软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2023,34(2):818−832 [doi: 10.13328/j.cnki.jos.006398]

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

CCA 安全的抗连续泄露的广播密钥封装机制*

乔子芮[1], 杨启良[1], 周彦伟[1,2,3], 杨 波[1], 夏 喆[4], 张明武[2,3]

[1](陕西师范大学 计算机科学学院, 陕西 西安 710062)

[2](广西密码学与信息安全重点实验室(桂林电子科技大学), 广西 桂林 541004)

[3](密码科学技术国家重点实验室, 北京 100878)

[4](武汉理工大学 计算机科学与技术学院, 湖北 武汉 430070)

通信作者: 周彦伟, E-mail: zyw@snnu.edu.cn; 杨波, E-mail: byang@snnu.edu.cn

摘 要: 传统公钥基础设施中的证书复杂管理和身份基密码体制中的密钥托管等问题在基于证书的密码体制下得到了很好的解决, 因此无证书密码体制近年来得到了广泛关注. 此外, 在现实应用中, 攻击者基于冷启动、边信道等各种各样的泄露攻击获得密码机制内部敏感状态(如秘密钥等)的泄露信息, 导致在传统理想模型下被证明安全的密码机制不再具有相应的安全性. 此外, 由于广播通信模式具有较高的消息通信效率, 多个具有广播通信功能的密码原语相继被提出. 针对基于证书密钥封装机制对泄露容忍性和广播通信等性能的需求, 提出抗泄露的基于证书的广播密钥封装机制的实例化构造, 并基于判定的 Diffie-Hellman 困难性假设对其选择密文攻击下的安全性进行了证明. 此外, 为进一步增强该构造的实用性, 研究了广播密钥封装机制的连续泄露容忍性, 通过定期更新用户密钥的方式实现了对连续泄露攻击的抵抗目标. 与现有工作的分析对比表明, 该构造在保证安全性可证明的基础上, 不仅实现了抵抗泄露攻击和广播通信的功能, 而且拥有较高的计算效率.

关键词: 基于证书的密钥封装机制; 广播通信; 连续泄露容忍性; 选择密文攻击

中图法分类号: TP309

中文引用格式: 乔子芮, 杨启良, 周彦伟, 杨波, 夏喆, 张明武. CCA 安全的抗连续泄露的广播密钥封装机制. 软件学报, 2023, 34(2): 818–832. http://www.jos.org.cn/1000-9825/6398.htm

英文引用格式: Qiao ZR, Yang QL, Zhou YW, Yang B, Xia Z, Zhang MW. Continuous Leakage-resilient Broadcast Key-encapsulation Mechanism with CCA Security. Ruan Jian Xue Bao/Journal of Software, 2023, 34(2): 818–832 (in Chinese). http://www.jos.org.cn/1000-9825/6398.htm

Continuous Leakage-resilient Broadcast Key-encapsulation Mechanism with CCA Security

QIAO Zi-Rui[1], YANG Qi-Liang[1], ZHOU Yan-Wei[1,2,3], YANG Bo[1], XIA Zhe[4], ZHANG Ming-Wu[2,3]

[1](School of Computer Science, Shaanxi Normal University, Xi'an 710062, China)

[2](Guangxi Key Laboratory of Cryptography and Information Security (Guilin University of Electronic Technology), Guilin 541004, China)

[3](State Key Laboratory of Cryptology, Beijing 100878, China)

[4](School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China)

Abstract: Certificate-based cryptography which is attracted great interest can solve the certificate management issue of the traditional public-key cryptography system, at the same time, which can also avoid the key escrow in the identity-based cryptography, thus, it has attracted attention of cryptography researchers. The traditional security models assume that any adversary cannot obtain the leakage information on the internal secret states, such as secret keys, however, some leakage can be leaked through various leakage attacks in the

* 基金项目: 国家重点研发计划(2017YFB0802000); 国家自然科学基金(62272287,61802242, U2001205); 广西密码学与信息安全重点实验室研究课题(GCIS202108); 河南省网络密码技术重点实验室研究课题(LNCT2021-A04)

收稿时间: 2021-04-13; 修改时间: 2021-05-24, 2021-06-07; 采用时间: 2021-06-11

actual environment. In addition, many cryptographic schemes with broadcast communication function were created, because broadcast communication has higher efficiency of message transmission. To further provide leakage resilience and broadcast communication for certificate-based broadcast key encapsulation mechanism (CB-BKEM), a concrete construction of CB-BKEM is proposed, and the leakage-resilient chosen-ciphertext attacks security is proved based on decisional Diffie-Hellman assumption. To further improve the practicability of CB-BKEM, continuous leakage-resilient CB-BKEM is researched, and the continuous leakage resilience of CB-BKEM can be obtained by performing key update. The performance analysis shows that the proposed construction has higher computational efficiency while maintaining the provable security, the leakage resilience and the broadcast communication.

Key words: certificate-based key-encapsulation mechanism; broadcast communication; continuous leakage resilience; chosen-ciphertext attacks

为了提升数据加密运算的效率, 在公钥密码体制和对称密码体制研究的基础上, 密码学者提出了称为混合加密的新密码学原语. 该原语继承了公钥密码体制密钥管理方面的优势, 同时具有对称密码体制运算效率高的优点. 通常情况下, 混合加密算法由密钥封装和数据封装两部分组成, 其中, 密钥封装机制作为公钥密码原语, 主要完成密钥的管理功能; 数据封装机制是对称密码体制, 主要负责加解密运算. 伴随着互联网技术的发展, 由于混合加密机制在密钥管理和计算效率等方面的优势, 能够充分地满足数据安全存储及访问授权的实际需求, 因此促进了混合加密机制的研究进程. 近年来, 关于混合加密机制核心组件密钥封装机制的研究已成为密码学领域当前的研究热点.

1 引 言

传统密码学机制的安全性证明中, 均假设攻击者对密码算法具体运行过程所涉及的秘密信息是未知的, 只能接触算法的相应输出和输入, 即对攻击者而言, 密码机制是以“黑盒”的模式运行. 然而, 伴随着冷启动、边信道等各种各样泄露攻击的出现, 使得攻击者能够获得密码机制参与者敏感状态(如秘密钥等)的部分泄露信息, 导致传统密码机制中所假设的“黑盒”运行环境在实际中将变成“灰盒”模式(由于敌手能够获得相应的秘密信息导致密码机制的运行过程不再是完全不可见的), 使得在理想的安全模型下被证明安全的密码原语不再保持其所声称的安全性. 近年来, 为进一步提升密码机制抵抗泄露攻击的能力, 研究者提出了抗泄露密码学的概念, 致力于密码原语抗泄露性的相关研究, 多个具备抗泄露性能的密码原语相继被提出, 如抗泄露的公钥加密[1,2]、抗泄露的身份基加密[3−5]、抗泄露的属性基加密[6−8]、抗泄露的无证书加密[8,9]、抗泄露的密钥协商[10]等.

传统公钥基础设施的密钥分发、管理和回收等繁琐的操作过程, 导致其存在证书的复杂管理问题. 为避免上述不足, Shamir 在 1985 年提出了身份基密码体制, 其中, 用户公钥将从用户的公开信息中获得. 然而, 该体制中需设定可信的密钥生成中心(key generation center, KGC)负责用户的注册, 并为其生成相应的完整秘密钥, 使得 KGC 能够掌握所有用户的秘密钥, 导致其存在密钥托管的不足. 针对上述不足, Gentry 在 2003 年提出了基于证书密码体制的新密码学原语, 该机制不仅避免了 PKI 中证书的复杂管理问题, 而且规避了身份基密码体制的密钥托管的不足. 近年来, 基于证书的加密(certificate-based encryption, CBE)、基于证书的签名、基于证书的混合加密等密码机制得到了广泛的关注, 其中, 对基于证书密钥封装机制(certificate-based key encapsulation mechanism, CB-KEM)的相关研究工作较少. 为提升基于证书混合加密机制的抗泄露能力, 本文将研究抗泄露的 CB-KEM; 并且为达到抵抗连续泄露攻击的目的, 在抗泄露 CB-KEM 研究的基础上, 本文将研究抗连续泄露的 CB-KEM, 以达到增强 CB-KEM 实用性的目的.

1.1 研究现状

在文献[11]中, Yu 等人给出了抗泄露 CBE 机制的形式化定义和泄露容忍的安全模型; 同时, 基于二源提取器设计了第一个抗泄露的 CBE 机制, 并基于判定的 Diffie-Hellman (decisional Diffie-Hellman, DDH)假设对上述构造的安全性进行了证明. 在文献[12]中, 为提升 CBE 机制的抗泄露能力, Li 等人提出了抵抗连续泄露攻击的 CBE 机制的具体构造, 并基于 BDHI (bilinear Diffie-Hellman inversion)困难性假设对其 CCA 安全性进行

了形式化证明. 针对物联网的安全性通信需求, 文献[13]设计了抗泄露 CBE 机制, 实现了对用户私钥、随机数和系统主密钥的泄露抵抗, 并在双线性群模型中对方案的安全性进行了证明. 为满足大规模无线终端对密码算法的高计算效率需求, Zhou 等人[14]设计了抗连续泄露的 CBE 机制, 并基于 DDH 困难性假设对方案的选择密文攻击(chosen-ciphertext attacks, CCA)的安全性进行了证明. 文献[15]设计了 CB-KEM 的高效构造方法, 并对相应构造的安全性进行了详细证明. 分析表明, 该机制具有较高的运算效率. 为满足混合加密的抗泄露性需求, 文献[16]设计了抵抗泄露攻击的 CB-KEM, 该机制在敌手获得用户私钥的相关泄露信息后, 依然保持其所声称的安全性, 然而双线性映射的所有导致其计算效率较低.

