $\mathcal{C}$ 提出关于 id_2 的密钥生成询问, 并获得相应的应答 d_2; 向 Simu 提出关于 id_1 的密钥生成询问, 并获得相应的应答 d_1; 最后 $\mathcal{A}^{\text{ent}}$ 返回 $\text{sk}_{\text{id}} = (d_1, d_2)$ 给 $\mathcal{A}$. 密钥生成询问的应答过程如图 1(a) 所示.

当 $\mathcal{A}^{\text{ent}}$ 收到 $\mathcal{A}$ 关于 id 的挑战前泄露询问 $\left(f_{1,i}^{\text{Pre}}, f_{2,i}^{\text{Pre}}\right)$ 时, 首先计算 $\text{id}_1 = H_1(\text{id})$ 和 $\text{id}_2 = H_2(\text{id})$; 然后向 $\mathcal{C}$ 提出关于 $\left(\text{id}_2, f_{2,i}^{\text{Pre}}\right)$ 的挑战前泄露询问, 并获得应答 $f_{2,i}^{\text{Pre}}(d_2)$; 向 Simu 提出关于 $\left(\text{id}_1, f_{1,i}^{\text{Pre}}\right)$ 的挑战前泄露询问, 并获得应答 $f_{1,i}^{\text{Pre}}(d_1)$; 最后 $\mathcal{A}^{\text{ent}}$ 返回 $\left(f_{1,i}^{\text{Pre}}(d_1), f_{2,i}^{\text{Pre}}(d_2)\right)$ 给 $\mathcal{A}$. 挑战前泄露询问的应答过程如图 1(b) 所示.

挑战阶段, 当收到敌手 $\mathcal{A}$ 提交的关于 $M_0, M_1 \in \{0,1\}^{l_u}$ (其中 $|M_0| = |M_1|$) 和 id^* 的挑战询问时, $\mathcal{A}^{\text{ent}}$ 随机选取 $x_1, x_2 \leftarrow \{0,1\}^t$, 并向 $\mathcal{C}$ 提交关于 x_2 的加密询问, 获得应答 c_2^*; 向 Simu 提交关于 x_1 的加密询问, 获得应答 c_1^*; 然后自行计算 $r = \text{2-Ext}(x_1, x_2)$ 和 $c_3^* = r \oplus M_\beta$, 其中 $\beta \leftarrow \{0,1\}$; 最后发送 $C_\beta^* = (c_1^*, c_2^*, c_3^*)$ 给 $\mathcal{A}$.

$\mathcal{A}^{\text{ent}}$ 对 $\mathcal{A}$ 所提交的挑战后泄露询问的应答方式与挑战前泄露询问的应答方式一致, 为回答关

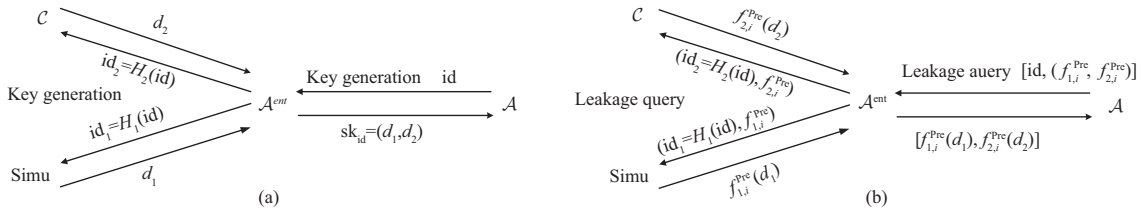

图 1 密钥生成询问和挑战前泄露询问的应答过程. (a) 密钥生成; (b) 泄露信息生成

Figure 1 Responses to the private key generation query and the leakage query before challenges. (a) Private key generation; (b) leakage information generation

