计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2017.20170446
54(10):2232-2243,2017

云计算中基于身份的双服务器密文等值判定协议

吴黎兵[1,2] 张宇波[2] 何德彪[1,2]
[1](软件工程国家重点实验室(武汉大学) 武汉 430072)
[2](武汉大学计算机学院 武汉 430072)
(wu@whu.edu.cn)

Dual Server Identity-Based Encryption with Equality Test for Cloud Computing

Wu Libing[1,2], Zhang Yubo[2], and He Debiao[1,2]
[1](*State Key Laboratory of Software Engineering* (*Wuhan University*), *Wuhan* 430072)
[2](*School of Computer Science*, *Wuhan University*, *Wuhan* 430072)

Abstract With the rapid development of cloud storage and the increasing awareness of privacy, more and more private data are encrypted before outsourcing to the cloud. Thus, how to search in encrypted data has been a new research item in the scope of searchable encryption. One of the solutions is public key encryption with equality test (PKEET). It can check whether the plaintexts of two ciphertexts encrypted under different public keys are the same, without leakage any information about the plaintexts. Recently, many public key encryption schemes with equality test have been proposed. However, in these schemes, there were only one server be used to perform the equality test, which means that they could not withstand the inner keywords guessing attack. To solve this problem, we propose the first dual server identity-based encryption scheme with equality test (DS-IBEET). And we prove the security under random oracle model. In addition, performance evaluation shows that our scheme is suitable for resource-limited mobile devices.

Key words equality test; identity-based encryption; cloud computing; dual server; searchable encryption

摘 要 随着云存储的快速普及以及公众隐私保护意识的提升,越来越多的隐私数据被加密存储在云上.因而,如何对密文数据特别是采用公钥密码体制加密的数据进行高效检索成为了一个重要研究内容.带密文等值判定的公钥加密协议是其中一种检索方法,它可以在不泄漏明文内容的情况下判定2段密文对应的明文是否相同.最近,一系列带密文等值判定的公钥加密协议被提出.然而,在这些协议中,只用了一个服务器来执行等值判定操作,不能抵抗恶意服务器的内部关键字猜测攻击.为了解决这个问题,首次提出了基于双服务器的带密文等值判定的公钥加密协议,并在随机预言机模型下证明了它的安全性.同时,也对设计的协议进行了性能分析,分析表明:该协议适合资源受限的移动设备.

关键词 等值判定;基于身份加密;云计算;双服务器;可搜索加密

中图法分类号 TP391

收稿日期:2017-06-11;**修回日期**:2017-07-28

基金项目:国家自然科学基金项目(61472287);湖北省自然科学基金重点项目(2015CFA068)
This work was supported by the National Natural Science Foundation of China (61472287) and the Key Program of Natural Science Foundation of Hubei Province of China (2015CFA068).

通信作者:何德彪(hedebiao@whu.edu.cn)

近年来,随着云计算的快速发展和普及,越来越多的用户将自己的数据存放在云端(如 Dropbox、亚马逊云、阿里云、百度云等).数据上传到云端后,用户可以通过网络远程操作自己的数据(包括新增、修改、删除、检索).同时,为了保护用户的数据隐私,数据上传到云端之前会进行加密.但这使得数据检索,甚至是在同一用户上传的数据中检索变得困难.解决方法之一是将所有数据下载下来,解密之后再进行检索.显然,此方法在数据量大、网络带宽小、响应时延要求高的环境中不具有可行性.另一种解决方法是使用可搜索加密(searchable encryption)技术.

可搜索加密技术是一种允许第三方对用户上传的密文数据进行检索,同时不会泄漏除搜索模式和搜索结果以外的任何信息的技术.可分为可搜索对称加密和可搜索公钥加密技术.前者适合对用户自己拥有的数据进行加密;后者可以搜索不同用户用不同公钥加密的数据,由 Boneh 等人首次提出[1].随后,很多相关的研究成果陆续被发布[2-11];最近,一些新型的带密文等值判定的可搜索公钥加密方案[12-17]被提出.但是这些方案都是基于单服务器的,容易遭受内部关键字猜测攻击.

为此,我们设计了一个基于双服务器的带密文等值判定的公钥加密协议(dual server identity-based encryption scheme with equality test, DS-IBEET).该协议的典型应用场景如图 1 所示.在一个移动健康社交网络(MHSN)中,用户(比如病人)想要通过医疗服务器和其他拥有相同症状的人建立联系,同时不泄漏他们的隐私信息.例如,Alice 和 Bob 是 2 个有相同症状的病人,他们彼此不认识,但是想要建立联系,以交流病情或相互鼓励.Alice 用

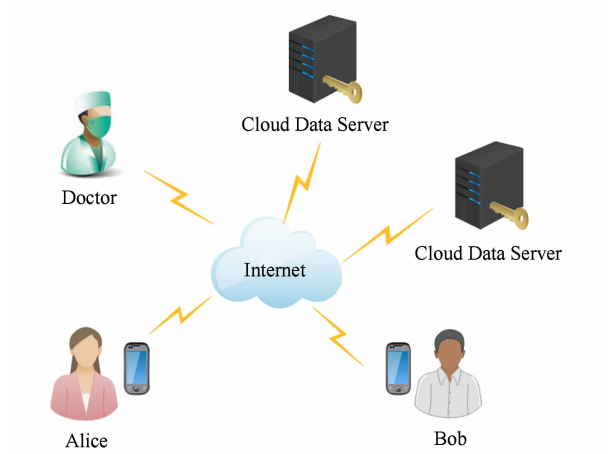

Fig. 1 A typical MHSN scenario

图 1 典型的移动健康社交网络场景

她医生的公钥 ID_{DA} 加密她的症状信息得到(ID_A, $IBEET(ID_{DA}, Symptom)$)生成 $Symptom$ 的陷门 td_A,发送(ID_A, $IBEET(ID_{DA}, Symptom)$, td_A)给服务器 SA.同理,Bob 发送(ID_B, $IBEET(ID_{DB}, Symptom)$, td_B))给服务器 SA.服务器 SA 首先进行初步判定,判定结果再发送给服务器 SB,SB 完成密文等值判定,可以得到 Alice 和 Bob 有相同的症状 $Symptom$,但是服务器 SA 和 SB 都不知道 $Symptom$ 的具体内容是什么.最后,服务器 SB 将判定结果发送给 Alice 和 Bob,让他们建立联系.显然,此协议也适用于多用户环境.

本文的主要贡献有 3 个方面:

1) 首次提出基于双服务器的密文等值判定协议,解决了传统的单服务器环境不能抵抗恶意服务器攻击的问题;

2) 在随机预言机模型下证明了该协议的安全性;

3) 分析了该协议的性能,表明其可以适用于资源受限的移动设备.

1 相关工作

Yang 等人首次提出带密文等值判定功能的公钥加密协议(public key encryption with equality test, PKEET)[13],用来比较 2 段密文对应的明文是否相同,且在比较过程中不泄露任何明文信息.然而,该协议没有对参与密文等值判定的用户进行授权,任何用户都可以进行密文等值判定;为了解决这一问题,Tang 提出了一种改进的带密文等值判定的公钥加密协议(FG-PKEET)[12],该协议仅允许获得授权的用户在可信第三方的帮助下执行细粒度密文等值判定,即用户可以通过授权来控制谁能对自己的密文进行等值判定,控制谁的密文可以和自己的密文进行比对;随后,为了抵抗离线消息恢复攻击,Tang 将 FG-PKEET 扩展到了双代理版本(ADG-PKEET)[15];同时,为了达到粗粒度控制的目的,Tang 提出了 AoN-PKEET 协议[14],在该协议中,用户一旦授权第三方代理进行密文等值判定,则该用户的所有密文都能被该代理用来进行等值判定.