伴随着广播通信技术的发展, 增强了消息传输的效率, 比传统方式更有效、更方便、更快捷. 文献[17]设计了基于证书的广播加密(certificate-based broadcast encryption, CBBE)机制, 并在标准模型下, 基于非静态的 q-ABDHE (q-augmented bilinear Diffie-Hellman exponent)困难性假设证明了该方案的安全性; 此外, 与现有的相关方案相比, 该方案在性能方面具有一定的优势; 同时, 以云计算环境下的数据存储访问为例, 详细介绍了该机制在真实环境中的具体应用. 为了保护通信用户的身份信息, 文献[18,19]设计了两个具备匿名性的 CBBE 机制, 其中, 文献[19]中解密的运算量是固定的. 然而, 上述机制是基于双线性映射构造的, 在计算效率方面依然存在可进一步提升的空间. 文献[20]提出了密钥和密文尺寸固定的 CBBE 机制. 基于无证书密码体制, 文献[21]提出了抗泄露的无证书密钥封装机制的具体构造, 并以该方案为底层基础工具, 设计了混合加密机制和密钥协商协议的通用构造方法.

综上所述, 目前对 CBE 机制的抗泄露能力和广播通信功能等方面的研究成果较为丰富, 然而对 CB-KEM 相关性能(如抗泄露性、广播通信等)的研究并未引起研究者的广泛关注. 因此, 本文将研究具备泄露容忍性和广播通信功能的 CB-KEM. 此外, 由于双线性映射的使用在一定程度上会增加相应构造的计算量, 为了获得更佳的计算效率, 本文将不使用计算效率较低的双线性映射运算. 在此基础上, 设计具有泄露容忍性和广播通信功能的 CB-KEM 的具体实例.

1.2 我们的工作

鉴于广播通信模式的高效率优势和泄露容忍的强安全性能, 本文将考虑 CB-KEM 的广播通信和泄露容忍等功能, 提出一种新密码学原语——抵抗泄露攻击的基于证书的广播密钥封装机制(certificate-based broadcast key encapsulation mechanism, CB-BKEM), 同时给出具体的形式化定义及抗泄露的安全模型, 并在此基础上开展下述研究工作.

(1) 对于加密机制而言, CCA 安全性是更加实用的安全属性, 本文设计抗泄露 CB-BKEM 的具体构造, 并在标准模型下, 基于 DDH 困难性假设对该实例泄露容忍的 CCA 安全性进行形式化证明; 此外, 不使用计算效率较低的双线性映射, 一定程度上能够提升本文构造的计算效率. 特别地, 每一个抗泄露 CB-BKEM 方案将蕴含一个抗泄露的基于证书的广播混合加密机制的实例化构造;

(2) 为抵抗连续泄露攻击, 本文在上述抗泄露 CB-BKEM 方案的基础上, 将设计抗连续泄露攻击的 CB-BKEM 方案, 为 CB-BKEM 提供抵抗连续泄露攻击能力的同时, 增强其实用性;

(3) 以云计算下的数据访问授权机制为例, 对抗泄露 CB-BKEM 的具体应用进行了探讨, 提出云数据的抗泄露广播授权机制, 并对协议的执行过程进行简要描述.

2 基础知识

本节主要介绍强随机性提取器和 DDH 困难性假设等基础知识.

2.1 强随机性提取器

令 $H_\infty(M)=-\log(\mathrm{Max}_m\Pr[M=m])$是变量 M 的最小熵, $\tilde{H}_\infty(M\mid N)=-\log(\boldsymbol{E}_{n\leftarrow M}[2^{-H_\infty(M|N=n)}])$ 是变量 M 在已知 N 时的平均最小熵, 令 $SD(M,N)=\frac{1}{2}\sum_{\omega\in\Omega}|\Pr[M=\omega]-\Pr[N=\omega]|$表示有限域$\Omega$上变量 M 和 N 间的统计距离.

定理 1. 已知 3 个变量 M、N 和 Z, 其中, 变量 Z 的取值至多有 2^l 个, 那么有下述关系成立:

$$\tilde{H}_\infty(M\mid(N,Z))\geqslant\tilde{H}_\infty(M\mid N)-l.$$

定义 1(强随机性提取器). 对于两个随机的变量 $M\in\{0,1\}^{l_n}$ 和 N, 对于 $X\leftarrow_R\{0,1\}^{l_t}$ 和 $Y\leftarrow_R\{0,1\}^{l_m}$, 若有 $\tilde{H}_\infty(M\mid N)\geqslant k$ 和 $SD((Ext(M,X),X,N),(Y,X,N))\leqslant\varepsilon$成立, 那么称 $Ext:\{0,1\}^{l_n}\times\{0,1\}^{l_t}\to\{0,1\}^{l_m}$ 是平均情况的(k,ε)-强随机性提取器, 其中, $\leftarrow_R$表示均匀随机选取.

2.2 安全性假设

定义 2(判定的 Diffie-Hellman 假设, DDH 假设). 令 G 是生成元和阶分别为 g 与素数 p 的乘法循环群. 对于任意给定的两个元组(g,g^m,g^n,g^{mn})和(g,g^m,g^n,g^x)(其中, $m,n,x\in Z_p^*$), DDH 问题的目标是判断 $g^{mn}=g^x$ 是否成立. 对于任意的算法$\mathcal{A}$, DDH 假设意味着其成功解决 DDH 问题的优势:

$$Adv^{DDH}(\mathcal{A})=\Pr[\mathcal{A}(g,g^m,g^n,g^{mn})=1]-\Pr[\mathcal{A}(g,g^m,g^n,g^x)=1]$$

是可忽略的.

3 基于证书广播密钥封装机制

本节将在现有抗泄露 CBE 机制[11,12]的基础上, 给出 CB-BKEM 的形式化定义及泄露容忍的安全模型. 为了方便理解 CB-BKEM 的安全模型, 首先对基于证书密码体制的敌手分类情况进行介绍.

3.1 敌手分类

对于基于证书密码体制而言, 无需要求完成系统建立的 KGC 是完全安全可信的, 那么恶意用户和恶意 KGC 都能对 CB-BKEM 发起相应的攻击, 其中, 恶意用户能够伪装成其他合法用户, 恶意 KGC 同样能够为用户实现注册服务. 具体地, 恶意用户基于公开钥替换的途径伪装成其他合法用户对 CB-BKEM 进行攻击; 恶意 KGC 使用已掌握的主私钥对 CB-BKEM 进行攻击. 因此, CB-BKEM 将受到两类概率多项式时间敌手的攻击, 分别记为$\mathcal{F}^1$ 和$\mathcal{F}^2$ 两类.

(1) 恶意用户$\mathcal{F}^1$ 可对用户公开钥进行替换, 但其无法接触主密钥. $\mathcal{F}^1$ 能够对除挑战身份之外的任意身份进行秘密钥和证书生成询问, 且不能在挑战之前对挑战身份所对应的公开钥进行替换;

(2) 恶意 KGC $\mathcal{F}^2$ 拥有主密钥, 但不能替换用户公开钥. $\mathcal{F}^2$ 能够对除挑战身份之外的任意身份进行秘密钥生成询问. 特别地, 由于$\mathcal{F}^2$ 已掌握系统主密钥, 因此无需进行证书生成询问.

3.2 形式化定义

一个 CB-BKEM 由下述 5 个概率多项式时间算法组成.

(1) $(Params,msk)\leftarrow Setup(1^\kappa)$. 初始化算法输入安全参数$\kappa$, 输出 $Params$ 和 msk, 其中, $Params$ 是公开参数, msk 是主私钥;

(2) $(pk_{id},sk_{id})\leftarrow KeyGen(id,\kappa)$. 密钥生成算法的输入是身份 id 和κ, 输出(pk_{id},sk_{id}), 其中, pk_{id} 是公开钥, sk_{id} 是秘密钥;

(3) $Cert_{id}\leftarrow CertGen(id,sk_{id},pk_{id})$. 证书生成算法的输入是 id、pk_{id} 和 msk, 为用户生成证书 $Cert_{id}$;

(4) $(C,k)\leftarrow Encap(\boldsymbol{ID},\boldsymbol{PK})$. 广播密钥封装算法的输入是接收者身份集合 $\boldsymbol{ID}=\{id_1,\ldots,id_n\}$及对应的公开钥集合 $\boldsymbol{PK}=\{pk_1,\ldots,pk_n\}$, 输出相应的封装密文 C 及封装密钥 k;

(5) $k\leftarrow Decap(sk_{id},Cert_{id},C)$. 解封装算法的输入是 sk_{id}、$Cert_{id}$ 和 C, 输出对应的封装密钥 k 或无效符号$\perp$.

图 1 所示为 CB-BKEM 中各算法间的运行关系, 其中, KGC 主要完成系统初始化和证书生成算法的运行, 主要负责用户的注册, 并为其生成相应的证书; 发送者通过运行封装算法为接收者集合中的每个用户生成相应的封装密钥; 集合中的任意接收者通过对封装密文进行解封装操作得到对应的封装密钥.

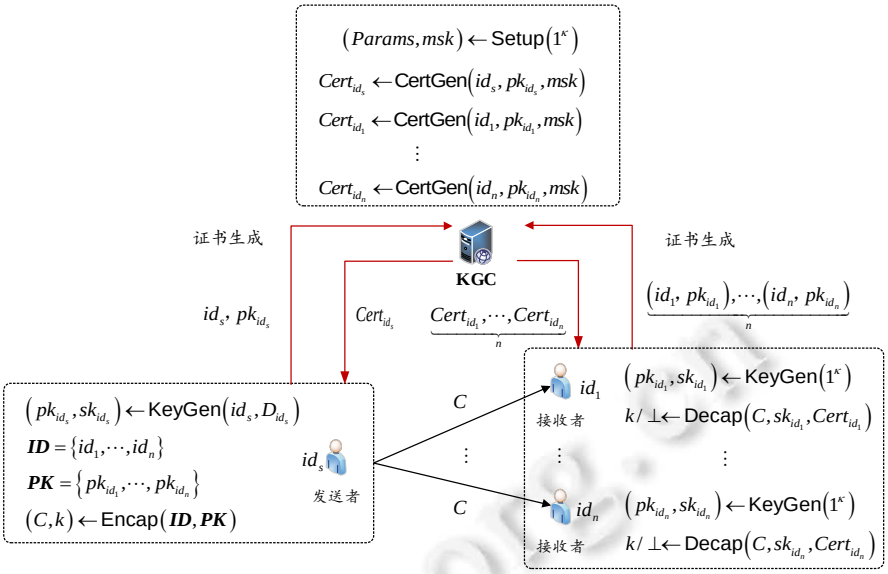

图 1 CB-BKEM 中各算法间的运行关系

3.3 正确性

CB-BKEM 的正确性要求，对于$(Params,msk)\leftarrow Setup(1^{\kappa})$和 $\boldsymbol{ID}=\{id_1,\ldots,id_n\}$，有:

$$\Pr\left[k\neq k'\left|\begin{array}{l}(pk_i,sk_i)\leftarrow KeyGen(id_i)_{i=1,2,\ldots,n}\\ Cert_i\leftarrow CertGen(msk,id_i,pk_i)_{i=1,2,\ldots,n}\\ (C,k)\leftarrow Encap(ID,PK)_{PK=\{pk_1,\ldots,pk_n\}}\\ k'\leftarrow Decap(sk_i,Cert_i,C)_{i=1,2,\ldots,n}\end{array}\right.\right]\leqslant negl(\kappa)$$

成立.