于 id 的挑战后泄露询问 $\left(f_{1,i}^{\mathrm{Post}}, f_{2,i}^{\mathrm{Post}}\right)$, $\mathcal{A}^{\mathrm{ent}}$ 计算 $\mathrm{id}_1 = H_1(\mathrm{id})$ 和 $\mathrm{id}_2 = H_2(\mathrm{id})$, 然后将 $\left(\mathrm{id}_1, f_{1,i}^{\mathrm{Post}}\right)$ 发给 Simu, 将 $\left(\mathrm{id}_2, f_{2,i}^{\mathrm{Post}}\right)$ 发给 $\mathcal{C}$, 并将获得的相关应答 $f_{1,i}^{\mathrm{Post}}(d_1)$ 和 $f_{2,i}^{\mathrm{Post}}(d_2)$ 发送给 $\mathcal{A}$, 其中 $d_1 = \mathrm{KeyGen}\left(\mathrm{msk}', \mathrm{id}_1\right)$ 和 $d_2 = \mathrm{KeyGen}\left(\mathrm{msk}', \mathrm{id}_2\right)$. 挑战后 $\mathcal{A}$ 可对除 id^* 之外的任意身份 $\mathrm{id} \neq \mathrm{id}^*$ 进行密钥生成询问, $\mathcal{A}^{\mathrm{ent}}$ 以挑战前密钥生成询问的应答方式进行回答.

引理3 Game_2 中敌手获胜的优势与 Game_1 中的优势是不可区分的, 则 $|\Pr[\mathcal{E}_2] - \Pr[\mathcal{E}_1]| \leqslant \mathrm{negl}(\kappa)$.

证明 若存在敌手能以不可忽略的优势区分 Game_2 和 Game_1, 那么存在一个区分者能以明显的优势攻破底层熵泄露容忍 IBE 机制的模拟不可区分性, 因此 Game_2 和 Game_1 是不可区分的. 也就是说, 若存在一个敌手 $\mathcal{A}$ 能以不可忽略的优势攻破 IBE 机制 Π_2 挑战后泄露容忍的 CPA 安全性, 那么就能构造一个熵泄露敌手 $\mathcal{A}^{\mathrm{ent}}$ 能以显而易见的优势攻破底层 IBE 机制 Π 的熵泄露容忍性.

游戏 Game_3: 该游戏与 Game_2 相类似, 但在该游戏中, $\mathcal{A}^{\mathrm{ent}}$ 使用模拟器 Simu 生成挑战密文的元素 c_1^* 和 c_2^*, 同时使用模拟器 Simu 回答关于部分密钥 d_1 和 d_2 的泄露询问.

初始化阶段, Simu 运行 $\left(\mathrm{Params}', \mathrm{msk}'\right) \leftarrow \mathrm{Setup}\left(1^\kappa\right)$, 并发送 Params' 给 $\mathcal{A}^{\mathrm{ent}}$; 然后, $\mathcal{A}^{\mathrm{ent}}$ 发送 $\left(\mathrm{Params}', H_1, H_2, \text{2-Ext}\right)$ 给 $\mathcal{A}$.

当 $\mathcal{A}^{\mathrm{ent}}$ 收到 $\mathcal{A}$ 关于 id 的密钥生成询问时, 首先计算 $\mathrm{id}_1 = H_1(\mathrm{id})$ 和 $\mathrm{id}_2 = H_2(\mathrm{id})$; 然后向 Simu 提出关于 id_1 和 id_2 的密钥生成询问, 并获得相应的应答 d_1 和 d_2; 最后 $\mathcal{A}^{\mathrm{ent}}$ 返回 $\mathrm{sk}_{\mathrm{id}} = (d_1, d_2)$ 给 $\mathcal{A}$.

当 $\mathcal{A}^{\mathrm{ent}}$ 收到 $\mathcal{A}$ 关于身份 id 的挑战前泄露询问 $\left(f_{1,i}^{\mathrm{Pre}}, f_{2,i}^{\mathrm{Pre}}\right)$ 时, 首先计算 $\mathrm{id}_1 = H_1(\mathrm{id})$ 和 $\mathrm{id}_2 = H_2(\mathrm{id})$; 然后向 Simu 提出关于 $\left(\mathrm{id}_1, f_{1,i}^{\mathrm{Pre}}\right)$ 和 $\left(\mathrm{id}_2, f_{2,i}^{\mathrm{Pre}}\right)$ 的挑战前泄露询问, 并将从 Simu 获得的相关应答 $f_{1,i}^{\mathrm{Pre}}(d_1)$ 和 $f_{2,i}^{\mathrm{Pre}}(d_2)$ 发送给 $\mathcal{A}$, 其中 $d_1 = \mathrm{KeyGen}\left(\mathrm{msk}', \mathrm{id}_1\right)$ 和 $d_2 = \mathrm{KeyGen}\left(\mathrm{msk}', \mathrm{id}_2\right)$.