Ma 等人提出了一个带委托密文等值判定的公钥加密协议(PKE-DET)[16],该协议中的等值判定操作被委托给第三方;Huang 等人提出了一个带密文等值判定授权的公钥加密协议[17],该协议的授权方式分为对用户授权和对密文授权;对用户授权模

式下,测试者可以对授权用户的所有密文进行等值判定查询,对密文授权模式下,测试者只能对授权的特定密文进行等值判定查询;随后,Ma 等人又提出了一种支持弹性授权的带密文等值判定的公钥加密方案[18],该方案拥有 4 种类型的授权方式.

上述方案都是基于传统公钥加密的方案,当用户数量较大时,存在证书管理问题. 为了解决这个问题,Ma 首次提出了一种基于身份的带密文等值判定的公钥加密方案[19],随后,在 Ma 的协议基础上,Wu 设计了一个更高效的方案[20],该方案减少了 HashToPoint 的使用.

然而,在已有方案中,不论是传统的公钥加密方案还是基于身份的加密方案,都不能抵抗恶意服务器的内部关键字猜测攻击. 特别是在关键字数量有限的情况下,攻击更容易成功.

2 数学基础

在本节中,我们主要介绍双线性对和 CDH (computational Diffie-Hellman problem)困难问题.

2.1 双线性对

设 G_1 是加法循环群,G_2 是乘法循环群,且 G_1、G_2 的阶都为素数 q. P 是群 G_1 的一个生成元. 如果 $e:G_1\times G_1\rightarrow G_2$ 满足 3 个条件,则称其为双线性对.

1) 双线性性. 对于任意 2 个点 $Q,R\in G_1$ 和任意 2 个随机数 $a,b\in Z_q^*$,存在 $e(a\cdot Q,b\cdot R)=e(Q,R)^{ab}$.

2) 非退化性. 对于任意生成元 $P\in G_1$,$e(P,P)\neq 1_{G_2}$.

3) 可计算性. 对于任意 2 个点 $Q,R\in G_1$,$e(Q,R)$是易计算的.

2.2 CDH 问题

设 P 是群 G_1 的一个生成元,$x,y\in Z^*$ 是 2 个未知随机数,若已知$\{P,xP,yP\}$,则在多项式时间内计算出 xyP 是困难的.

3 模 型

3.1 系统模型

本文的系统模型如图 2 所示. 系统有 5 个参与者:

1) 密钥生成中心. 负责生成系统主密钥、医生和病人的公私钥对以及系统参数,并将医生和病人的私钥通过安全信道分别发送给他们.

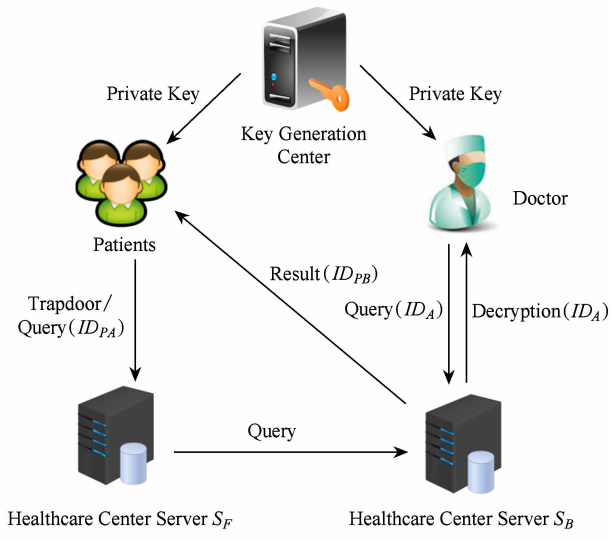

Fig. 2 System model

图 2 系统模型

2) 病人. 病人需要先在密钥生成中心注册,获得私钥. 将隐私信息(例如症状)加密后发送给服务器. 同时,发送一个陷门给服务器,用来执行等值判定操作. 病人加密的信息只有他的医生可以解密.

3) 医生. 医生需要先在密钥生成中心注册,获得私钥. 医生可以解密自己的病人加密的信息.

4) 前服务器 S_F. 负责接收病人的密文等值判定请求,并将获得的中间结果发送给服务器 S_B.

5) 后服务器 S_B. 负责接收服务器 S_F 的中间结果,完成后续密文等值判定工作,并返回最终的结果给病人.

3.2 协议架构

协议包含 7 个算法:

1) 初始化算法 $Setup(1^k)$. 输入安全参数 k,产生系统参数 $params$.

2) 密钥生成算法 $KeyGen(params)$. 输入系统参数和身份 ID,输出解密钥 dk.

3) 陷门生成算法 $TrapGen(dk)$. 此算法由用户执行,根据解密钥生成陷门 td.

4) 加密算法 $Encrypt(params,dk,M)$. 此算法由用户执行. 用户对明文 M 进行加密得到密文 C.

5) 解密算法 $Decrypt(params,dk,C)$. 此算法由用户执行,对密文 C 进行解密.

6) 前服务器测试算法 $TestSF(params,C_A,td_{ID_A},C_B,td_{ID_B})$. 此算法由前服务器 S_F 执行,输出判定 $MA=MB$ 是否成立的中间结果 X_A,X_B.

7) 后服务器测试算法 $TestSB(params,C_A,td_{ID_A},C_B,td_{ID_B})$. 此算法由后服务器 S_B 执行,判定 $MA=MB$ 是否成立.

3.3 安全模型

我们分别定义 DS-IBEET 协议面向恶意前服务器和恶意后服务器的安全模型.

3.3.1 恶意前服务器

本节我们定义恶意前服务器下的安全性.包括在选择关键字攻击下的 IBEET 密文语义安全和关键字猜测攻击下的陷门不可区分性.

1) 选择关键字攻击下的语义安全(SS-CKA).选择关键字攻击下的语义安全保证了敌手不能通过关键字对应的 IBEET 密文来区分关键字.即 IBEET 密文不会泄漏对应关键字的任何信息.我们用以下游戏来定义选择关键字攻击下的密文语义安全模型.

① 初始化算法($Setup$).挑战者运行初始化算法 $Setup$ 生成密钥对(pk_{S_F},sk_{S_F})和(pk_{S_B},sk_{S_B}),将(pk_{S_F},sk_{S_F},pk_{S_B})发送给敌手.

② 查询阶段 1($Query\text{-}phase\text{-}\mathrm{I}$).敌手可以适应性地对任意关键字和任意 IBEET 密文向挑战者发送查询请求.挑战者返回 0 或者 1 给敌手.

③ 挑战($challenge$).敌手发送 2 个关键字(kw_0,kw_1)给挑战者,挑战者随机选择 $b\in\{0,1\}$ 并计算:

$$CT^*_{kw}\leftarrow Encrypt(params,pk_{S_F},pk_{S_B},kw_b),$$

挑战者将 CT^*_{kw} 发送给敌手.

④ 查询阶段 2($Query\text{-}phase\text{-}\mathrm{II}$).敌手继续查询除了挑战关键字($kw_0$,$kw_1$)之外的任意关键字和 IBEET 密文.挑战者返回 0 或者 1 给敌手.

⑤ 输出($output$).最后,敌手输出它对 b 的猜测 $b'\in\{0,1\}$.如果 $b=b'$,则敌手赢得这个游戏.

我们称上述游戏中的恶意服务器 $\mathcal{A}$ 为 SS-CKA 敌手,定义它赢得游戏的优势为

$$Adv^{\text{SS-CKA}}_{S_F,\mathcal{A}}(k)=Pr[b=b']-1/2.$$

2) 关键字猜测攻击下的不可区分性(IND-KGA).在这个安全模型中,陷门不会泄漏关键字的任何信息给恶意前服务器.我们用以下游戏定义安全模型:

① 初始化算法($Setup$).挑战者运行初始化算法生成密钥对(pk_{S_F},sk_{S_F})和(pk_{S_B},sk_{S_B}),将(pk_{S_F},sk_{S_F},pk_{S_B})发送给敌手.