3.4 安全性

对于敌手$\mathcal{F}^1$和$\mathcal{F}^2$, CB-BKEM 泄露容忍的安全模型分别通过下述两个实验来加以描述.

3.4.1 敌手$\mathcal{F}^1$攻击下泄露容忍的 CCA 安全性

在 $Exp^{CCA}_{CB\text{-}BKEM,\mathcal{F}^1}(\kappa,\lambda)$ 中，若$\mathcal{F}^1$ 获胜的优势是可忽略的，那么在存在泄露的情况下，相应的 CB-BKEM 在选择密文攻击(chosen-ciphertext attacks, CCA)下封装密钥具有不可区分性.

$Exp^{CCA}_{CB\text{-}BKEM,\mathcal{F}^1}(\kappa,\lambda)$:

$(Params,msk)\leftarrow Setup(\kappa)$;

$\boldsymbol{ID}^*=\{id_1^*,\ldots,id_n^*\}\leftarrow(\mathcal{F}^1)^{\mathcal{O}^{KenGen}(\cdot),\mathcal{O}^{CertGen}(\cdot),\mathcal{O}^{Decap}(\cdot),\mathcal{O}^{\lambda,\kappa}_{sk_{id}}(\cdot)}(Params)$;

$\boldsymbol{PK}^*=\{pk_1^*,\ldots,pk_n^*\}$, $pk_i^*\leftarrow KeyGen(id_i^*,msk)_{i=1,2,\ldots,n}$;

$(C^*,k_1)=Encap(\boldsymbol{ID}^*,\boldsymbol{PK}^*)$和$k_0\leftarrow_R\mathcal{K}$;

$\beta\leftarrow_R\{0,1\}$;

$\beta'\leftarrow(\mathcal{F}^1)^{\mathcal{O}^{KeyGen}_{id\notin\boldsymbol{ID}^*}(\cdot),\mathcal{O}^{CertGen}_{id\notin\boldsymbol{ID}^*}(\cdot),\mathcal{O}^{Decap}_{\neq(\boldsymbol{ID}^*,C^*)}(\cdot)}(Params,C^*,k_\beta)$;

If $\beta'=\beta$, *output* 1; Otherwise, *output* 0.

其中，$\mathcal{O}^{KeyGen}(\cdot)$表示$\mathcal{F}^1$ 向挑战者提出关于任意身份 id $(id\in\mathcal{ID})$的秘密钥生成询问；$\mathcal{O}^{KeyGen}_{id\notin\boldsymbol{ID}^*}(\cdot)$ 表示$\mathcal{F}^1$ 向挑战者执行除集合 $\boldsymbol{ID}^*=\{id_1^*,\ldots,id_n^*\}$ 之外任意身份 id $(id\in\mathcal{ID}\wedge id\notin\boldsymbol{ID}^*)$的秘密钥生成询问; $\mathcal{O}^{CertGen}(\cdot)$表示$\mathcal{F}^1$ 向挑战者

提出关于任意身份 id ($id\in\mathcal{ID}$)的证书生成询问; $\mathcal{O}^{CertGen}_{id\notin \boldsymbol{ID}^*}(\cdot)$ 表示$\mathcal{F}^1$ 向挑战者执行除集合 $\boldsymbol{ID}^*=\{id_1^*,...,id_n^*\}$ 之外任何身份 id ($id\in\mathcal{ID}\wedge id\notin\boldsymbol{ID}^*$)的证书生成询问; $\mathcal{O}^{Decap}(\cdot)$表示$\mathcal{F}^1$ 向挑战者提出关于任意身份密文对(id,C)的解封装询问; $\mathcal{O}^{Decap}_{\neq(\boldsymbol{ID}^*,C^*)}(\cdot)$ 表示敌手$\mathcal{F}^1$ 向挑战者执行除($\boldsymbol{ID}^*$,C^*)之外其他身份密文对($\boldsymbol{ID}$,C)≠($\boldsymbol{ID}^*$,C^*)的解封装询问; $\mathcal{F}^1$ 向泄露谕言机 $\mathcal{O}^{\lambda,\kappa}_{sk_{id}}(\cdot)$ 提交泄露函数 $f_i(\cdot)$, $\mathcal{O}^{\lambda,\kappa}_{sk_{id}}(\cdot)$ 检测关于同一 sk_{id} 的泄露信息是否超过泄露参数λ, 若未超过, 则返回相应的泄露 $f_i(sk_{id})$给$\mathcal{F}^1$.

在 $Exp^{CCA}_{CB\text{-}BKEM,\mathcal{F}^1}(\kappa,\lambda)$ 中, $\mathcal{F}^1$ 获胜的优势定义为

$$Adv^{CCA}_{CB\text{-}BKEM,\mathcal{F}^1}(\kappa,\lambda)=|\Pr[Exp^{CCA}_{CB\text{-}BKEM,\mathcal{F}^1}(\kappa,\lambda)=1]-\Pr[Exp^{CCA}_{CB\text{-}BKEM,\mathcal{F}^1}(\kappa,\lambda)=0]|.$$

3.4.2 敌手$\mathcal{F}^2$攻击下泄露容忍的 CCA 安全性

在 $Exp^{CCA}_{CB\text{-}BKEM,\mathcal{F}^2}(\kappa,\lambda)$ 中, 若$\mathcal{F}^2$ 获胜的优势是可忽略的, 那么在存在泄露的情况下, 相应的 CB-BKEM 在选择密文攻击下封装密钥具有不可区分性.

$Exp^{CCA}_{CB\text{-}BKEM,\mathcal{F}^2}(\kappa,\lambda_1)$:

$(Params,msk)\leftarrow Setup(\kappa)$;

$\boldsymbol{ID}^*=\{id_1^*,...,id_n^*\}\leftarrow(\mathcal{F}^2)^{\mathcal{O}^{KeyGen}(\cdot),\mathcal{O}^{Decap}(\cdot),\mathcal{O}^{\lambda,\kappa}_{sk_{id}}(\cdot)}(Params,msk)$;

$\boldsymbol{PK}^*=\{pk_1^*,...,pk_n^*\}$, $pk_i^*\leftarrow KeyGen(id_i^*,msk)_{i=1,2,...,n}$;

$(C^*,k_1)=Encap(\boldsymbol{ID}^*,\boldsymbol{PK}^*)$和$k_0\leftarrow_R\mathcal{K}$;

$\beta\leftarrow_R\{0,1\}$;

$\beta'\leftarrow(\mathcal{F}^2)^{\mathcal{O}^{KeyGen}_{id\notin\boldsymbol{ID}^*}(\cdot),\mathcal{O}^{Decap}_{\neq(\boldsymbol{ID}^*,C^*)}(\cdot)}(Params,msk,C^*,k_\beta)$;

If $\beta=\beta$, *output* 1; Otherwise, *output* 0.

在 $Exp^{CCA}_{CB\text{-}BKEM,\mathcal{F}^2}(\kappa,\lambda)$ 中, $\mathcal{F}^2$ 获胜的优势定义为

$$Adv^{CCA}_{CB\text{-}BKEM,\mathcal{F}^2}(\kappa,\lambda)=|\Pr[Exp^{CCA}_{CB\text{-}BKEM,\mathcal{F}^2}(\kappa,\lambda)=1]-\Pr[Exp^{CCA}_{CB\text{-}BKEM,\mathcal{F}^2}(\kappa,\lambda)=0]|.$$

定义 6(泄露容忍的选择密文攻击安全性). 对于$\mathcal{F}^1$ 和$\mathcal{F}^2$, 若有

$$Adv^{CCA}_{CB\text{-}BKEM,\mathcal{F}^1}(\kappa,\lambda)\leqslant negl(\kappa),\ Adv^{CCA}_{CB\text{-}BKEM,\mathcal{F}^2}(\kappa,\lambda)\leqslant negl(\kappa)$$

成立, 那么在存在泄露的环境下相应的 CB-BKEM 具有 CCA 安全性.

相对于 CCA 安全性而言, 在 CB-BKEM 的选择明文攻击(chosen-plaintext attacks, CPA)安全性实验中, 敌手在任何阶段都不能对任意的封装密文进行解封装询问, 即不能访问谕言机$\mathcal{O}^{Decap}(\cdot)$. 为方便理解 CB-BKEM 的 CCA 与 CPA 安全性实验间的区别与联系, 以$\mathcal{F}^1$ 为例, 下文将给出 CB-BKEM 在$\mathcal{F}^1$ 攻击下泄露容忍的 CPA 安全性实验. 相类似地, 根据$\mathcal{F}^2$ 攻击下泄露容忍的 CCA 安全性实验, 能够得到$\mathcal{F}^2$ 攻击下泄露容忍的 CPA 安全性实验.