挑战阶段, 当收到敌手 $\mathcal{A}$ 提交的关于 $M_0, M_1 \in \{0,1\}^{l_u}$ (其中 $|M_0| = |M_1|$) 和 id^* 的挑战询问时, $\mathcal{A}^{\mathrm{ent}}$ 随机选取 $x_1, x_2 \leftarrow \{0,1\}^t$, 并向 Simu 提交关于 x_1 和 x_2 的加密询问, 获得 $\mathcal{C}$ 返回的相应应答 c_1^* 和 c_2^*; 然后自行计算 $r = \text{2-Ext}(x_1, x_2)$ 和 $c_3^* = r \oplus M_\beta$, 其中 $\beta \leftarrow \{0,1\}$; 最后发送 $C_\beta^* = (c_1^*, c_2^*, c_3^*)$ 给 $\mathcal{A}$.

$\mathcal{A}^{\mathrm{ent}}$ 对 $\mathcal{A}$ 所提交的挑战后泄露询问的应答方式与挑战前泄露询问的应答方式一致, 为回答关于 id 的挑战后泄露询问 $\left(f_{1,i}^{\mathrm{Post}}, f_{2,i}^{\mathrm{Post}}\right)$, $\mathcal{A}^{\mathrm{ent}}$ 计算 $\mathrm{id}_1 = H_1(\mathrm{id})$ 和 $\mathrm{id}_2 = H_2(\mathrm{id})$ 后将 $\left(\mathrm{id}_1, f_{1,i}^{\mathrm{Pre}}\right)$ 和 $\left(\mathrm{id}_2, f_{2,i}^{\mathrm{Pre}}\right)$ 发送给 Simu, 并将获得的相应应答 $f_{1,i}^{\mathrm{Post}}(d_1)$ 和 $f_{2,i}^{\mathrm{Post}}(d_2)$ 发送给 $\mathcal{A}$, 其中 $d_1 = \mathrm{KeyGen}\left(\mathrm{msk}', \mathrm{id}_1\right)$ 和 $d_2 = \mathrm{KeyGen}\left(\mathrm{msk}', \mathrm{id}_2\right)$. 挑战后 $\mathcal{A}$ 可对除 id^* 之外的任意身份 $\mathrm{id} \neq \mathrm{id}^*$ 进行密钥生成询问, $\mathcal{A}^{\mathrm{ent}}$ 使用与挑战前密钥生成询问的相同应答方式进行回答.

引理4 Game_3 中敌手获胜的优势与 Game_2 中的优势是不可区分的, 则 $|\Pr[\mathcal{E}_3] - \Pr[\mathcal{E}_2]| \leqslant \mathrm{negl}(\kappa)$.

与引理 3 相类似的, 由底层熵泄露容忍安全的 IBE 机制的模拟安全性可知, Game_3 和 Game_2 是不可区分的. 本文不再赘述引理 4 的证明过程.

游戏 Game_4: 该游戏与 Game_3 相类似, 但 $\mathcal{A}^{\text{ent}}$ 在挑战阶段仅发送挑战密文元素 c_1^* 和 c_2^* 给 $\mathcal{A}$, 挑战后泄露询问结束后再发送挑战密文元素 c_3^* 给 $\mathcal{A}$.

与 Game_4 相比, 在 Game_3 中, $\mathcal{A}$ 在挑战后泄露询问中具有适应性选择泄露函数的能力. 由下述推论可知, 在参数 l_u 的作用下, $\mathcal{A}$ 在 Game_3 中的优势是有限的.