② 查询阶段 1($Query\text{-}phase\text{-}\mathrm{I}$).敌手可以适应性的对任意关键字和密文向挑战者发送查询请求.挑战者返回 0 或者 1 给敌手.

③ 挑战($challenge$).敌手发送 2 个关键字(kw_0,kw_1)给挑战者,挑战者随机选择 $b\in\{0,1\}$ 并计算:

$$CT^*_{kw}\leftarrow TrapGen(params,s_1).$$

挑战者将 CT^*_{kw} 发送给敌手.

④ 查询阶段 2($Query\text{-}phase\text{-}\mathrm{II}$).敌手继续查询除了挑战关键字($kw_0$,$kw_1$)之外的任意关键字和 IBEET 密文.挑战者返回 0 或者 1 给敌手.

⑤ 输出($output$).最后,敌手输出它对 b 的猜测 $b'\in\{0,1\}$.如果 $b=b'$,则敌手赢得这个游戏.

我们称上述游戏中的恶意服务器 $\mathcal{A}$ 为 IND-KGA 敌手,定义它赢得游戏的优势为

$$Adv^{\text{IND-KGA}}_{S_F,\mathcal{A}}(k)=Pr[b=b']-1/2.$$

3.3.2 恶意后服务器

恶意后服务器的安全模型和恶意前服务器的安全模型类似.

1) 选择关键字攻击下的语义安全(SS-CKA).此安全模型与恶意前服务器的安全模型类似,游戏不同之处为在初始化算法($Setup$)中,挑战者将(pk_{S_F},pk_{S_B},sk_{S_B})发送给敌手.我们称该游戏中的恶意服务器 $\mathcal{A}$ 为 SS-CKA 敌手,定义它赢得游戏的优势为

$$Adv^{\text{SS-CKA}}_{S_B,\mathcal{A}}(k)=Pr[b=b']-1/2.$$

2) 关键字猜测攻击下的不可区分性(IND-KGA).此安全模型与恶意前服务器的安全模型类似,游戏不同之处为在初始化算法($Setup$)中,挑战者将(pk_{S_F},pk_{S_B},sk_{S_B})发送给敌手.我们称该游戏中的恶意服务器 $\mathcal{A}$ 为 IND-KGA 敌手,定义它赢得游戏的优势为

$$Adv^{\text{IND-KGA}}_{S_B,\mathcal{A}}(k)=Pr[b=b']-1/2.$$

3) 关键字猜测攻击下的不可区分性(IND-KGA-Ⅱ).为了保证恶意服务器 SB 不能从内部测试状态信息获取关键字的任何信息,我们通过如下游戏定义安全模型:

① 初始化算法($Setup$).挑战者运行初始化算法生成密钥对(pk_{S_F},sk_{S_F})和(pk_{S_B},sk_{S_B}),将(pk_{S_F},pk_{S_B},sk_{S_B})发送给敌手.

② 查询阶段 1($Query\text{-}phase\text{-}\mathrm{I}$).敌手可以适应性地对任意关键字和密文的内部测试状态向挑战者发送查询请求.挑战者返回(X_A,X_B)给敌手.

③ 挑战($challenge$).敌手发送 3 个不同的关键字(kw_0,kw_1,kw_2)给挑战者,挑战者随机选择$\{b_1,b_2\}\subset\{0,1,2\}$并计算:

$$CT^*_{kw}\leftarrow Encrypt(params,pk_{S_F},pk_{S_B},kw_{b_1}),$$

$$T^*_{kw}\leftarrow TrapGen(params,pk_{S_F},pk_{S_B},kw_{b_2}),$$

$$C^*_{\text{ITS}}\leftarrow TestSF(params,sk_{S_F},CT^*_{kw},T^*_{kw}),$$

挑战者将 $C^*_{\text{ITS}}=\{X_A,X_B\}$ 发送给敌手.

④ 查询阶段 1(*Query-phase-*Ⅱ). 敌手可以适应性的对挑战关键字(kw_{b_1},kw_{b_2})以外关键字的内部测试状态向挑战者发送查询请求. 挑战者返回(X_A,X_B)给敌手.

⑤ 输出(*output*). 最后,敌手输出它对$\{b_1,b_2\}$的猜测$\{b'_1,b'_2\}\subset\{0,1,2\}$. 如果$\{b_1,b_2\}=\{b'_1,b'_2\}$,则敌手赢得这个游戏.

我们称上述游戏中的恶意后服务器 $\mathcal{A}$ 为 IND-KGA-Ⅱ敌手,定义它赢得游戏的优势为

$$Adv_{S_B,\mathcal{A}}^{\text{IND-KGA-Ⅱ}}(k)=Pr[\{b_1,b_2\}=\{b'_1,b'_2\}]-1/3.$$

在以上游戏中,b_1 和 b_2 可以相等. 此时,敌手可以获知 IBEET 密文和陷门对应的关键字是相同的,而敌手的攻击目标是猜测 3 个关键字(kw_0,kw_1,kw_2)中哪 2 个被挑战者选定了. 根据以上 5 个游戏,我们给出 DS-IBEET 协议的安全性定义如下:

定义 1. 给定安全参数 k,对任意多项式时间敌手 $\mathcal{A}_i(i=1,2,\cdots,5)$,如果 $Adv_{S_F,\mathcal{A}_1}^{\text{SS-CKA}}(k)$,$Adv_{S_B,\mathcal{A}_2}^{\text{SS-CKA}}(k)$,$Adv_{S_F,\mathcal{A}_3}^{\text{IND-KGA}}(k)$,$Adv_{S_B,\mathcal{A}_4}^{\text{IND-KGA}}(k)$,$Adv_{S_B,\mathcal{A}_5}^{\text{IND-KGA-Ⅱ}}(k)$都是可以忽略的,那么我们称 DS-IBEET 协议是安全的.

4 基于身份的双服务器等值判定协议

4.1 协议具体内容

该协议由 7 个算法组成:初始化算法、密钥生成算法、陷门生成算法、加密算法、解密算法、前服务器测试算法、后服务器测试算法.

4.1.1 初始化算法 $Setup(1^k)$

给定安全参数 $k\in Z^*$,Hash 函数 $H_1:\{0,1\}^*\rightarrow G_1$,$H_2:G_T\rightarrow G_1$,$h_3:G_T\rightarrow\{0,1\}^{l_1+l_2}$,$h:G_T\rightarrow\{0,1\}^*$. 选择随机数$(s_1,s_2)$作为前服务器 S_F 的密钥并计算 $g_1=g^{s_1}$,$g_2=g^{s_2}$. 选择随机数 s_3 作为后服务器 S_B 的密钥并计算 $g_3=g^{s_3}$. 公布参数 $params=(p,G_1,G_T,e,g,g_1,g_2,g_3,H_1,H_2,h_3,h)$.

4.1.2 密钥生成算法 $KeyGen(Params,s_1,s_2)$

给定身份 $ID\in\{0,1\}^*$,计算 $h_{ID}=H_1(ID)\in G_1^*$,解密钥 $dk_{ID}=(h_{ID}^{s_1},h_{ID}^{s_2})$.

4.1.3 陷门生成算法 $TrapGen(Params,s_1)$

用户将解密钥的一部分 $td_{ID}=h_{ID}^{s_1}$作为陷门.

4.1.4 加密算法 $Encrypt(Params,M)$

给定关键字明文 M,计算 $h_{ID}=H_1(ID)$,选择随机数$(r_1,r_2,r_3)\in Z_p^3$. 计算密文 $C=(C_1,C_2,C_3,C_4,C_5)$,其中:

$$C_1=g^{r_1},$$
$$C_2=g^{r_2},$$
$$C_3=M^{r_1}\times h(U_3^{r_3})\times H_2(U_1^{r_2}),$$
$$C_4=g^{r_3},$$
$$C_5=(M\parallel r_1)\oplus h_3(U_2^{r_3}),$$
$$U_1=e(h_{ID},g_1)\in G_T,$$
$$U_2=e(h_{ID},g_2)\in G_T,$$
$$U_3=e(h_{ID},g_3)\in G_T.$$

4.1.5 解密算法 $Decrypt(Params,C)$

使用解密钥 $dk_{ID}=(h_{ID}^{s_1},h_{ID}^{s_2})$对密文 $C=(C_1,C_2,C_3,C_4,C_5)$进行解密,计算为

$$C_5\oplus h_3(e(h_{ID}^{s_2},C_4))=M\parallel r_1,$$

如果 $C_1=g^{r_1}$ 成立,则输出 M.