$Exp^{CPA}_{CB\text{-}BKEM,\mathcal{F}^1}(\kappa,\lambda)$:

$(Params,msk)\leftarrow Setup(\kappa)$;

$\boldsymbol{ID}^*=\{id_1^*,...,id_n^*\}\leftarrow(\mathcal{F}^1)^{\mathcal{O}^{KenGen}(\cdot),\mathcal{O}^{CertGen}(\cdot),\mathcal{O}^{\lambda,\kappa}_{sk_{id}}(\cdot)}(Params)$;

$\boldsymbol{PK}^*=\{pk_1^*,...,pk_n^*\}$, $pk_i^*\leftarrow KeyGen(id_i^*,msk)_{i=1,2,...,n}$;

$(C^*,k_1)=Encap(\boldsymbol{ID}^*,\boldsymbol{PK}^*)$和$k_0\leftarrow_R\mathcal{K}$;

$\beta\leftarrow_R\{0,1\}$;

$\beta'\leftarrow(\mathcal{F}^1)^{\mathcal{O}^{KeyGen}_{id\notin\boldsymbol{ID}^*}(\cdot),\mathcal{O}^{CertGen}_{id\notin\boldsymbol{ID}^*}(\cdot)}(Params,C^*,k_\beta)$;

If $\beta'=\beta$, *output* 1; Otherwise, *output* 0.

4 抗泄露的广播密钥封装机制

本节将给出 CCA 安全的抗泄露的广播密钥封装机制的具体实例, 并基于经典的 DDH 困难性假设, 对本文构造的安全性进行形式化证明.

4.1 具体构造

(1) 初始化算法$(Params,msk)\leftarrow Setup(1^{\kappa})$进行下述操作.

令 G 是生成元和阶分别为 g 与素数 p 的乘法循环群; 选取 4 个密码学哈希函数 $H:G\rightarrow\{0,1\}^{l_k}$, $H_1:\mathcal{ID}\rightarrow Z_p^*$, $H_2:\mathcal{ID}\times G\times G\rightarrow Z_p^*$和$H_3:\{0,1\}^*\rightarrow Z_p^*$, 其中, $\mathcal{ID}$是用户身份空间. 令 $Ext:G_2\times\{0,1\}^{l_t}\rightarrow\{0,1\}^{l_k}$ 是平均情况的$(\log q-\lambda,\varepsilon)$强随机性提取器, 其中, λ是泄露参数, ε是κ上可忽略的值.

选取随机数 $\alpha\leftarrow_R Z_p^*$ 和 $g_1\leftarrow_R G$, 计算 $g_2=g^{\alpha}$; 令主密钥 $msk=\alpha$, 并公开系统参数:

$$Params=\{p,G,g,g_1,g_2,H,H_1,H_2,H_3,Ext\}.$$

(2) 密钥生成算法$(pk_{id},sk_{id})\leftarrow KeyGen(id)$进行下述操作.

用户 U_{id} 选取随机数 $a,b,c,d\leftarrow_R Z_p^*$, 计算 $pk_{id}^1=g^{aH_1(id)}g_1^b$和$pk_{id}^2=g^{cH_1(id)}g_1^d$, 最后输出$(sk_{id},pk_{id})$, 其中, $sk_{id}=(a,b,c,d)$, $pk_{id}=(pk_{id}^1,pk_{id}^2)$.

(3) 证书生成算法 $Cert_{id}\leftarrow CertGen(msk,id,pk_{id})$进行下述操作.

KGC 选取随机数 $t_{id}\leftarrow_R Z_p^*$, 并计算 $T_{id}=g^{t_{id}}$和$u_{id}=t_{id}+\alpha H_2(id,T_{id},pk_{id}^1,pk_{id}^2)$; 然后输出用户 U_{id} 的证书 $Cert_{id}=\{T_{id},u_{id}\}$, 其中, u_{id}是证书的核心秘密部分, T_{id}是公开的证书合法性辅助验证信息(与用户公开钥 pk_{id}一起对外公开).

(4) 广播密钥封装算法$(C,k)\leftarrow Encap(\boldsymbol{ID},\boldsymbol{PK})$进行下述操作.

令$\boldsymbol{ID}=\{id_1,\ldots,id_n\}$表示接收者身份集合及相对应的公开钥集合为$\boldsymbol{PK_{ID}}=\{pk_1,\ldots,pk_n\}$, 并且相应的证书辅助验证参数集合为 $\boldsymbol{T}=\{T_1,\ldots,T_n\}$.

选取随机数 $r,\eta\leftarrow_R Z_p^*$, 计算$U_1=g^r$, $U_2=g_1^r$和$k=H(U_2^{\eta})$.

选取随机字符串 $S\leftarrow_R\{0,1\}^{l_t}$, 对于 $i=1,\ldots,n$, 计算:

$$N_i=(pk_i^2)^r(pk_i^1T_ig_2^{H_2(id_i,T_i,pk_i^1,pk_i^2)})^r,\ V_i=(pk_i^1)^r(pk_i^2T_ig_2^{H_2(id_i,T_i,pk_i^1,pk_i^2)})^{r\omega_i}\text{和}W_i=Ext(N_i,S)\oplus k,$$

其中, $\omega_i=H_3(id_i,U_1,U_2,W_i,pk_i,S)$.

令集合 $\boldsymbol{W}=\{W_1,\ldots,W_n\}$和 $\boldsymbol{V}=\{V_1,\ldots,V_n\}$, 输出封装密文 $C=(\boldsymbol{ID},U_1,U_2,\boldsymbol{W},\boldsymbol{V},S)$及封装密钥 k.

(5) 解封装算法 $k\leftarrow Decap(C,sk_{id},Cert_{id})$进行下述操作.

收到封装密文 $C=(\boldsymbol{ID},U_1,U_2,\boldsymbol{W},\boldsymbol{V},S)$后, 接收者 id_i 计算$\omega_i=H_3(id_i,U_1,U_2,W_i,pk_i,S)$.

若等式 $V_i=U_1^{(a+\omega_ic)H(id_i)+\omega_iu_i}U_2^{b+\omega_id}$ 成立, 计算 $N_i'=U_1^{((a+c)H(id_i)+u_i)}U_2^{b+d}$, 并输出封装密钥 $k=Ext(N_i',S)\oplus W_i$; 否则, 输出特殊的符号$\perp$.

4.2 正确性

本文 CCA 安全的抗泄露 CB-BKEM 构造的正确性可由下述等式获得.

$$\begin{aligned}N_i'&=U_1^{((a+c)H(id_i)+u_i)}U_2^{b+d}\\&=g^{r((a+c)H(id_i)+u_i)}g_1^{r(b+d)}\\&=g^{raH(id_i)}g^{rcH(id_i)}g^{ru_i}g_1^{rb}g_1^{rd}\\&=(g^{aH(id_i)}g_1^b)^rg^{r(t_i+\alpha H_2(id_i,T_i,pk_i^1,pk_i^2))}(g^{cH(id_i)}g_1^d)^r\\&=(pk_i^1g^{t_i}g_2^{H_2(id_i,T_i,pk_i^1,pk_i^2)})^r(pk_i^2)^r\\&=(pk_i^1T_ig_2^{H_2(id_i,T_i,pk_i^1,pk_i^2)})^r(pk_i^2)^r\\&=N_i,\end{aligned}$$

$$\begin{aligned}U_1^{(a+\omega_i c)H(id_i)+\omega_i u_i}U_2^{b+\omega_i d} &= g^{r((a+\omega_i c)H(id_i))}g^{r\omega_i u_i}g_1^{r(b+\omega_i d)}\\ &= g^{raH(id_i)}g^{rc\omega_i H(id_i)}g^{r\omega_i u_i}g_1^{rb}g_1^{r\omega_i d}\\ &= (g^{aH(id_i)}g_1^b)^r(g^{cH(id_i)}g_1^d g^{t_i+\alpha H_2(id_i,T_i,pk_i^1,pk_i^2)})^{r\omega_i}\\ &= (pk_i^1)^r(pk_i^2 g^{t_i}g_2^{H_2(id_i,T_i,pk_i^1,pk_i^2)})^{r\omega_i}\\ &= (pk_i^1)^r(pk_i^2 T_i g_2^{H_2(id_i,pk_i^1,pk_i^2)})^{r\omega_i}\\ &= V_i.\end{aligned}$$

4.3 安全性分析

下面分别证明本文实例在敌手$\mathcal{F}^1$和$\mathcal{F}^2$的攻击下是 CCA 安全的, 其中, 定理 2 表明在$\mathcal{F}^1$的选择密文攻击下, 我们的 CB-BKEM 构造是 CCA 安全的; 定理 3 表明在$\mathcal{F}^2$的选择密文攻击下, 我们的 CB-BKEM 构造是 CCA 安全的. 特别地, 当条件 $xy=z$ 成立时, 则称(g,g^x,g^y,g^z)是 DDH 元组; 否则, 称其为非 DH 元组.

定理 2. 在多项式时间内进行了 Q_1 次秘密钥生成询问、Q_2 次证书生成询问后, 对于泄露参数$\lambda\leqslant\log q-l_k-\omega(\log\kappa)$, 若$\mathcal{F}^1$在实验 $Exp_{CB\text{-}BKEM,\mathcal{F}^1}^{CCA}(\kappa,\lambda)$ 中获胜的优势是 $Adv_{CB\text{-}BKEM,\mathcal{F}^1}^{CCA}(\kappa,\lambda)$, 那么算法$\mathcal{B}$解决 DDH 困难问题的优势是 $Adv_{\mathcal{B}}^{DDH}(\kappa)\geqslant\left(\frac{n}{Q_1+Q_2+n}\right)Adv_{CB\text{-}BKEM,\mathcal{F}^1}^{CCA}(\kappa,\lambda)$, 其中, n 是广播用户数.

证明: 作为解决 DDH 困难问题的敌手, 挑战者$\mathcal{B}$收到 DDH 困难问题的公开参数(p,g,G)和相应的挑战元组(g,g^x,g^y,T), 其中, $T=g^{xy}$ 或 $T\leftarrow_R G$ (也就是 $T=g^z$, 其中, $z\leftarrow_R Z_p^*$), 并通过与恶意用户$\mathcal{F}^1$间的下述消息交互解决 DDH 困难问题, 即: 当 $T=g^{xy}$ 时$\mathcal{B}$输出 1, 表示(g,g^x,g^y,T)是 DH 元组; 否则输出 0, 表示(g,g^x,g^y,T)是非 DH 元组. 此外, $\mathcal{B}$维护一个列表 L 用于记录实验执行过程中敌手$\mathcal{F}^1$提交询问的相应应答结果, 结构为(id,sk_{id},pk_{id}), 且该列表初始为空.