推论1 若存在敌手 $\mathcal{A}$ 能以优势 ρ 在 Game_3 中获胜, 那么存在敌手 $\mathcal{A}'$ 能以优势 $\frac{1}{2^{l_u}}\rho$ 在 Game_4 中获胜, 即对于 $\rho \geqslant 0$, 若有 $\Pr[\mathcal{E}_3] \geqslant \frac{1}{2}+\rho$, 那么有 $\Pr[\mathcal{E}_4] \geqslant \frac{1}{2}+\frac{1}{2^{l_u}}\rho$ 成立.

证明 在 Game_4 的挑战阶段, $\mathcal{A}'$ 随机选取密文元素 $\tilde{c}_3^* \in \{0,1\}^{l_u}$, 连同收到的元素 c_1^* 和 c_2^* 组成完整的挑战密文 $C_\beta^* = (c_1^*, c_2^*, \tilde{c}_3^*)$ 发送给 $\mathcal{A}$. 当 $\mathcal{A}'$ 收到真实密文元素 c_3^* 后, 若其猜测错误 $(c_3^* \neq \tilde{c}_3^*)$, 则 $\mathcal{A}'$ 终止; 否则, $\mathcal{A}$ 获得了 Game_3 中的交互消息. 由上述基于 $\mathcal{A}$ 构造 $\mathcal{A}'$ 的通用形式可知, 若 $\mathcal{A}'$ 猜测密文元素正确, 那么当敌手 $\mathcal{A}$ 在 Game_3 中获胜, 那么 $\mathcal{A}'$ 将在 Game_4 中获胜. 综上所述, 若存在 $\mathcal{A}$ 能以优势 ρ 在 Game_3 中获胜, 由于 $\mathcal{A}'$ 以概率 $\frac{1}{2^{l_u}}$ 猜测正确, 因此 $\mathcal{A}'$ 的优势是 $\frac{1}{2^{l_u}}\rho$.

推论2 $\left|\Pr[\mathcal{E}_4] - \frac{1}{2}\right| = 2\varepsilon$.

证明 下面本文使用最小熵的性质计算 $\mathcal{A}'$ 在 Game_4 中获胜的优势. 在 Game_4 中, 用 T 表示 $\mathcal{A}'$ 在挑战后泄露询问结束时 (收到密文元素 c_3^* 之前) 收到的消息集合. 令 $T = (T_1, T_2)$, 其中 T_1 表示包含密文元素 c_1^* (c_1^* 是 x_1 的加密密文) 和关于部分密钥 d_1 的泄露信息的消息集合, T_2 表示包含密文元素 c_2^* (c_2^* 是 x_2 的加密密文) 和关于部分密钥 d_2 的泄露信息的消息集合. 由底层 IBE 机制的熵泄露容忍性可知 $l'_{\text{Post}} \leqslant l_t - l_v - \omega(\log\kappa)$ (其中 l_t 是消息的原始最小熵, l_v 是隐藏消息的对称密钥的最小熵), 因此有 $\tilde{H}_\infty(x_1|T_1) \geqslant l_t - l'_{\text{Post}} - \omega(\log\kappa) \geqslant l_v$. 由状态分离模型的定义可知, x_1 和 T_2 是相互独立的, 那么有 $\tilde{H}_\infty(x_1|T) = \tilde{H}_\infty(x_1|T_1) \geqslant l_v$ 成立. 类似地, 可以得到 $\tilde{H}_\infty(x_2|T) \geqslant l_v$.

由于 x_1 和 x_2 在挑战后泄露询问结束时的平均最小熵为 l_v, 那么 (l_v, ε)- 二源提取器 $\mathsf{2\text{-}Ext}(x_1,x_2)$ 的输出与长度为 l_u 的随机字符串间的统计距离至多为 ε, 因此两个分布 $\mathsf{2\text{-}Ext}(x_1,x_2) \oplus M_0$ 和 $\mathsf{2\text{-}Ext}(x_1, x_2) \oplus M_1$ 间的统计距离至多为 2ε. 也就是说, 对于任意的 $U_m \in \{0,1\}^{l_u}$, 有 $\mathsf{SD}(\mathsf{2\text{-}Ext}(x_1,x_2) \oplus M_1, U_m) \leqslant \varepsilon$ 和 $\mathsf{SD}(\mathsf{2\text{-}Ext}(x_1,x_2) \oplus M_2, U_m) \leqslant \varepsilon$ 成立, 因此有 $\mathsf{SD}(\mathsf{2\text{-}Ext}(x_1,x_2) \oplus M_1, \mathsf{2\text{-}Ext}(x_1,x_2) \oplus M_2) \leqslant 2\varepsilon$. 由上述分析可知, $\mathcal{A}'$ 在 Game_4 中获胜的优势为 2ε, 因此有 $\left|\Pr[\mathcal{E}_4] - \frac{1}{2}\right| = 2\varepsilon$.