4.1.6 前服务器测试算法 $TestSF(C_A,td_{ID_A},C_B,td_{ID_B})$

此算法由服务器 S_F 执行,给定 2 个密文 $C_A=(C_{1,A},C_{2,A},C_{3,A},C_{4,A},C_{5,A})$ 和 $C_B=(C_{1,B},C_{2,B},C_{3,B},C_{4,B},C_{5,B})$以及分别与之对应的陷门 $td_{ID_A}=h_{ID_A}^{s_1}$和 $td_{ID_B}=h_{ID_B}^{s_1}$,得到判定 $M_A=M_B$ 是否成立的中间结果 X_A 和 X_B. 即:

$$X_A=\frac{C_{3,A}}{H_2(e(h_{ID_A}^{s_1},C_{2,A}))},$$
$$X_B=\frac{C_{3,B}}{H_2(e(h_{ID_B}^{s_1},C_{2,B}))}.$$

4.1.7 后服务器测试算法 $TestSB(C_A,X_A,C_B,X_B)$

此算法由服务器 S_B 执行,计算并比较下列等式是否成立:

$$e(C_{1,A},h(e(h_{ID_B}^{s_3},C_{4,B}))^{-1}\times X_B)=e(C_{1,B},h(e(h_{ID_A}^{s_3},C_{4,A}))^{-1}\times X_A),$$

若成立,则输出 1,表明 $M_A=M_B$,否则输出 0.

4.2 协议正确性分析

$$X_A=\frac{C_{3,A}}{H_2(e(h_{ID_A}^{s_1},C_{2,A}))}=M_A^{r_{1,A}}\times h(U_3^{r_{3,A}})=M_A^{r_{1,A}}\times h(e(h_{ID_A},g^{s_3})^{r_{3,A}}),$$

$$X_B=\frac{C_{3,B}}{H_2(e(h_{ID_B}^{s_1},C_{2,B}))}=M_B^{r_{1,B}}\times h(U_3^{r_{3,B}})=M_B^{r_{1,B}}\times h(e(h_{ID_B},g^{s_3})^{r_{3,B}}),$$

$$e(C_{1,A},h(e(h_{ID_B}^{s_3},C_{4,B}))^{-1}\times X_B)=e(g^{r_{1,A}},X_B\times h(e(h_{ID_B}^{s_3},g^{r_{3,B}}))^{-1})=e(g^{r_{1,A}},M_B),$$

同理:

$$e(C_{1,B},h(e(h_{ID_A}^{s_3},C_{4,A}))^{-1}\times X_A)=e(g^{r_{1,B}},M_A),$$

由于:

$$e(g^{r_{1,A}},M_B^{r_{1,B}})=e(g,M_B)^{r_{1,A}r_{1,B}},$$
$$e(g^{r_{1,B}},M_A^{r_{1,A}})=e(g,M_A)^{r_{1,A}r_{1,B}},$$

若等式 $e(g^{r_{1,A}},M_B^{r_{1,B}})=e(g^{r_{1,B}},M_A^{r_{1,A}})$成立,则有 $M_A=M_B$,从而协议正确.

5 安全性分析

本节我们分析上述协议的安全性.

定理 1. 本文提出的 DS-IBEET 协议在选择关键字攻击下是语义安全的.

上述定理能够通过下述引理 1 和引理 2 证明.

引理 1. 对于任意多项式时间敌手 $\mathcal{A}$,$Adv_{S_F,\mathcal{A}}^{\text{SS-CKA}}(k)$ 是可忽略的.

证明. 首先定义下列游戏:

Game0. 这是针对恶意前服务器的初始版本 SS-CKA 游戏.

1) 初始化算法(*Setup*). 挑战者运行初始化算法生成密钥对(pk_{S_F},sk_{S_F})和(pk_{S_B},sk_{S_B}),将(pk_{S_F},sk_{S_F},pk_{S_B})发送给敌手.

2) 查询阶段 1(*Query-phase-*Ⅰ). 敌手向挑战者发送下列查询:

① *h-query*(T). 此预言机维护一个初始为空的列表 L_h,给定参数 T,随机选择 $H\in G_1$ 并将$\langle T,H\rangle$添加到列表 L_h. 返回 $h(T)$.

② H_1-*query*(T). 此预言机维护一个初始为空的列表 L_{H_1},给定参数 T,随机选择 $H_1\in G_1$ 并将$\langle T,H_1\rangle$添加到列表 L_{H_1}. 返回 $H_1(T)$.

③ H_2-*query*(T). 此预言机维护一个初始为空的列表 L_{H_2},给定参数 T,随机选择 $H_2\in G_1$ 并将$\langle T,H_2\rangle$添加到列表 L_{H_2}. 返回 $H_2(T)$.

④ h_3-*query*(T). 此预言机维护一个初始为空的列表 L_{h_3},给定参数 T,随机选择 $h_3\in G_1$ 并将$\langle T,h_3\rangle$添加到列表 L_{h_3}. 返回 $h_3(T)$.

⑤ *Encrypt-query*($Params,kw$). 此预言机维护一个初始为空的列表 $L_{Enc}=\langle kw,C\rangle$,给定参数和明文,对明文关键字 kw 进行加密:

$$C\leftarrow Encrypt(Params,kw),$$

其中:

$$C=(C_1,C_2,C_3,C_4,C_5),$$
$$C_1=g^{r_1},$$
$$C_2=g^{r_2},$$
$$C_3=M^{r_1}\times h(U_3^{r_3})\times H_2(U_1^{r_2}),$$
$$C_4=g^{r_3},$$
$$C_5=(M\parallel r_1)\oplus h_3(U_2^{r_3}),$$
$$U_1=e(h_{ID},g_1)\in G_T,$$
$$U_2=e(h_{ID},g_2)\in G_T,$$
$$U_3=e(h_{ID},g_3)\in G_T.$$

并将$\langle kw,C\rangle$添加到列表 L_{Enc} 中,挑战者返回 C 给敌手.

3) 挑战(*challenge*). 敌手 $\mathcal{A}$ 选择 2 个关键字(kw_0,kw_1),并将它们发送给挑战者,挑战者随机选择 $b\in\{0,1\}$并计算密文 C_{kw_b},将 C_{kw_b} 发送给敌手 $\mathcal{A}$.

4) 查询阶段 2(*Query-phase-*Ⅱ). 此阶段与查询阶段 1 一致.

5) 输出(*output*). 最终,敌手 $\mathcal{A}$ 输出对 b 的猜测值 b'. 如果 $b=b'$,则 $\mathcal{A}$ 赢得游戏.

我们定义 $\mathcal{A}$ 赢得游戏 Game0 的优势为 $Adv_{S_F,\mathcal{A}}^{\text{Game0}}(k)$. 根据 3.3 节定义的 SS-CKA 模型,有:

$$Adv_{S_F,\mathcal{A}}^{\text{Game0}}(k)=Adv_{S_F,\mathcal{A}}^{\text{SS-CKA}}(k).$$

Game1. Game1 与 Game0 基本相同,区别在于,在 Game1 中,挑战者用随机数 W_1 替换了 Hash 函数 $h:G_T\rightarrow\{0,1\}^*$. 具体游戏过程如下:

1) 初始化算法(*Setup*). 挑战者运行初始化算法生成密钥对(pk_{S_F},sk_{S_F})和(pk_{S_B},sk_{S_B}),将(pk_{S_F},sk_{S_F},pk_{S_B})发送给敌手.