(1) 初始化. 该阶段, 算法$\mathcal{B}$进行下述操作.

选取 4 个密码学哈希函数 $H:G\to\{0,1\}^{l_k}$, $H_1:\mathcal{ID}\to Z_p^*$, $H_2:\mathcal{ID}\times G\times G\to Z_p^*$和$H_3:\{0,1\}^*\to Z_p^*$, 其中, $\mathcal{ID}$是用户身份空间; 令 $Ext:G\times\{0,1\}^{l_t}\to\{0,1\}^{l_k}$ 是平均情况的$(\log q-\lambda,\varepsilon_1)$强随机性提取器; 令 $g_1\leftarrow g^x$, 随机选取 $\alpha\leftarrow_R Z_p^*$, 计算 $g_2=g^\alpha$. 最后, 秘密保存 $msk=\alpha$, 输出 $Params=\{G,g,g_1,g_2,H,H_1,H_2,H_3,Ext\}$给$\mathcal{F}^1$.

(2) 阶段 1. 敌手$\mathcal{F}^1$将对下述询问进行适应性的询问.

① 公开钥生成询问. 当算法$\mathcal{B}$收到敌手$\mathcal{F}^1$提出的公开钥生成询问$(id,public\ key\ extraction)$时, 若有$(id,sk_{id},pk_{id})\in L$, 则返回 pk_{id} 给$\mathcal{F}^1$; 否则, $\mathcal{B}$从 Z_p^* 中选取 4 个随机数 a、b、c 和 d, 计算 $pk_{id}^1=g^{aH_1(id)}g_1^b$和$pk_{id}^2=g^{cH_1(id)}g_1^d$, 并返回 $pk_{id}=(pk_{id}^1,pk_{id}^2)$ 给$\mathcal{F}^1$, 同时, 在 L 中添加(id,sk_{id},pk_{id}), 其中, $sk_{id}=(a,b,c,d)$;

② 秘密钥生成询问. 当算法$\mathcal{B}$收到敌手$\mathcal{F}^1$提出的秘密钥生成询问$(id,private\ key\ extraction)$时, 若$(id,sk_{id},pk_{id})\in L$, 则返回 sk_{id} 给$\mathcal{F}^1$; 否则, 对 id 进行公开钥生成询问后, 返回 L 中相应(id,sk_{id},pk_{id})的 sk_{id} 给$\mathcal{F}^1$;

③ 证书生成询问. 当算法$\mathcal{B}$收到敌手$\mathcal{F}^1$提出的证书生成询问$(id,pk_{id},certificate\ generation)$时, $\mathcal{B}$随机选取 $t_{id}\leftarrow_R Z_p^*$, 计算相应的证书 $Cert_{id}=\{T_{id},u_{id}\}=(g^{t_{id}},\alpha H_2(id,T_{id},pk_{id}))$, 并返回给敌手$\mathcal{F}^1$. 其中, $\mathcal{B}$通过执行相应的公开钥生成询问获得 id 所对应的公开钥 pk_{id}. 特别地, 敌手$\mathcal{B}$掌握系统主密钥α;

④ 公开钥替换询问. 敌手$\mathcal{F}^1$能够将任意身份 id 的公开钥 pk_{id} 替换为其所选择的内容 pk'_{id};

⑤ 泄露询问. 当算法$\mathcal{B}$收到敌手$\mathcal{F}^1$提出的泄露询问 $(id,f_i:\{0,1\}^*\to\{0,1\}^{\lambda_i},leakage)$时, 其中, $f_i:\{0,1\}^*\to\{0,1\}^{\lambda_i}$ 是高效可计算的泄露函数, 若$(id,sk_{id},pk_{id})\in L$, 则返回 $f_i(sk_{id})$给$\mathcal{F}^1$; 否则, 对 id 进行公开钥生成询问后, 返回 L 中相应记录(id,sk_{id},pk_{id})的 $f_i(sk_{id})$给$\mathcal{F}^1$. 特别地, 在整个生命周期中, $\mathcal{F}^1$获得的关于同一秘密钥 sk_{id} 的泄露总量不能超过参数λ;

⑥ 解封装询问. 当算法$\mathcal{B}$收到敌手$\mathcal{F}^1$提交的关于$(C,\boldsymbol{ID})$的解封装询问时, $\mathcal{B}$对集合 $\boldsymbol{ID}$ 中的 id_i 进行秘密钥生成询问和证书生成询问, 获得相应的 sk_{id} 和 $Cert_{id}$; 然后运行 $k\leftarrow Decap(C,sk_{id},Cert_{id})$, 并输出解封装结果 k.

(3) 挑战. $\mathcal{F}^1$ 提交挑战身份集合 $\boldsymbol{ID}^*=\{id_1,\ldots,id_n\}$给$\mathcal{B}$, 其中, 对任意的身份 $id_i\in\boldsymbol{ID}^*$未提交秘密钥生成询问和证书生成询问. 算法$\mathcal{B}$进行下述主要操作.

① 令 $U_1=g^y$ (隐含地设置 $r=y$)和 $U_2=T$;

② 对 $id_i\in\boldsymbol{ID}^*$, 运行相应的密钥生成算法产生对应的 sk_{id}、pk_{id} 和 $Cert_{id}$, 其中, $sk_{id}=(a,b,c,d)$, $pk_{id}=(pk_{id}^1,pk_{id}^2)=(g^{aH_1(id)}g_1^b,g^{cH_1(id)}g_1^d)$, $Cert_i=\{T_i,u_i\}=\{g^{t_i},t_i+\alpha H_2(id,T_i,pk_i)\}$;

③ 选取随机数 $\eta\leftarrow_R Z_p^*$, 计算 $k_\beta=H(U_2^\eta)$;

④ 选取随机字符串 $S\leftarrow_R\{0,1\}^{l_t}$, 对于 $i=1,\ldots,n$, 计算:

$$N_i=(pk_i^2)^r(pk_i^1T_ig_2^{H_2(id_i,T_i,pk_i^1,pk_i^2)})^r,\ V_i=(pk_i^1)^r(pk_i^2T_ig_2^{H_2(id_i,T_i,pk_i^1,pk_i^2)})^{r\omega_i}\text{ 和 }W_i=Ext(N_i,S)\oplus k_\beta,$$

其中, $\omega_i=H_3(id_i,U_1,U_2,W_i,pk_i,S)$;

⑤ 令 $\boldsymbol{W}=\{W_1,W_2,\ldots,W_n\}$和 $\boldsymbol{V}=\{V_1,\ldots,V_n\}$, 输出封装密文 $C=(\boldsymbol{ID},U_1,U_2,\boldsymbol{W},\boldsymbol{V},S)$及相应的封装密钥 k_β.

当 $T=g^{xy}$ 时, $U_2=g^{xy}=g_1^y$, 则 k_β是与挑战密文 $C=(\boldsymbol{ID},U_1,U_2,\boldsymbol{W},\boldsymbol{V},S)$相对应的封装密钥; 当 $T\leftarrow_R G$ 时, 则 k_β是空间 $\{0,1\}^{l_t}$ 上的随机元素.

(4) 阶段 2. 与阶段 1 相类似, $\mathcal{B}$响应$\mathcal{F}^1$ 提出的相关询问. 但是, $\mathcal{F}^1$ 未对集合 $\boldsymbol{ID}^*=\{id_1,\ldots,id_n\}$中的身份提交证书生成和秘密钥生成等询问. 此外, 该阶段禁止提交泄露询问.

(5) 输出. 敌手$\mathcal{F}^1$ 输出对 k_β的判断. 若$\beta=1$, 敌手$\mathcal{B}$输出 1, 意味着(g,g^x,g^y,T)是一个 DH 元组; 否则, 敌手$\mathcal{B}$输出 0, 意味着(g,g^x,g^y,T)是一个非 DH 元组.

$\mathcal{F}^1$ 提交的秘密钥生成询问的次数是 Q_1, 提交的证书生成询问的次数是 Q_2, 则整个实验中, $\mathcal{F}^1$ 共提交了 Q_1+Q_2+n 个不同的用户身份(挑战身份集合 $\boldsymbol{ID}^*=\{id_1,\ldots,id_n\}$包含 n 个用户身份). 令事件$\mathcal{E}$表示敌手$\mathcal{F}^1$ 未对 $\boldsymbol{ID}^*=\{id_1,\ldots,id_n\}$中的身份提交证书生成和秘密钥生成等询问, 则有 $\Pr[\mathcal{E}]=\dfrac{n}{Q_1+Q_2+n}$.

由强随机性提取器 Ext 的安全性可知, 泄露参数需满足关系$\lambda\leqslant\log q-l_k-\omega(\log\kappa)$, 其中, $\omega(\log\kappa)$表示计算过程中敌手获得的额外泄露.

若敌手$\mathcal{F}^1$ 攻破本文实例 CCA 安全性的优势为 $Adv^{CCA}_{CB\text{-}BKEM,\mathcal{F}^1}(\kappa,\lambda)$, 并且$\mathcal{F}^1$ 未对 $\boldsymbol{ID}^*=\{id_1,\ldots,id_n\}$中的身份提交证书生成和秘密钥生成等询问, 那么算法$\mathcal{B}$解决 DDH 困难问题的优势是

$$Adv^{DDH}_{\mathcal{B}}(\kappa)\geqslant\left(\frac{n}{Q_1+Q_2+n}\right)Adv^{CCA}_{CB\text{-}BKEM,\mathcal{F}^1}(\kappa,\lambda).\qquad\square$$

定理 3. 在多项式时间内进行了 Q_1 次秘密钥生成询问后, 对于泄露参数$\lambda\leqslant\log q-l_k-\omega(\log\kappa)$, 若$\mathcal{F}^2$ 在实验 $Exp^{CCA}_{CB\text{-}BKEM,\mathcal{F}^2}(\kappa,\lambda)$ 中获胜的优势是 $Adv^{CCA}_{CB\text{-}BKEM,\mathcal{F}^2}(\kappa,\lambda)$, 那么算法$\mathcal{B}$解决 DDH 困难问题的优势是 $Adv^{DDH}_{\mathcal{B}}(\kappa)\geqslant\left(\dfrac{n}{Q_1+n}\right)Adv^{CCA}_{CB\text{-}BKEM,\mathcal{F}^2}(\kappa,\lambda)$, 其中, n 是广播用户数.