引理5 敌手在 Game_3 中敌手获胜的优势是可忽略的, 即 $\left|\Pr[\mathcal{E}_3] - \frac{1}{2}\right| \leqslant \text{negl}(\kappa)$.

证明 推论 1 表明若 $\mathcal{A}'$ 能以优势 ψ 在 Game_4 中获胜, 那么存在 $\mathcal{A}$ 能以优势 $2^{l_u}\psi$ 在 Game_3 中获胜; 也就是说 $\mathcal{A}$ 在 Game_3 中获胜的优势是 $\mathcal{A}'$ 在 Game_4 中获胜优势的 2^{l_u} 倍. 推论 2 表明敌手 $\mathcal{A}'$ 在 Game_4 中获胜的优势是 $\psi = 2\varepsilon$. 因此 $\mathcal{A}$ 在 Game_3 中获胜的优势为 $2^{l_u} \cdot 2\varepsilon = 2^{l_u} \cdot 2^{1-l_u-\omega(\log\kappa)} = 2^{1-\omega(\log\kappa)} = \text{negl}(\kappa)$. 由游戏间的不可区分性可知, $\mathcal{A}'$ 在 Game_4 中获胜的优势是可忽略的, 则有 $\left|\Pr[\mathcal{E}_4] - \frac{1}{2}\right| \leqslant \text{negl}(\kappa)$ 成立.

在游戏的挑战阶段之前, $\mathcal{A}^{\text{ent}}$ 所拥有的熵为 l_{Pre}, 其所能回答的熵至多为 l_{Pre}, 则有 $l'_{\text{Pre}} \leqslant l_{\text{Pre}}$. 由于收到挑战密文后, $\mathcal{A}^{\text{ent}}$ 所拥有的熵为 l_{Post}, 且挑战后泄露询问的应答长度至少为 l_u 比特, 则收到挑战密文之后 $\mathcal{A}^{\text{ent}}$ 所能回答的熵为 $l_{\text{Post}} - l_u$, 因此有 $l'_{\text{Post}} \leqslant l_{\text{Post}} - l_u$.

综上所述, 对于挑战前和挑战后的泄露参数 $l'_{\text{Pre}} \leqslant l_{\text{Pre}}$ 和 $l'_{\text{Post}} \leqslant \min(l_{\text{Post}} - l_u, l_t - l_v - \omega(\log\kappa))$, 本文 IBE 机制在状态分离模型下具有挑战后泄露容忍的 CPA 安全性.

4.3 状态分离模型下 CCA 安全的 IBE 机制

对于 IBE 机制而言, CCA 安全性是性能更优属性, 下面将在上述构造的基础上提出状态分离模型下 CCA 安全的 IBE 机制.

令 $\Pi=(\mathrm{Setup},\mathrm{KeyGen},\mathrm{Enc},\mathrm{Dec})$ 是消息空间为 $\{0,1\}^{l_t}$, 密文空间为 $\mathcal{C}$ 和身份空间为 $\mathcal{ID}_1$ 的有熵泄露容忍性的 IBE 机制, 且泄露参数为 l_{Pre} 和 l_{Post}; $H_1:\mathcal{ID}_2\to\mathcal{ID}_1$ 和 $H_2:\mathcal{ID}_2\to\mathcal{ID}_1$ 是两个安全的哈希函数; $\mathsf{2\text{-}Ext}_1:\{0,1\}^{l_t}\times\{0,1\}^{l_t}\to\{0,1\}^{l_u}$ 和 $\mathsf{2\text{-}Ext}_2:\{0,1\}^{l_t}\times\{0,1\}^{l_t}\to\{0,1\}^{l_u}$ 是两个平均情况的 (l_v,ε)- 二源提取器; $\prod_{\mathrm{MAC}}=(\mathrm{Tag},\mathrm{Vrify})$ 是安全的消息验证码, 且密钥空间为 $\{0,1\}^{l_u}$ 和消息空间为 $\mathcal{C}\times\mathcal{C}\times\{0,1\}^{l_t}$.