2) 查询阶段 1(*Query-phase-*Ⅰ). 敌手向挑战者发送下列查询:

① *h-query*(T):此预言机维护一个初始为空的列表 L_h,给定参数 T,随机选择 $H\in G_1$ 并将$\langle T,H\rangle$添加到列表 L_h. 返回 $h(T)$.

② H_1-*query*(T):此预言机维护一个初始为空的列表 L_{H_1},给定参数 T,随机选择 $H_1\in G_1$ 并将$\langle T,H_1\rangle$添加到列表 L_{H_1}. 返回 $H_1(T)$.

③ H_2-*query*(T):此预言机维护一个初始为空的列表 L_{H_2},给定参数 T,随机选择 $H_2\in G_1$ 并将$\langle T,H_2\rangle$添加到列表 L_{H_2}. 返回 $H_2(T)$.

④ h_3-*query*(T):此预言机维护一个初始为空的列表 L_{h_3},给定参数 T,随机选择 $h_3\in G_1$ 并将$\langle T,h_3\rangle$添加到列表 L_{h_3}. 返回 $h_3(T)$.

⑤ *Encrypt-query*($Params,kw$). 此预言机维护一个初始为空的列表 $L_{Enc}=\langle kw,C\rangle$,给定参数和明文,对明文关键字 kw 进行加密:

$$C\leftarrow Encrypt(Params,kw),$$

其中:

$$C=(C_1,C_2,C_3,C_4,C_5),$$
$$C_1=g^{r_1},$$
$$C_2=g^{r_2},$$
$$C_3=M^{r_1}\times W_1\times H_2(U_1^{r_2}),$$
$$C_4=g^{r_3},$$
$$C_5=(M\parallel r_1)\oplus h_3(U_2^{r_3}),$$

$$U_1=e(h_{ID},g_1)\in G_T,$$
$$U_2=e(h_{ID},g_2)\in G_T,$$
$$U_3=e(h_{ID},g_3)\in G_T,$$

挑战者返回 C 给敌手.

3) 挑战(*challenge*). 敌手 $\mathcal{A}$ 选择 2 个关键字(kw_0,kw_1),并将它们发送给挑战者,挑战者随机选择 $b\in\{0,1\}$并计算密文 C_{kw_b},将 C_{kw_b} 发送给敌手 $\mathcal{A}$.

4) 查询阶段 2(*Query-phase-*Ⅱ). 此阶段与查询阶段 1 一致,但不可查询(kw_0,kw_1)的密文,若 $kw=kw_0$ 或者 $kw=kw_1$,则返回⊥并结束游戏,记为事件 *Event*1.

5) 输出(*output*). 最终,敌手 $\mathcal{A}$ 输出对 b 的猜测值 b'. 如果 $b=b'$,则 $\mathcal{A}$ 赢得游戏.

根据随机预言模型(random oracle model)的性质以及 Difference Lemma[21],在事件 *Event*1 不发生的情况下,敌手 $\mathcal{A}$ 赢得游戏 Game1 的优势满足:

$$|Adv^{\text{Game1}}_{S_F,\mathcal{A}}(k)-Adv^{\text{Game0}}_{S_F,\mathcal{A}}(k)|\leqslant Pr[Event1].$$

由于 H_2 是随机预言机,W_1 是随机选择的,且在查询过程中未返回给敌手,故根据 CDH 假设,$Pr[Event1]$是可忽略的.

Game2. Game2 与 Game1 基本相同,区别在于,在 Game2 中,挑战者用随机数 W_2 替换了 C_3. 具体游戏过程如下:

1) 初始化算法(*Setup*). 挑战者运行初始化算法生成密钥对(pk_{S_F},sk_{S_F})和(pk_{S_B},sk_{S_B}),将$(pk_{S_F},sk_{S_F},pk_{S_B})$发送给敌手.

2) 查询阶段 1(*Query-phase-*Ⅰ). 敌手向挑战者发送下列查询:

① *h-query*(T): 此预言机维护一个初始为空的列表 L_h,给定参数 T,随机选择 $H\in G_1$ 并将$\langle T,H\rangle$添加到列表 L_h. 返回 $h(T)$.

② H_1-*query*(T): 此预言机维护一个初始为空的列表 L_{H_1},给定参数 T,随机选择 $H_1\in G_1$ 并将$\langle T,H_1\rangle$添加到列表 L_{H_1}. 返回 $H_1(T)$.

③ H_2-*query*(T): 此预言机维护一个初始为空的列表 L_{H_2},给定参数 T,随机选择 $H_2\in G_1$ 并将$\langle T,H_2\rangle$添加到列表 L_{H_2}. 返回 $H_2(T)$.

④ h_3-*query*(T): 此预言机维护一个初始为空的列表 L_{h_3},给定参数 T,随机选择 $h_3\in G_1$ 并将$\langle T,h_3\rangle$添加到列表 L_{h_3}. 返回 $h_3(T)$.

⑤ *Encrypt-query*$(Params,kw)$. 此预言机维护一个初始为空的列表 $L_{Enc}=\langle kw,C\rangle$,给定参数和明文,对明文关键字 kw 进行加密:

$$C\leftarrow Encrypt(Params,kw),$$

其中:

$$C=(C_1,C_2,C_3,C_4,C_5),$$
$$C_1=g^{r_1},$$
$$C_2=g^{r_2},$$
$$C_3=W_2,$$
$$C_4=g^{r_3},$$
$$C_5=(M\,\|\,r_1)\oplus h_3(U_2^{r_3}),$$
$$U_1=e(h_{ID},g_1)\in G_T,$$
$$U_2=e(h_{ID},g_2)\in G_T,$$
$$U_3=e(h_{ID},g_3)\in G_T,$$

挑战者返回 C 给敌手.

3) 挑战(*challenge*). 敌手 $\mathcal{A}$ 选择 2 个关键字(kw_0,kw_1),并将它们发送给挑战者,挑战者随机选择 $b\in\{0,1\}$并计算密文 C_{kw_b},将 C_{kw_b} 发送给敌手 $\mathcal{A}$.

4) 查询阶段 2(*Query-phase-*Ⅱ). 此阶段与查询阶段 1 一致,但不可查询(kw_0,kw_1)对应的密文,若 $kw=kw_0$ 或者 $kw=kw_1$,则返回⊥并结束游戏,记为事件 *Event*2.

5) 输出(*output*). 最终,敌手 $\mathcal{A}$ 输出对 b 的猜测值 b'. 如果 $b=b'$,则 $\mathcal{A}$ 赢得游戏.

根据随机预言模型的性质以及 Difference Lemma[21],在事件 *Event*2 不发生的情况下,敌手 $\mathcal{A}$ 赢得游戏 Game2 的优势满足:

$$|Adv^{\text{Game2}}_{S_F,\mathcal{A}}(k)-Adv^{\text{Game1}}_{S_F,\mathcal{A}}(k)|\leqslant Pr[Event2].$$

由于 W_2 是随机选择的,且在查询过程中未返回给敌手,故根据 CDH 假设,$Pr[Event2]$是可忽略的. 证毕.

引理 2. 对于任意多项式时间敌手 $\mathcal{A}$,$Adv^{\text{SS-CKA}}_{S_B,\mathcal{A}}(k)$是可忽略的.

证明. 引理 2 的证明过程与引理 1 类似. 这里不再赘述.

定理 2. 我们的 DS-IBEET 协议在关键字猜测攻击下是不可区分的.

上述定理可以通过引理 3、引理 4 和引理 5 证明.

引理 3. 对于任意多项式时间敌手 $\mathcal{A}$,$Adv^{\text{IND-KGA}}_{S_F,\mathcal{A}}(k)$是可忽略的.