在定理 2 中, DDH 困难问题挑战元组(g,g^x,g^y,T)的元素并未嵌在 msk 中, 因此在定理 2 的证明过程中, 算法$\mathcal{B}$持有完整的 msk (即$\mathcal{B}$具备向敌手提供主私钥的能力), 那么可以使用定理 1 的证明思路对定理 3 进行证明. 此外, 在定理 3 的证明中, $\mathcal{F}^2$ 未对 $\boldsymbol{ID}^*=\{id_1,\ldots,id_n\}$中的身份提交秘密钥生成询问的概率为 $\dfrac{n}{Q_1+n}$. 综上所述, 若$\mathcal{F}^2$ 攻破本文实例 CCA 安全性的优势是 $Adv^{CCA}_{CB\text{-}BKEM,\mathcal{F}^2}(\kappa,\lambda)$, 且$\mathcal{F}^2$ 对 $\boldsymbol{ID}^*=\{id_1,\ldots,id_n\}$中的任何身份未提交秘密钥生成询问, 那么算法$\mathcal{B}$解决 DDH 困难问题的优势为 $Adv^{DDH}_{\mathcal{B}}(\kappa)\geqslant\left(\dfrac{n}{Q_1+n}\right)Adv^{CCA}_{CB\text{-}BKEM,\mathcal{F}^2}(\kappa,\lambda)$.

5 抗连续泄露攻击的广播密钥封装机制

事实上, 密码机制在实际应用时将面临连续的泄露攻击, 因此密码原语的连续泄露容忍性更加接近现实

应用环境的实际需求. 文献[21]指出, 当一个抗泄露的密码机制同时满足: (1) 密钥更新算法执行前后, 密码机制的安全性和性能保持不变; (2) 两次密钥更新算法的执行间隔内秘密信息的泄露量是有界的等条件时, 那么该机制能够通过定期执行密钥更新算法实现密码原语的连续泄露容忍性. 由上述结论可知: 实现密码机制抵抗连续泄露攻击的实质是为抵抗有界泄露攻击的密码原语提供密钥更新算法; 并且该算法执行前后, 密码原语在保持原有安全性和功能的同时, 相应的公开参数是不变的. 因此, 在抵抗有界泄露攻击的密码原语中, 若秘密钥和公开参数间存在多对一关系, 那么该机制能够通过密钥更新的方法实现连续泄露容忍性, 否则无法实现. 在本文基础的 CB-BKEM 构造中, 公开钥 $pk_{id}=(pk_{id}^1, pk_{id}^2)$ 保持不变时对应的秘密钥 $sk_{id}=(a,b,c,d)$同样是不变的, 因此上述公秘密钥的结构关系无法实现抵抗连续泄露攻击的目的.

本节将(a,b,c,d)视为 CB-BKEM 构造的底层核心秘密, (a,b,c,d)与用户公开钥 $pk_{id}=(pk_{id}^1, pk_{id}^2)$ 间是一对一的映射关系, 但用户秘密钥 $sk_{id}=(\boldsymbol{t}_1,\boldsymbol{t}_2)$与$(a,b,c,d)$间是多对一的映射关系, 此时密钥更新算法能够实现用户秘密钥空间中各元素间的变换, 但它们对应的底层核心秘密(a,b,c,d)是不发生变化的, 确保公开钥在用户秘密钥更新前后是不发生变化的. 综上所述, 密钥更新算法的底层核心思想如图 2 所示, 以底层核心秘密信息为桥梁, 使得用户秘密钥sk_{id}与公开钥pk_{id}间存在隐含的多对一的映射关系. 文献[1]采用矩阵的形式在用户秘密钥与用户公开钥间建立了多对一的映射关系, 然而大量矩阵的乘法运算导致密钥更新算法的执行效率较低; 而本文的密钥更新算法仅需进行向量的加减运算, 大幅度提升了密钥更新操作的计算效率.

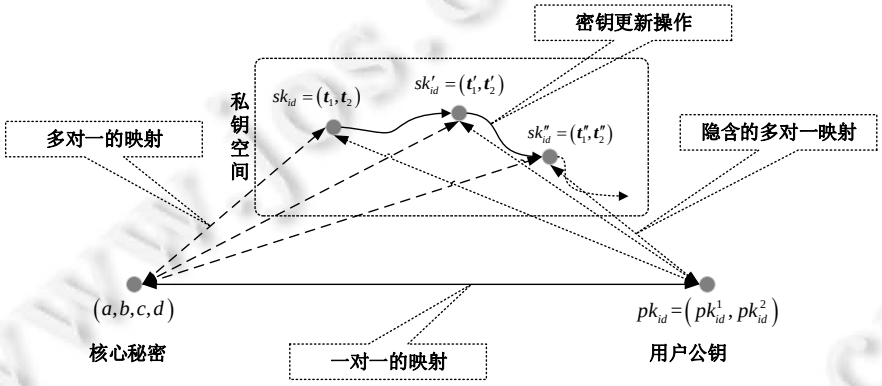

图 2 抗连续泄露 CB-BKEM 中密钥更新算法的底层核心思想

为了进一步提高 CB-BKEM 的实用性, 本节将为 CB-BKEM 提供抵抗连续泄露攻击的能力, 在上述构造的基础上提出抵抗连续泄露攻击的 CB-BKEM 实例$\Pi'=(Setup', KeyGen', Encap', Decap', Update')$, 其中, 初始化算法 $Setup'$、证书生成算法 $CertGen'$和封装算法 $Encap'$与第 4 节中原始机制Π的相关算法保持一致. 特别地, $Update$ 是密钥更新算法, 对用户秘密钥进行周期性更新, 使得更新前后的用户秘密钥对于任意敌手而言都是不可区分的. 密钥更新算法的具体定义如下所示.

$sk'_{id}\leftarrow Update(sk_{id})$. 密钥更新算法输入原始的用户秘密钥 sk_{id}, 输出满足条件$|sk_{id}|=|sk'_{id}|$和 $SD(sk_{id}, sk'_{id})\leqslant negl(\kappa)$ 的更新秘密钥 sk'_{id}, 其中, $|sk_{id}|$表示 sk_{id} 的长度.

5.1 具体构造

(1) 密钥生成算法$(pk_{id},sk_{id})\leftarrow KeyGen'(id)$的主要操作包括:

随机选取向量 $\boldsymbol{t}_1\leftarrow_R (Z_p^*)^4$和$\boldsymbol{t}_2\leftarrow_R (Z_p^*)^4$, 计算:

$$(a,b,c,d)=\boldsymbol{t}_1+\boldsymbol{t}_2,\ pk_{id}^1=g^{aH_1(id)}g_1^b和pk_{id}^2=g^{cH_1(id)}g_1^d.$$

输出用户 U_{id} 的公私钥对(sk_{id},pk_{id}), 其中, $sk_{id}=(\boldsymbol{t}_1,\boldsymbol{t}_2)$, $pk_{id}=(pk_{id}^1, pk_{id}^2)$.

(2) 密钥更新算法 $sk'_{id}\leftarrow Update'(sk_{id})$ 的主要操作包括:

选取随机向量 $\boldsymbol{\omega} \leftarrow_R (Z_p^*)^4$，并计算 $\boldsymbol{t}_1' = \boldsymbol{t}_1 + \boldsymbol{\omega}$和$\boldsymbol{t}_2' = \boldsymbol{t}_2 - \boldsymbol{\omega}$.

输出更新后的秘密钥 $sk_{id} = (\boldsymbol{t}_1', \boldsymbol{t}_2')$.

密钥更新算法执行前后，更新的秘密钥 sk_{id}' 和原始秘密钥 sk_{id} 所对应的底层核心秘密(a,b,c,d)是不变的，确保密钥更新前后公开参数和性能是不变的；由于 $\boldsymbol{\omega} \leftarrow_R (Z_p^*)^4$，那么 sk_{id}' 和 sk_{id} 是不可区分的；密钥更新算法使用新的随机数生成了新的秘密钥 sk_{id}'，使得之前关于 sk_{id} 的泄露信息对 sk_{id}' 是无用的，对敌手而言，关于秘密钥泄露信息的量需要从 0 开始记录，因此在密钥更新算法执行间隔内泄露是有界的，然而在整个生命周期内泄露是无界的.

(3) 解封装算法 $k \leftarrow Decap''(C, sk_{id}, Cert_{id})$的主要操作包括:

① 计算$(a,b,c,d)=\boldsymbol{t}_1+\boldsymbol{t}_2$;

② 若等式 $V_i = U_1^{(a+\eta_i c)H(id_i)+\omega_i u_i} U_2^{b+\eta_i d}$ (其中，$\eta_i=H_3(id_i,U_1,U_2,W_i,S)$)成立，计算 $N_i' = U_1^{((a+c)H(id_i)+u_i)} U_2^{b+d}$，则输出封装密钥 $k = Ext(N_i', S) \oplus W_i$；否则，输出特殊的符号$\perp$.

5.2 正确性及安全性分析

连续泄露的最大优势是，可通过定期执行密钥更新算法将连续泄露的问题转化为有界泄露容忍性进行研究. 本文抵抗连续泄露攻击的实例Π'是在底层基础机制Π的基础之上设计的，因此该机制的正确性和安全性可由底层 CB-BKEM 的相应性质获得；而该机制的连续泄露容忍性可由密钥更新算法和底层 CB-BKEM 的有界泄露容忍性获得.