状态分离模型下 CCA 安全的 IBE 机制 $\Pi_3=(\mathrm{Setup}_3,\mathrm{KeyGen}_3,\mathrm{Enc}_3,\mathrm{Dec}_3)$ 由下述算法组成:

(i) $(\mathrm{Params},\mathrm{msk})\leftarrow\mathrm{Setup}_3(1^\kappa)$

计算 $(\mathrm{Params}',\mathrm{msk}')\leftarrow\mathrm{Setup}(1^\kappa)$, 输出 $\mathrm{Params}=\left(\mathrm{Params}',H_1,H_2,\prod_{\mathrm{MAC}},\mathsf{2\text{-}Ext}_1,\mathsf{2\text{-}Ext}_2\right)$ 为系统公开参数, 并令主私钥为 $\mathrm{msk}=\mathrm{msk}'$.

(ii) $\mathrm{sk}_{\mathrm{id}}\leftarrow\mathrm{KeyGen}_3(\mathrm{msk},\mathrm{id})$

对于 $i=1,2$, 计算 $\mathrm{id}_i=H_i(\mathrm{id})$ 和 $d_i\leftarrow\mathrm{KeyGen}(\mathrm{msk},\mathrm{id}_i)$. 最后输出 $\mathrm{sk}_{\mathrm{id}}=(d_1,d_2)$.

(iii) $C\leftarrow\mathrm{Enc}_3(\mathrm{id},M)$, 其中 $M\in\{0,1\}^{l_u}$

随机选取 $x_1,x_2\leftarrow\{0,1\}^t$, 对于 $i=1,2$, 计算 $\mathrm{id}_i=H_i(\mathrm{id})$ 和 $c_i\leftarrow\mathrm{Enc}(\mathrm{id}_i,x_i)$.

计算 $k_1=\mathsf{2\text{-}Ext}_1(x_1,x_2)$, $c_3=k_1\oplus M$, $k_2=\mathsf{2\text{-}Ext}_2(x_1,x_2)$ 和 $\mathrm{Tag}\leftarrow\mathrm{Tag}(k_2,(c_1,c_2,c_3))$.

输出 $C=(c_1,c_2,c_3,\mathrm{Tag})$.

(iv) $M\leftarrow\mathrm{Dec}_3(\mathrm{sk}_{\mathrm{id}},C)$, 其中 $\mathrm{sk}_{\mathrm{id}}=(d_1,d_2)$ 和 $C=(c_1,c_2,c_3,\mathrm{Tag})$

对于 $i=1,2$, 计算 $x_i\leftarrow\mathrm{Dec}(d_i,c_i)$;

计算 $k_2=\mathsf{2\text{-}Ext}_2(x_1,x_2)$, 若有 $\mathrm{Vrify}(k_2,\mathrm{Tag},(c_1,c_2,c_3))=1$ 成立, 则计算 $k_1=\mathsf{2\text{-}Ext}_1(x_1,x_2)$ 和 $M=k_1\oplus c_3$, 并输出 M 作为相应密文 C 的解密结果; 否则输出终止符 $\perp$.

定理3 若 Π 是泄露参数为 l_{Pre} 和 l_{Post} 的熵泄露容忍的 IBE 机制; $\mathsf{2\text{-}Ext}_1$ 和 $\mathsf{2\text{-}Ext}_2$ 是两个平均情况的 (l_v,ε)- 二源提取器, $\prod_{\mathrm{MAC}}$ 是安全的消息验证码, 那么上述通用构造是状态分离模型下抗挑战后泄露的 CCA 安全的 IBE 机制, 泄露参数 l'_{Pre} 和 l'_{Post} 满足 $l'_{\mathrm{Pre}}\leqslant l_{\mathrm{Pre}}$ 和 $l'_{\mathrm{Post}}\leqslant\min(l_{\mathrm{Post}}-l_u,l_t-l_v-\omega(\log\kappa))$.