证明. 在我们的 DS-IBEET 协议中,陷门是解密钥的一部分,$td_{ID}=h^{s_1}_{ID}$,与具体的关键字无关,故陷门不会泄漏任何关键字的信息. 证毕.

引理 4. 对于任意多项式时间敌手 $\mathcal{A}$,$Adv^{\text{IND-KGA}}_{S_B,\mathcal{A}}(k)$是可忽略的.

证明. 引理 4 的证明过程与引理 3 类似. 这里不再赘述.

引理 5. 对于任意多项式时间敌手 $\mathcal{A}$, $Adv_{S_B,\mathcal{A}}^{\text{IND-KGA-II}}(k)$是可忽略的.

证明. 首先定义下列游戏:

Game0. 这是针对恶意后服务器的初始版本游戏.

1) 初始化算法($Setup$). 挑战者运行初始化算法生成密钥对对(pk_{S_F}, sk_{S_F})和(pk_{S_B}, sk_{S_B}),将(pk_{S_F}, pk_{S_B}, sk_{S_B})发送给敌手.

2) 查询阶段 1($Query\text{-}phase\text{-}\mathrm{I}$). 敌手向挑战者发送下列查询:

① $h\text{-}query(T)$: 此预言机维护一个初始为空的列表 L_h,给定参数 T,随机选择 $H\in G_1$ 并将$\langle T, H\rangle$添加到列表 L_h. 返回 $\$h(T)\$$.

② $H_1\text{-}query(T)$: 此预言机维护一个初始为空的列表 L_{H_1},给定参数 T,随机选择 $H_1\in G_1$ 并将$\langle T, H_1\rangle$添加到列表 L_{H_1}. 返回 $H_1(T)$.

③ $H_2\text{-}query(T)$: 此预言机维护一个初始为空的列表 L_{H_2},给定参数 T,随机选择 $H_2\in G_1$ 并将$\langle T, H_2\rangle$添加到列表 L_{H_2}. 返回 $H_2(T)$.

④ $h_3\text{-}query(T)$: 此预言机维护一个初始为空的列表 L_{h_3},给定参数 T,随机选择 $h_3\in G_1$ 并将$\langle T, h_3\rangle$添加到列表 L_{h_3}. 返回 $h_3(T)$.

⑤ $Encrypt\text{-}query(Params, kw)$. 此预言机维护一个初始为空的列表 $L_{Enc}=\langle kw, C\rangle$,给定参数和明文,对明文关键字 kw 进行加密:

$$C\leftarrow Encrypt(Params, kw),$$

其中:

$$C=(C_1, C_2, C_3, C_4, C_5),$$
$$C_1=g^{r_1},$$
$$C_2=g^{r_2},$$
$$C_3=M^{r_1}\times h(U_3^{r_3})\times H_2(U_1^{r_2}),$$
$$C_4=g^{r_3},$$
$$C_5=(M\parallel r_1)\oplus h_3(U_2^{r_3}),$$
$$U_1=e(h_{ID}, g_1)\in G_T,$$
$$U_2=e(h_{ID}, g_2)\in G_T,$$
$$U_3=e(h_{ID}, g_3)\in G_T,$$

并将$\langle kw, C\rangle$添加到列表 L_{Enc} 中. 挑战者返回 C 给敌手.

⑥ $TestSF(C_A, td_{ID_B}, C_B, td_{ID_B})$: 此预言机维护一个初始为空的列表 $L_{SF}=\langle C_A, td_{ID_B}, C_B, td_{ID_B}, X_A, X_B\rangle$,给定密文和相应的陷门,计算:

$$X_A=\frac{C_{3,A}}{H_2(e(h_{ID_A}^{s_1}, C_{2,A}))},$$
$$X_B=\frac{C_{3,B}}{H_2(e(h_{ID_B}^{s_1}, C_{2,B}))}.$$

3) 挑战($challenge$). 敌手 $\mathcal{A}$ 选择 3 个不同的关键字(kw_0, kw_1, kw_2),并将它们发送给挑战者,挑战者随机选择$\{b_1, b_2\}\subset\{0,1,2\}$并计算:

$$CT_{kw}^*\leftarrow Encrypt(params, pk_{S_F}, pk_{S_B}, kw_{b_1}),$$
$$T_{kw}^*\leftarrow TrapGen(params, pk_{S_F}, pk_{S_B}, kw_{b_2}),$$
$$C_{\text{ITS}}^*\leftarrow TestSF(params, sk_{S_F}, CT_{kw}^*, T_{kw}^*).$$

挑战者将 $C_{\text{ITS}}^*=\{X_A, X_B\}$发送给敌手.

4) 输出($output$). 最后,敌手 $\mathcal{A}$ 输出它对$\{b_1, b_2\}$的猜测$\{b_1', b_2'\}\subset\{0,1,2\}$. 如果$\{b_1, b_2\}=\{b_1', b_2'\}$,则 $\mathcal{A}$ 赢得这个游戏.

我们定义 $\mathcal{A}$ 赢得游戏 Game0 的优势为 $Adv_{S_B,\mathcal{A}}^{\text{Game0}}(k)$. 根据前文定义的 IND-KGA-II模型,有:

$$Adv_{S_B,\mathcal{A}}^{\text{Game0}}(k)=Adv_{S_B,\mathcal{A}}^{\text{IND-KGA-II}}(k).$$

Game1. Game1 与 Game0 基本相同,区别在于,在 Game1 中,挑战者用随机数 W_1, W_2 替换了 Hash 函数 $H_2: G_T\rightarrow G_1$. 具体游戏过程如下:

1) 初始化算法($Setup$). 挑战者运行初始化算法生成密钥对(pk_{S_F}, sk_{S_F})和(pk_{S_B}, sk_{S_B}),将(pk_{S_F}, pk_{S_B}, sk_{S_B})发送给敌手.

2) 查询阶段 1($Query\text{-}phase\text{-}\mathrm{I}$). 敌手向挑战者发送下列查询:

① $h\text{-}query(T)$: 此预言机维护一个初始为空的列表 L_h,给定参数 T,随机选择 $H\in G_1$ 并将$\langle T, H\rangle$添加到列表 L_h,返回 $h(T)$.

② $H_1\text{-}query(T)$: 此预言机维护一个初始为空的列表 L_{H_1},给定参数 T,随机选择 $H_1\in G_1$ 并将$\langle T, H_1\rangle$添加到列表 L_{H_1},返回 $H_1(T)$.

③ $H_2\text{-}query(T)$: 此预言机维护一个初始为空的列表 L_{H_2},给定参数 T,随机选择 $H_2\in G_1$ 并将$\langle T, H_2\rangle$添加到列表 L_{H_2},返回 $H_2(T)$.

④ $h_3\text{-}query(T)$: 此预言机维护一个初始为空的列表 L_{h_3},给定参数 T,随机选择 $h_3\in G_1$ 并将$\langle T, h_3\rangle$添加到列表 L_{h_3},返回 $h_3(T)$.

⑤ $Encrypt\text{-}query(Params, kw)$. 此预言机维护一个初始为空的列表 $L_{Enc}=\langle kw, C\rangle$,给定参数和明文,对明文关键字 kw 进行加密:

$$C\leftarrow Encrypt(Params, kw),$$

其中:

$$C=(C_1, C_2, C_3, C_4, C_5),$$
$$C_1=g^{r_1},$$

$$C_2=g^{r_2},$$
$$C_3=M^{r_1}\times h(U_3^{r_3})\times H_2(U_1^{r_2}),$$
$$C_4=g^{r_3},$$
$$C_5=(M\parallel r_1)\oplus h_3(U_2^{r_3}),$$
$$U_1=e(h_{ID},g_1)\in G_T,$$
$$U_2=e(h_{ID},g_2)\in G_T,$$
$$U_3=e(h_{ID},g_3)\in G_T,$$

挑战者返回 C 给敌手.