5.3 性能及效率分析

在同时为 n 个用户生成封装密钥的前提条件下，表 1 所示为我们的构造Π'与相应 CB-KEM[16,22,23]和无证书密钥封装机制[24]在计算效率、通信效率和性能等方面的比较结果. 在计算效率方面，本文构造并未使用计算量较大的双线性映射运算，而现有的 CB-KEM[16]是基于双线性运算构造的；在通信效率方面，本文的 CB-BKEM 具有广播通信的功能，由于密文中包含了用户的身份列表，因此导致密文的长度较长，但保持了文献[16]在公开钥和公开参数等方面的优势；在性能方面，文献[22,23]不具备抵抗泄露攻击和广播通信的功能，文献[16,24]仅具有抵抗泄露攻击的能力，而无法实现广播通信；然而，我们的方案具有泄露容忍性和广播通信的功能，相较于文献[16,22–24]而言，本文构造的性能更优.

表 1 与现有 CB-KEM 实例的对比

相关方案	计算效率		通信效率			机制性能	
	封装算法	解封装算法	密文	公开钥	公开参数	泄露容忍	广播通信
文献[16]	$3nE_\omega+nE_e$	$4E_e$	$2n$	1	3	✓	×
文献[22]	$5nE_\omega+2nE_e$	$3E_\omega+1E_e$	$2n$	3	5	×	×
文献[23]	$6nE_\omega+3nE_e$	$4E_\omega+4E_e$	$3n$	4	$2l_{id}+5$	×	×
文献[25]	$6nE_a$	$6E_a$	$4n$	2	3	✓	×
本文构造Π'	$(4n+3)E_\omega$	$4E_\omega$	$2n+3$	2	3	✓	✓

表 1 中: E_e 表示双线性映射运算, E_ω表示群上的指数运算, E_a 表示群上的数乘运算，并且相应的系数表示该运算的次数；✓表示机制具有相应的性能，×表示不具备；n 表示广播通信的用户数，l_{id} 表示用户的身份长度. 此外，表中通信效率主要统计相关比较项目的元素个数；计算效率的比较中未记录哈希函数和异或运算等.

本文在个人计算机(CPU: Intel(R) Core i5-4200 H；主频: 2.8 GHz；内存: 2 GB；硬盘: 128 GB 固态；操作系统: Ubuntu 18.04 64 位; PBC 算法库的版本号: PBC-0.5.14)上对基础密码操作的运行时间进行了测算，对双线性映射操作和群上的指数运算通过求 20 次测量的平均值得到相应的运行时间，即 E_e=1.365 (ms)和E_ω=1.112 (ms).

本文在图 3 中分别展示了当 n=10、n=100 和 n=1000 时，本文构造Π'与相应 CB-KEM[16,22,23]的计算效率的比较结果. 图 4 所示为 n=10 时，本文构造与相应 CB-KEM[16,22,23]的封装和解封装算法的效率比较结果. 随着广播通信用户数量的增多，本文的 CB-BKEM 计算效率优势更加明显，因此与传统机制相比，本文方案在

计算效率和安全性等方面的优势使其能够更好地满足多用户环境下隐私数据的加密需求, 如云计算下的数据访问授权等.

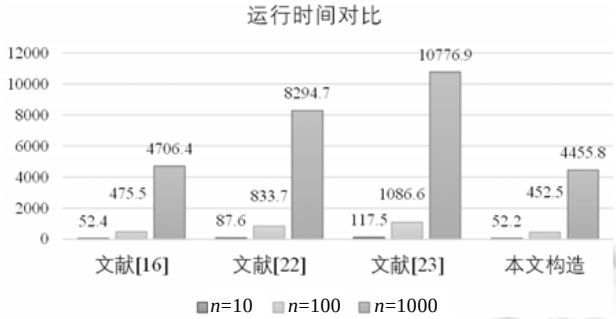

图 3　本文构造与相应 CB-KEM 的计算效率比较结果(单位: ms)

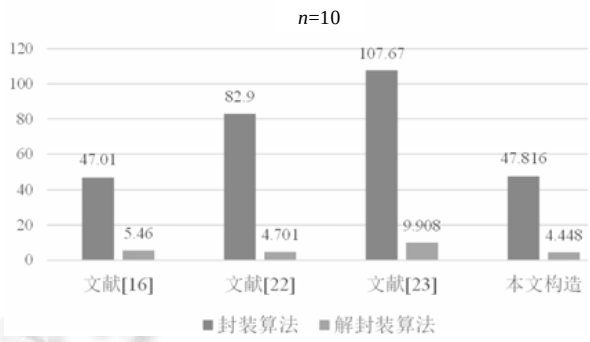

图 4　本文构造与相应 CB-KEM 的封装和解封装算法的效率比较结果(n=10, 单位: ms)

6　应用探讨

随着云计算等网络存储技术的普及, 云存储数据的安全性受到了用户的广泛关注. 为保障远程数据存储的安全性, 数据拥有者通常对数据加密后进行存储; 然而, 实际应用时一份数据往往需要多个用户共享, 但为每个用户分别产生授权的传统方法将导致数据的使用效率较低, 因此需要具备广播通信功能的数据授权机制, 实现云计算中密文数据的高效授权访问功能.

在讨论实际应用之前, 为进一步增强 CB-BKEM 的实用性, 对本文实例Π'的密钥封装和解封装算法分别进行改进, 使其为不用的消息用户生成不同的封装密钥.

(1) 密钥封装算法$(C,\boldsymbol{K})\leftarrow Encap''(\boldsymbol{ID},\boldsymbol{PK_{ID}})$

① 选取随机数 $r\leftarrow_R Z_p^*$, 计算 $U_1=g^r$ 和 $U_2=g_1^r$;

② 随机选取 $S\leftarrow_R\{0,1\}^{l_t}$, 对接收者集合 $\boldsymbol{ID}=\{id_1,\ldots,id_n\}$ (其中, $\boldsymbol{PK_{ID}}=\{pk_1,\ldots,pk_n\}$)中的每个身份 id_i 计算:

$$N_i=(pk_i^2)^r(pk_i^1T_ig_2^{H_2(id_i,T_i,pk_i^1,pk_i^2)})^r,\ V_i=(pk_i^1)^r(pk_i^2T_ig_2^{H_2(id_i,T_i,pk_i^1,pk_i^2)})^{r\eta_i}\text{ 和}k_i=Ext(N_i,S),$$

其中, $\eta_i=H_3(id_i,U_1,U_2,W_i,S)$;

③ 令 $\boldsymbol{V}=\{V_1,\ldots,V_n\}$, 输出封装密文 $C=(ID,U_1,U_2,\boldsymbol{V},S)$及相对应的封装密钥集合 $\boldsymbol{K}=\{k_1,\ldots,k_n\}$, 其中, 对接收者 id_i 产生的封装密钥为 k_i.

(2) 解封装算法 $k_i\leftarrow Decap''(C,sk_{id_i},Cert_{id_i})$ 的主要操作包括:

① 计算$(a,b,c,d)=\boldsymbol{t}_1+\boldsymbol{t}_2$;

② 若等式 $V_i=U_1^{(a+\eta_ic)H(id_i)+\omega_iu_i}U_2^{b+\eta_id}$ (其中, $\eta_i=H_3(id_i,U_1,U_2,W_i,S)$)成立, 计算 $N_i'=U_1^{((a+c)H(id_i)+u_i)}U_2^{b+d}$, 则输出封装密钥 $k_i=Ext(N_i,S)$; 否则, 输出特殊的符号$\perp$.

本节以云数据的访问控制机制为例，探讨 CB-BKEM 实例的应用前景. 在云计算环境下，数据拥有者需要将在云环境中存储数据的使用权授权给使用者，针对该实际需求，如图 5 所示，我们基于 CB-BKEM 设计了一个云数据的抗泄露广播授权机制，具体过程简要叙述如下.

(1) 云存储中心的 KGC 服务器运行初始化算法建立 CB-BKEM 的系统环境，各系统用户 U_{id} 从 CA 处获得公开参数 $Params$ 及各自的证书 $Cert_{id}$;

(2) 数据拥有者欲向集合 $\boldsymbol{ID}=\{id_1,\ldots,id_n\}$中的用户授权对数据 $\boldsymbol{M}=\{M_1,\ldots,M_n\}$的使用权限，首先，运行 CB-BKEM 改进的广播封装算法$(C,\boldsymbol{K}=\{k_1,\ldots,k_n\})\leftarrow Encap''(\boldsymbol{ID},\boldsymbol{PK_{ID}})$; 然后，用封装密钥 k_i 将数据 M_i 用对称加密算法 $Data_i=Dem.Enc(k_i,M_i)$加密后上传存储至云存储中心; 最后，将封装密文 C 广播给集合 $\boldsymbol{ID}=\{id_1,\ldots,id_n\}$中的每一个数据使用者 id_i. 通过上述运算，数据拥有者将授权用户 id_i 仅可以使用数据 M_i;

(3) 数据使用者 id_i 收到数据拥有者的广播密文 C 后，首先运行 CB-BKEM 的解封装算法 $k\leftarrow Decap''(C,sk_{id},Cert_{id})$; 然后，对从云存储中心下载的密态数据 $Data_i$ 使用封装密钥 k_i 对其进行解密 $M_i=Dem.Enc(k,Data_i)$, 即可获得授权数据 M_i. 特别地，本节仅仅是简要的描述，实际应用中还需加入更多的访问属性以实现细粒度的数据授权.