定理 3 的证明与定理 2 相类似, 区别在于定理 3 中涉及消息验证码的运算和解密询问, 一旦敌手对密文扩张后进行解密询问, 那么底层消息验证码的安全性将被攻破. 此外, 文献 [21] 已明确给出基于双封装密钥的身份基哈希证明系统和消息验证码的安全属性证明 IBE 机制 CCA 安全性的方法, 因此本文不再赘述定理 3 的证明过程. 需要强调的是, 由 4.2 和 4.3 小节可知, 本文提出了基于现有的任意 IB-HPS 构造挑战后泄露容忍 CCA 安全的 IBE 机制的通用方法, 具体的实例化可通过现有 IB-HPS 的构造实现. 特别地, 若使用 U-IB-HPS 替换本文通用构造中的 IB-HPS, 则可获得抗密钥连续泄露攻击的挑战后泄露容忍 CCA 安全的 IBE 机制的通用构造方法.

5 结论

由于当前对 IBE 机制挑战后泄露容忍性的研究工作较少, 针对该不足, 为获得 IBE 机制挑战后的泄露容忍性, 本文提出了熵泄露容忍 IBE 机制的性质要求和安全性定义, 同时证明了基于 IB-HPS 所构造的 IBE 机制是熵泄露容忍安全的; 在状态分离模型中, 联合熵泄露容忍的 IBE 机制和二源提取器构造了 CPA 安全的 IBE 机制, 并对该机制的挑战后泄露容忍的 CPA 安全性进行了形式化证明; 为设计具备更优安全性的 IBE 机制, 通过在上述构造中增加新的对称密码原语 —— 消息验证码, 设计了具有 CCA 安全性的抗挑战后泄露 IBE 机制的通用构造.

参考文献

1 Zhou Y, Yang B, Xia Z, et al. Novel generic construction of leakage-resilient PKE scheme with CCA security. Des Codes Cryptogr, 2021, 89: 1575–1614

2 Alwen J, Dodis Y, Wichs D. Leakage-resilient public-key cryptography in the bounded-retrieval model. In: Lecture Notes in Computer Science. Berlin: Springer, 2009. 5677: 36–54

3 Dodis Y, Haralambiev K, Adriana L A, et al. Efficient public-key cryptography in the presence of key leakage. In: Lecture Notes in Computer Science. Berlin: Springer, 2010. 6477: 613–631

4 Halevi S, Lin H J. After-the-fact leakage in public-key encryption. In: Lecture Notes in Computer Science. Berlin: Springer, 2011. 6597: 107–124

5 Naor M, Segev G. Public-key cryptosystems resilient to key leakage. SIAM J Comput, 2012, 41: 772–814

6 Zhang Z Y, Chow S S M, Cao Z F. Post-challenge leakage in public-key encryption. Theor Comput Sci, 2015, 572: 25–49

7 Zhao Y, Liang K T, Yang B, et al. CCA secure public key encryption against after-the-fact leakage without NIZK proofs. Secur Commun Networks, 2019, 2019: 1–8

8 Chakraborty S, Rangan C P. Public key encryption resilient to post-challenge leakage and tampering attacks. In: Lecture Notes in Computer Science. Berlin: Springer, 2019. 11405: 23–43

9 Faonio A, Venturi D. Efficient public-key cryptography with bounded leakage and tamper resilience. In: Lecture Notes in Computer Science Springer. Berlin: Springer, 2016. 10031: 877–907

10 Chakraborty S, Paul G, Rangan C P. Efficient compilers for after-the-fact leakage: from CPA to CCA-2 secure pKE to AKE. In: Lecture Notes in Computer Science. Berlin: Springer, 2017. 10342: 343–362

11 Fujisaki E, Kawachi A, Nishimaki R, et al. Post-challenge leakage resilient public-key cryptosystem in split state model. IEICE Trans Fundamentals, 2015, 98: 853–862

12 Alawatugoda J, Boyd C, Stebila D. Continuous after-the-fact leakage-resilient key exchange. In: Lecture Notes in Computer Science. Berlin: Springer, 2014. 8544: 258–273