⑥ $TestSF(C_A,td_{ID_B},C_B,td_{ID_B})$:此预言机维护一个初始为空的列表 $L_{SF}=\langle C_A,td_{ID_B},C_B,td_{ID_B},X_A,X_B\rangle$,给定密文和相应的陷门,计算:

$$X_A=\frac{C_{3,A}}{W_1},$$
$$X_B=\frac{C_{3,B}}{W_2}.$$

3) 挑战($challenge$). 敌手 $\mathcal{A}$ 选择 3 个不同的关键字(kw_0,kw_1,kw_2),并将它们发送给挑战者,挑战者随机选择$\{b_1,b_2\}\subset\{0,1,2\}$并计算:

$$CT^*_{kw}\leftarrow Encrypt(params,pk_{S_F},pk_{S_B},kw_{b_1}),$$
$$T^*_{kw}\leftarrow TrapGen(params,pk_{S_F},pk_{S_B},kw_{b_2}),$$
$$C^*_{\mathrm{ITS}}\leftarrow TestSF(params,sk_{S_F},CT^*_{kw},T^*_{kw}),$$

挑战者将 $C^*_{\mathrm{ITS}}=\{X_A,X_B\}$ 发送给敌手.

4) 查询阶段 2($Query\text{-}phase\text{-}\mathrm{II}$). 此阶段与查询阶段 1 一致,但不可查询$(kw_0,kw_1)$的密文,若 $kw=kw_0$ 或者 $kw=kw_1$,则返回 $\perp$ 并结束游戏,记为事件 $Event3$.

5) 输出($output$). 最后,敌手 $\mathcal{A}$ 输出它对$\{b_1,b_2\}$的猜测$\{b_1',b_2'\}\subset\{0,1,2\}$. 如果$\{b_1,b_2\}=\{b_1',b_2'\}$,则 $\mathcal{A}$ 赢得这个游戏.

根据随机预言模型的性质以及 Difference Lemma[21],在事件 $Event3$ 不发生的情况下,敌手 $\mathcal{A}$ 赢得游戏 Game1 的优势满足:

$$|Adv^{\mathrm{Game1}}_{S_B,\mathcal{A}}(k)-Adv^{\mathrm{Game0}}_{S_B,\mathcal{A}}(k)|\leqslant Pr[Event3].$$

由于 W_1,W_2 是随机选择的,且在查询过程中未返回给敌手,故根据 CDH 假设,$Pr[Event3]$是可忽略的.

Game2. Game2 与 Game1 基本相同,区别在于,在 Game2 中,挑战者用随机数 W_3,W_4 替换了 X_A,X_B. 具体游戏过程如下:

1) 初始化算法($Setup$). 挑战者运行初始化算法生成密钥对(pk_{S_F},sk_{S_F})和(pk_{S_B},sk_{S_B}),将$(pk_{S_F},pk_{S_B},sk_{S_B})$发送给敌手.

2) 查询阶段 1($Query\text{-}phase\text{-}\mathrm{I}$). 敌手向挑战者发送下列查询:

① $h\text{-}query(T)$:此预言机维护一个初始为空的列表 L_h,给定参数 T,随机选择 $H\in G_1$ 并将$\langle T,H\rangle$添加到列表 L_h. 返回 $h(T)$.

② $H_1\text{-}query(T)$:此预言机维护一个初始为空的列表 L_{H_1},给定参数 T,随机选择 $H_1\in G_1$ 并将$\langle T,H_1\rangle$添加到列表 L_{H_1}. 返回 $H_1(T)$.

③ $H_2\text{-}query(T)$:此预言机维护一个初始为空的列表 L_{H_2},给定参数 T,随机选择 $H_2\in G_1$ 并将$\langle T,H_2\rangle$添加到列表 L_{H_2}. 返回 $H_2(T)$.

④ $h_3\text{-}query(T)$:此预言机维护一个初始为空的列表 L_{h_3},给定参数 T,随机选择 $h_3\in G_1$ 并将$\langle T,h_3\rangle$添加到列表 L_{h_3}. 返回 $h_3(T)$.

⑤ $Encrypt\text{-}query(Params,kw)$. 此预言机维护一个初始为空的列表 $L_{Enc}=\langle kw,C\rangle$,给定参数和明文,对明文关键字 kw 进行加密:

$$C\leftarrow Encrypt(Params,kw),$$

其中:

$$C=(C_1,C_2,C_3,C_4,C_5),$$
$$C_1=g^{r_1},$$
$$C_2=g^{r_2},$$
$$C_3=M^{r_1}\times h(U_3^{r_3})\times H_2(U_1^{r_2}),$$
$$C_4=g^{r_3},$$
$$C_5=(M\parallel r_1)\oplus h_3(U_2^{r_3}),$$
$$U_1=e(h_{ID},g_1)\in G_T,$$
$$U_2=e(h_{ID},g_2)\in G_T,$$
$$U_3=e(h_{ID},g_3)\in G_T,$$

挑战者返回 C 给敌手.

⑥ $TestSF(C_A,td_{ID_B},C_B,td_{ID_B})$:此预言机维护一个初始为空的列表 $L_{SF}=\langle C_A,td_{ID_B},C_B,td_{ID_B},X_A,X_B\rangle$,给定密文和相应的陷门,计算:

$$X_A=W_3,$$
$$X_B=W_4.$$

3) 挑战($challenge$). 敌手 $\mathcal{A}$ 选择 3 个不同的关键字(kw_0,kw_1,kw_2),并将它们发送给挑战者,挑战者随机选择$\{b_1,b_2\}\subset\{0,1,2\}$并计算:

$$CT^*_{kw}\leftarrow Encrypt(params,pk_{S_F},pk_{S_B},kw_{b_1}),$$
$$T^*_{kw}\leftarrow TrapGen(params,pk_{S_F},pk_{S_B},kw_{b_2}),$$
$$C^*_{\mathrm{ITS}}\leftarrow TestSF(params,sk_{S_F},CT^*_{kw},T^*_{kw}),$$

挑战者将 $C^*_{\mathrm{ITS}}=\{X_A,X_B\}$ 发送给敌手.

4) 查询阶段 2($Query\text{-}phase\text{-}\mathrm{II}$). 此阶段与查询阶段 1 一致,但不可查询$(kw_0,kw_1)$的密文,若 $kw=kw_0$ 或者 $kw=kw_1$,则返回 $\perp$ 并结束游戏,记为事件 $Event4$.

5) 输出(*output*). 最后,敌手 $\mathcal{A}$ 输出它对$\{b_1, b_2\}$的猜测$\{b_1', b_2'\} \subset \{0,1,2\}$. 如果$\{b_1, b_2\} = \{b_1', b_2'\}$,则 $\mathcal{A}$ 赢得这个游戏.

根据随机预言模型的性质以及 Difference Lemma[21],在事件 *Event*4 不发生的情况下,敌手 $\mathcal{A}$ 赢得游戏 Game2 的优势满足:

$$|Adv_{S_B,\mathcal{A}}^{\text{Game2}}(k) - Adv_{S_B,\mathcal{A}}^{\text{Game1}}(k)| \leqslant Pr[Event4].$$

由于 W_3, W_4 是随机选择的,且在查询过程中未返回给敌手,故根据 CDH 假设,$Pr[Event4]$是可忽略的. 证毕.

6 性能分析

本文从计算开销和通信开销 2 个方面对我们设计的协议的性能进行分析.

为了对协议的性能进行评估,我们在阿里巴巴的 ECS 云主机上调用 MIRACL 库[22],获得一些基本密码操作的执行时间. 云主机的配置环境如表 1 所示,椭圆曲线参数如表 2 所示. 我们使用 Tate 对,大素数 p 为 512 b,大素数阶 q 为 160 b. 设 M, Exp, BP, H, h 和 PA 分别表示标量乘法、群 G_1 上的模指数运算、双线性对运算、HashToPoint 运算、普通 Hash 运算和点加运算. 这些基本操作的执行时间如表 3 所示.