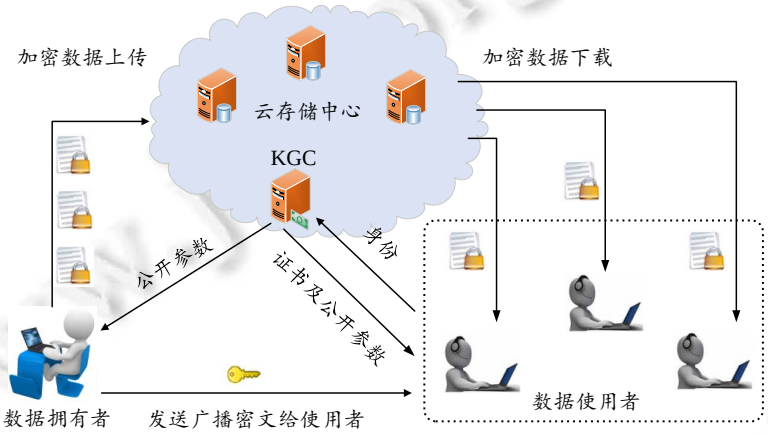

图 5 云数据的抗泄露广播授权机制

7 结束语

本文针对现实环境中广播通信的抗泄露应用需求，同时为充分发挥对称加密高计算效率的优势，提出了具有 CCA 安全的抗泄露 CB-BKEM，并基于经典的 DDH 困难性假设对本文构造的 CCA 安全性进行了证明. 此外，本文以云计算环境为例，对 CB-BKEM 实例的具体应用进行了探索，提出了云数据的抗泄露广播授权机制，并对协议的执行过程进行了简要描述.

本文实例中授权接收者身份列表的使用，方便了接收者确认通信密文的参数，但增加了密文的长度. 下一阶段，我们将结合当前抗泄露密码机制的研究成果[1−6,24−28]，研究匿名的 CB-BKEM，取消身份列表的传输，进一步缩短封装密文的长度.

References:

[1] Zhou YW, Yang B, Zhang WZ, Mu Y. CCA2 secure public-key encryption scheme tolerating continual leakage attacks. Security and Communication Networks, 2016, 9(17): 4505−4519. [doi: 10.1002/sec.1643]

[2] Zhou YW, Yang B. Continuous leakage-resilient public-key encryption scheme with CCA security. The Computer Journal, 2017, 60(8): 1161−1172. [doi: 10.1093/comjnl/bxw110]

[3] Li JG, Teng ML, Zhang YC, Yu QH. A leakage-resilient CCA-secure identity-based encryption scheme. The Computer Journal, 2016, 59(7): 1066−1075. [doi: 10.1093/comjnl/bxv128]

[4] Li JG, Teng ML, Zhang YC. Identity-based broadcast encryption with continuous leakage resilience. Information Sciences, 2018, 429(3):177−193. [doi: 10.1016/j.ins.2017.11.008]

[5] Li JG, Guo YY, Yu QH, Lu Y, Zhang YC. Provably secure identity-based encryption resilient to post-challenge continuous auxiliary inputs leakage. Security and Communication Networks, 2016, 9(10): 1016−1024. [doi: 10.1002/sec.1396]

[6] Li JG, Yu QH, Zhang YC. Hierarchical attribute-based encryption with continuous leakage resilience. Information Sciences, 2019, 484: 113−134. [doi: 10.1016/j.ins.2019.01.052]

[7] Li JG, Yu QH, Zhang YC. Key-policy attribute-based encryption against continual auxiliary input leakage. Information Sciences, 2019, 470: 175−188. [doi: 10.1016/j.ins.2018.07.077]

[8] Zhou YW, Yang B. Leakage-resilient CCA2-secure certificateless public-key encryption scheme without bilinear pairing. Information Processing Letters, 2018, 130: 16−24. [doi: 10.1016/j.ipl.2017.09.012]

[9] Zhou YW, Yang B. Continuous leakage-resilient certificateless public key encryption with CCA security. Knowledge-based Systems, 2017, 136: 27−36. [doi: 10.1016/j.knosys.2017.08.019]

[10] Wu JD, Tseng YM, Huang SS. An identity-based authenticated key exchange protocol resilient to continuous key leakage. IEEE Systems Journal, 2019, 13(4): 3968−3979. [doi: 10.1109/JSYST.2019.2896132]

[11] Yu QH, Li JG, Zhang YC. Leakage-resilient certificate-based encryption. Security and Communication Networks, 2015, 8(18): 3346−3355. [doi: 10.1002/sec.1258]

[12] Li JG, Guo YY, Yu QH, Lu Y, Zhang YC, Zhang YC, Zhang FT. Continuous leakage-resilient certificate-based encryption. Information Sciences, 2016, 355−356: 1−14. [doi: 10.1016/j.ins.2016.03.032]

[13] Tseng YM, Wu JD, Hung RW, Chien HY. Leakage-resilient certificate-based encryption scheme for iot environments. In: Proc. of the 9th Int'l Conf. on Awareness Science and Technology. Fukuoka: IEEE, 2018. 251−256. [doi: 10.1109/ICAwST.2018.8517196]

[14] Zhou YW, Yang B, Wang T, Xia Z, Hou HX. Continuous leakage-resilient certificate-based encryption scheme without bilinear pairings. The Computer Journal, 2020, 63(4): 508−524. [doi: 10.1093/comjnl/bxz085]

[15] Lu Y, Li JG. Efficient constructions of certificate-based key encapsulation mechanism. Int'l Journal of Internet Protocol Technology, 2014, 8(2/3): 96−106. [doi: 10.1504/IJIPT.2014.066374]

[16] Wu JD, Tseng YM, Huang SS, Tsai TT. Leakage-resilient certificate-based key encapsulation scheme resistant to continual leakage. IEEE Open Journal of the Computer Society, 2020, 1: 131−144. [doi: 10.1109/OJCS.2020.3008961]

[17] Chen LQ, Li JG, Lu Y, Zhang YC. Adaptively secure certificate-based broadcast encryption and its application to cloud storage service. Information Sciences, 2020, 538: 273−289. [doi: 10.1016/j.ins.2020.05.092]

[18] Chen LQ, Li JG, Zhang YC. Anonymous certificate-based broadcast encryption with personalized messages. IEEE Trans. on Broadcasting, 2020, 66(4): 867−881. [doi: 10.1109/TBC.2020.2984974]

[19] Li JG, Chen LQ, Lu Y, Zhang YC. Anonymous certificate-based broadcast encryption with constant decryption cost. Information Sciences, 2018, 454−455: 110−127. [doi: 10.1016/j.ins.2018.04.067]

[20] Chen LQ, Li JG, Zhang YC. Adaptively secure efficient broadcast encryption with constant-size secret key and ciphertext. Soft Computing, 2020, 24(6): 4589−4606. [doi: 10.1007/s00500-019-04219-5]

[21] Zhou YW, Yang B, Qiao ZR, Xia Z, Zhang MW. Leakage-resilient certificateless key-encapsulation mechanism and application. SCIENTIA SINICA Informationis, 2021, 51(12): 2119−2133 (in Chinese with English abstract).

[22] Dodis Y, Haralambiev K, López-Alt A, Wichs D. Cryptography against continuous memory attacks. In: Proc. of the FOCS 2010. 2010. 511−520.

[23] Li JG, Yang HS, Zhang YC. Certificate-based key encapsulation mechanism with tags. Ruan Jian Xue Bao/Journal of Software, 2012, 23(8): 233−242 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4127.htm [doi: 10.3724/SP.J.1001.2012.04127]

[24] Zhou YW. Research on the (continuous) leakage resilience of several cryptographic primitives [Ph.D. Thesis]. Xi'an: Shaanxi Normal University, 2018 (in Chinese with English abstract).

[25] Lu Y, Li JG. Efficient and provably secure certificate-based key encapsulation mechanism in the standard model. Journal of Computer Research and Development, 2014(7): 1497−1505 (in Chinese with English abstract).

[26] Hou HX. Research on provably secure identity-based public-key cryptosystems under the key leakage attacks [Ph.D. Thesis]. Xi'an: Shaanxi Normal University, 2020 (in Chinese with English abstract).

[27] Zhou YW, Hu BJ, Yang B, Xia Z, Zhang MW. (Hierarchical) Identity-based key encapsulation mechanism with leakage-resilience. Chinese Journal of Computers, 2021, 44(4): 820−835 (in Chinese with English abstract).

[28] Zhou YW, Yang B, Xia Z, Lai QQ, Zhang MW, Mu Y. Revocable identity-based encryption scheme with leakage-resilience. Chinese Journal of Computers, 2020, 43(8): 1534−1554 (in Chinese with English abstract).

附中文参考文献:

[21] 周彦伟, 杨波, 乔子芮, 夏喆, 张明武. 抗泄露的无证书密钥封装机制及应用. 中国科学: 信息科学, 2021, 51(12): 2119−2133.

[23] 李继国, 杨海珊, 张亦辰. 带标签的基于证书密钥封装机制. 软件学报, 2012, 23(8): 233−242. http://www.jos.org.cn/1000-9825/4127.htm [doi: 10.3724/SP.J.1001.2012.04127]

[24] 周彦伟. 几类密码学基础原语的抗(连续)泄露性研究 [博士学位论文]. 西安: 陕西师范大学, 2018.

[25] 陆阳, 李继国. 标准模型下高效安全的基于证书密钥封装机制. 计算机研究与发展, 2014(7): 1497−1505.

[26] 侯红霞. 密钥泄露攻击下可证明安全的身份基公钥密码方案研究 [博士学位论文]. 西安: 陕西师范大学, 2020.

[27] 周彦伟, 杨波, 胡冰洁, 夏喆, 张明武. 抗泄露的(分层)身份基密钥封装机制. 计算机学报, 2021, 44(4): 820−835.

[28] 周彦伟, 杨波, 夏喆, 来齐齐, 张明武, 穆怡. 抵抗泄露攻击的可撤销 IBE 机制. 计算机学报, 2020, 43(8): 1534−1554.

乔子芮(1985－), 女, 博士生, 主要研究领域为密码学, 信息安全.

杨波(1973－), 男, 博士, 教授, 博士生导师, 主要研究领域为密码学, 信息安全.

杨启良(1990－), 男, 博士, 主要研究领域为密码学, 信息安全.

夏喆(1982－), 男, 博士, 副教授, 主要研究领域为密码学, 信息安全.

周彦伟(1986－), 男, 博士, 副教授, 主要研究领域为密码学, 匿名通信.

张明武(1973－), 男, 博士, 教授, 博士生导师, 主要研究领域为密码学, 信息安全.