13 Alawatugoda J, Stebila D, Boyd C. Modelling after-the-fact leakage for key exchange. In: Proceedings of ACM Symposium on Information, Computer and Communications Security, 2014. 207–216

14 Cai C L, Qin X R, Yuen T H, et al. Tight leakage-resilient identity-based encryption under multi-challenge setting. In: Proceedings of ACM Asia Conference on Computer and Communications Security, 2022. 42–53

15 Hou H X, Yang B, Zhang M R, et al. Fully secure wicked identity-based encryption resilient to continual auxiliary-inputs leakage. J Inf Security Appl, 2020, 53: 102521

16 Tomita T, Ogata W, Kurosawa K, et al. CCA-secure leakage-resilient identity-based encryption without q-type assumptions. IEICE Trans Fundamentals, 2020, 103: 1157–1166

17 Li J G, Guo Y Y, Yu Q H, et al. Provably secure identity-based encryption resilient to post-challenge continuous auxiliary input leakage. Secur Comm Networks, 2016, 9: 1016–1024

18 Alwen J, Dodis Y, Naor M, et al. Public-key encryption in the bounded-retrieval model. In: Lecture Notes in Computer Science. Berlin: Springer, 2010. 6110: 113–134

19 Zhou Y W, Yang B, Wang T, et al. Novel updatable identity-based hash proof system and its applications. Theor Comput Sci, 2020, 804: 1–28

20 Boneh D, Katz J. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In: Lecture Notes in Computer Science. Berlin: Springer, 2005. 3376: 87–103

21 Zhou Y W, Yang B, Xia Z, et al. A new construction of leakage-resilient CCA secure IBE scheme. Sci Sin Inform, 2021, 51: 1013–1029 [周彦伟, 杨波, 夏喆, 等. CCA 安全的抗泄露 IBE 机制的新型构造. 中国科学: 信息科学, 2021, 51: 1013–1029]

22 Dodis Y, Ostrovsky R, Reyzin L, et al. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM J Comput, 2008, 38: 97–139

After-the-fact leakage resilience in identity-based encryption

Yanwei ZHOU[1,2,3,4], Zhaolong WANG[1,2], Zirui QIAO[1*], Bo YANG[1*], Chunxiang GU[3], Zhe XIA[5] & Mingwu ZHANG[4,6]

1. *School of Computer Science, Shaanxi Normal University, Xi'an* 710062*, China*;
2. *State Key Laboratory of Cryptography, Beijing* 100878*, China*;
3. *Henan Key Laboratory of Network Cryptography Technology, Zhengzhou* 450040*, China*;
4. *Guangxi Key Laboratory of Cryptography and Information Security, Guilin* 541004*, China*;
5. *School of Computer Science and Technology, Wuhan University of Technology, Wuhan* 430070*, China*;
6. *School of Computer, Hubei University of Technology, Wuhan* 430068*, China*

* Corresponding author. E-mail: qzr_snnu@163.com, byang@snnu.edu.cn

Abstract In encryption scheme research, we always assume that leakage is achieved by the adversary before the challenge stage and that the leakage query cannot be submitted after seeing the challenge ciphertext. The above constraints seem necessary but limit the effectiveness of the results because, in an actual scenario, the adversary usually tries to obtain the corresponding key information through various methods after accessing the ciphertext; therefore, after-the-fact leakage is closer to the actual situation. In this paper, the requirements and security definition of an entropic leakage-resilient identity-based encryption (IBE) scheme are proposed. Thereafter, an IBE scheme with after-the-fact leakage resilience and chosen-plaintext attack (CPA) security is created using an entropic leakage-resilient IBE scheme and two-source extractor; the after-the-fact leakage-resilient CPA security of the proposed scheme can be proved from the corresponding security of the underlying cryptographic tools. To further obtain an IBE scheme with better security, based on the aforementioned generic construction of after-the-fact leakage-resilient IBE, we take message authentication codes as another basic tool to create the generic construction of a chosen-ciphertext attack secure IBE scheme with after-the-fact leakage resilience.

Keywords after-the-fact leakage resilience, identity-based hash proof system, identity-based encryption, entropic leakage resilience