Table 1 System Information

表 1 系统配置信息

Notations	Value
System	Ubuntu 14.04
System Type	64-bit Operating System
CPU	Intel® Xeon® E5-26300@2.30 GHz
Memery Size/GB	1

Table 2 Parameters of Elliptic Curve

表 2 椭圆曲线参数

Param	Value
n/b	512
p	8BA2A5229BD9C57CFC8ACEC76DFDBF3E3E1952C6B3193ECF5C571FB502FC5DF410F9267E9F2A605BB0F76F52A79E8043BF4AF0EF2E9FA78B0F1E2CDFC4E8549B
A	1
B	0
cof	117454A4537B38AF9F9159D8EDBFB7E7C7C2E48760E930A461D5F451F9D9210DC70095F4B241FF57F1BB0549C
q	8000000000000000000000000000000020001

Table 3 Execution Time of Basic Operations

表 3 基本操作执行时间 ms

Operations	Execution Time
BP	5.275
PA	0.012
M	1.97
Exp	0.331
H	5.101
h	0.009

根据统计,我们的 DS-IBEET 协议的各个算法的计算开销如表 4 所示. 加密算法、解密算法、前服务器测试算法和后服务器测试算法的运行时间分别为 24.9 ms,17.952 ms,24.692 ms 和 14.508 ms. 显然,此协议可以运行在资源受限的移动设备上(如病人的手机/平板电脑等).

Table 4 Computation Cost

表 4 计算开销 ms

Algorithms	Operations and Cost Time
Encryption	$3BP+M+6Exp+H+2h=24.9$
Decryption	$2BP+M+Exp+H=17.952$
TestSF	$2BP+2M+2H=24.692$
TestSB	$2BP+2M+2h=14.508$

本协议的通信开销如表 5 所示. 其中,公钥长度为群 G_1 中元素比特长度的 2 倍,密文长度为群 G_1 中元素比特长度的 4 倍加上一个 Z_q 上大数比特长度. 陷门长度为群 G_1 中元素比特长度.

Table 5 Communication Cost

表 5 通信开销

Scheme	Size of Pubkey	Size of Ciphertext	Size of Trapdoor	Keyword Search
DS-IBEET	$2\|G_1\|$	$4\|G_1\|+\|Z_q\|$	$\|G_1\|$	√

$|G_1|$: The bit length of element in G_1.

$|Z_q|$: Bit length of number in Z_q.

7 结 论

最近,一些带密文等值判定的公钥加密协议被提出. 然而,它们仅有一个服务器参与密文等值判定过程,这种设计容易遭受恶意服务器内部关键字猜测攻击. 为了解决这个问题,本文首次提出了基于双服务器的带密文等值判定的公钥加密协议,并在随机预言机模型下证明了其安全性. 同时,我们利用 MIRACL 大数库对其计算性能进行了评估,并分析

了其通信开销,结果表明我们的 DS-IBEET 协议性能良好,且能够在资源受限的移动设备上运行.

由于基于身份的加密协议存在密钥托管问题,我们下一步工作将尝试研究一个无证书的带等值判定的公钥加密协议.

参 考 文 献

[1] Boneh D, Di Crescenzo G, Ostrovsky R, et al. Public key encryption with keyword search [C] //Advances in Cryptology—EUROCRYPT 2004. Berlin: Springer, 2004: 506-522

[2] Fang Liming, Willy S, Ge Chunpeng, et al. Public key encryption with keyword search secure against keyword guessing attacks without random oracle [J]. Information Sciences, 2013, 238: 221-241

[3] Abdalla M, Bellare M, Catalano D, et al. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions [J]. Journal of Cryptology, 2008, 21(3): 350-391

[4] Hwang Y H, Lee P J. Public key encryption with conjunctive keyword search and its extension to a multi-user system [C] //Proc of the 1st Int Conf on Pairing-Based Cryptography. Berlin: Springer, 2007: 2-22

[5] Orencik C, Selcuk A, Savas E, et al. Multi-keyword search over encrypted data with scoring and search pattern obfuscation [J]. International Journal of Information Security, 2016, 15(3): 251-269

[6] Ibraimi L, Nikova S, Hartel P, et al. Public-key encryption with delegated search [C] //Proc of the 9th Int Conf on Applied Cryptography and Network Security (ACNS2011). Berlin: Springer, 2011: 532-549

[7] Byun J W, Rhee H S, Park H A, et al. Off-line key-word guessing attacks on recent keyword search schemes over encrypted data [C] // Proc of the 3rd VLDB Workshop on Secure Data Management (SDM 2006). Berlin: Springer, 2006: 75-83

[8] Shi Jie, Lai Junzuo, Li Yingjiu, et al. Authorized keyword search on encrypted data [C] //Proc of European Symp on Research in Computer Security (ESORICS 2014). Berlin: Springer, 2014: 419-435

[9] Cao Ning, Wang Cong, Li Ming, et al. Privacy-preserving multi-keyword ranked search over encrypted cloud data [J]. IEEE Trans on Parallel and Distributed Systems, 2014, 25(1): 222-233

[10] Wang Kaixuan, Li Yuxi, Zhou Fucai, et al. Multi-keyword fuzzy search over encrypted data [J]. Journal of Computer Research and Development, 2017, 54(2): 348-360 (in Chinese)
(王恺璇, 李宇溪, 周福才, 等. 面向多关键字的模糊密文搜索方法[J]. 计算机研究与发展, 2017, 54(2): 348-360)

[11] Chen Dongdong, Cao Zhenfu, Dong Xiaolei. Online/offline ciphertext-policy attribute-based searchable encryption [J]. Journal of Computer Research and Development, 2016, 53(10): 2365-2375 (in Chinese)
(陈冬冬, 曹珍富, 董晓蕾. 在线/离线密文策略属性基可搜索加密[J]. 计算机研究与发展, 2016, 53(10): 2365-2375)

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

[13] Yang Guomin, Tan C H, Huang Qiong, et al. Probabilistic public key encryption with equality test [C] //Proc of Cryptographers' Track at the RSA Conference. Berlin: Springer, 2010: 119-131

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

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

[16] Ma Sha, Zhang Mingwu, Huang Qiong, et al. Public key encryption with delegated equality test in a multi-user setting [J]. The Computer Journal, 2014, 58(4): 986-1002

[17] Huang Kaibin, Tso R, Chen Y-C, et al. PKE-AET: Public key encryption with authorized equality test [J]. The Computer Journal, 2015, 58(10): 2686-2697

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

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

[20] Wu Libing, Zhang Yubo, Choo Kim-Kwang, et al. Efficient and secure identity-based encryption scheme with equality test in cloud computing [J]. Future Generation Computer Systems, 2017, 73: 22-31

[21] Victor S. Sequences of games: A tool for taming complexity in security proofs [OL]. (2004-11-30) [2017-06-10]. https://eprint.iacr.org/2004/332.pdf

[22] CertiVox. MIRACL cryptographic library: Multiprecision integer and rational arithmetic C/C++ library [OL]. (2006-08-01) [2017-06-10]. https://github.com/miracl/MIRACL

Wu Libing, born in 1972. Received his BS and MS degrees in computer science from Central China Normal University, Wuhan, China, in 1994 and 2001, respectively. Received his PhD degree in computer science from Wuhan University in 2006. Professor in the School of Computer Science, Wuhan University. Senior member of IEEE and CCF. His main research interests include distributed computing, trusted software and wireless sensor networks.

Zhang Yubo, born in 1988. Received his BS degree in computer science and technology from Wuhan University of Science and Technology, Wuhan, China in 2011. PhD candidate in computer system architecture from Wuhan University. Student member of CCF. His main research interests include cryptography and information security.

He Debiao, born in 1980. Received his PhD degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University in 2009. Professor of the State Key Laboratory of Software Engineering, School of Computer Science, Wuhan University. His main research interests include cryptography and information security, in particularly, cryptographic protocols.