



基于单服务器的模指数安全外包计算方案

丁伟杰^{1,2}

- (1. 浙江警察学院计算机与信息技术系, 浙江 杭州 310053;
2. 浙江工业大学信息工程学院, 浙江 杭州 310023)

摘要: 目前, 在离散对数密码协议中, 模指数外包计算方案大部分都是针对素数的, 很少有有关合数的研究成果。并且大多数模指数外包计算方案都是基于双服务器的, 该类方案通常要求两个服务器中至少有一个是诚实的, 但是在实际环境中可能并不存在完全可信的云服务器。基于单个不可信服务器模型提出了一个新的复合模指数安全外包计算方案。该方案采用新的数学分割方式, 能够保证底数和指数的隐私性。与已有方案相比, 该方案的外包计算结果可验证率也有很大程度的提高, 用户能够以 119/120 的概率检测出错误结果。

关键词: 云计算; 外包计算; 模指数

中图分类号: TN918

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2018001

Secure outsource computing scheme of modular exponentiation based on single server

DING Weijie^{1,2}

1. Department of Computer and Information Technology, Zhejiang Police College, Hangzhou 310053, China
2. College of Information and Engineering, Zhejiang University of Technology, Hangzhou 310023, China

Abstract: At present, in discrete-log based cryptographic protocols, most of the computational models of modular exponentiation are for primes, while less work has been done for composite. What's more, most schemes are based on two servers, in which it requires at least one server to be honest. However, there may not be a fully trusted cloud server in the actual environment. Then a new secure method for outsourcing exponentiation modular a composite which based on a single server was proposed. The scheme used a new mathematical division method, it could ensure the privacy of the base and exponentiation. Compared with the existing schemes, the checkability of our scheme can be greatly improved. The user can detect the error result with the probability of 119/120.

Key words: cloud computing, outsource computing, modular exponentiation

1 引言

云计算^[1,2]作为一种新兴的计算模式, 能够为

用户提供大量的虚拟化计算资源。外包计算能解决用户资源受限且计算量繁重这一难题, 因而引起了人们的广泛关注。但外包计算在为带来

收稿日期: 2017-03-20; 修回日期: 2017-07-31

基金项目: 国家自然科学基金资助项目 (No.U1509219)

Foundation Item: The National Natural Science Foundation of China (No.U1509219)

便捷的同时,也带来了相应的安全挑战^[3,4]。用户的外包数据中往往包含了一些敏感信息,因此需要保证用户数据的隐私性。通常外包计算需要消耗大量的计算资源,云服务器很可能为了节省资源(或存在一些 bug)^[5]而返回不正确的计算结果,因此也要保证计算结果的可验证性。

随着云计算的快速发展,云计算辐射的领域越来越广,并给人们的日常生活带来了诸多便利。云计算在公安系统中的应用非常广泛,目前很多公安电子数据取证工作都开始使用云计算,在公安信息中心建设中也开始使用云计算技术^[6,7]。伴随现今“互联网+”技术发展的浪潮,云计算将在公安系统中发挥更加重要的作用。

在密码系统中,模指数运算是最常见、最耗时操作之一,如何将模指数运算安全高效地外包给服务器成为了外包计算中最亟待解决的问题。Chaum 等人^[8]首次提出了“wallets with observers”的概念,在每次执行任务时允许用户在自己的设备上安装一块硬件来执行计算。Hohenberger 等人^[9]形式化定义了这一概念,提出了第一个模指数安全外包计算的方案,该方案基于两个服务器。在实现外包计算时,该方案提供了一个量化外包计算效率及结果可验证率的框架,但方案中结果的可验证率仅为 1/2。Chen 等人^[10]提出了一个新的基于两个服务器的模指数安全外包计算方案,其性能与结果可验证率皆优于参考文献[9],它的结果可验证率可达到 2/3。Ye 等人^[11]在 Chen 方案的基础上提出了一个新的基于两个服务器的模指数安全外包计算方案,该方案可验证率得到了很大的提高,它的结果可验证率可达到 19/20。但参考文献[9-11]的可验证率都不能达到 1,因此,外包者都存在被恶意服务器欺骗的风险。参考文献[9-11]都是基于两个服务器的外包计算模型,但是在实际环境中可能并不存在完全可信的云服务器。Dijk 等人^[12]提出了基于单个服务器的模指数外包计算模型,但服务器能够获得模指数运算中

的底数,不能保证输入的隐私性。Wang 等人^[13]提出了一个基于单个服务器的模指数安全外包计算方案,但结果的可验证概率仅为 1/2。

本文基于一种新的数学分割方式,提出了一个新的复合模指数安全外包计算方案 CME。该方案是基于单个服务器的,同时实现底数和指数的隐私性。与之前提出的方案相比,该方案对外包计算结果的可验证率有了极大程度的提高,当云服务器不诚实时,外包用户能够以 119/120 的概率检测出错误。

2 相关简介

2.1 系统模型

算法中涉及两个实体,分别为用户 T 和云服务器 U 。其具体交互过程如图 1 所示,具体介绍如下。

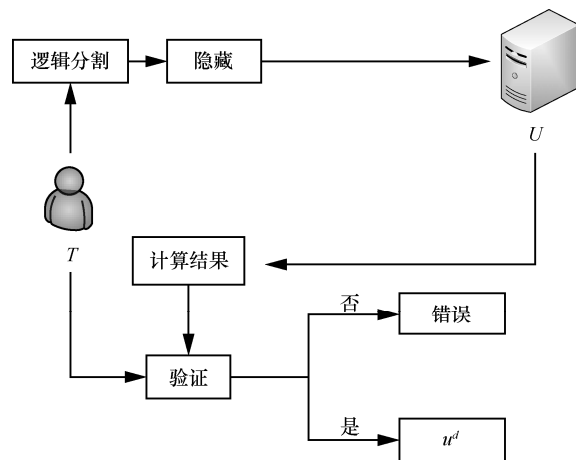


图1 系统模型

(1) 预处理

用户 T 调用子程序 RandN, 产生随机数对, 利用这些随机数对对原始数据进行盲化处理。

(2) 分割

用户 T 利用预处理中生成的随机数对对原始数据进行逻辑分割, 并将盲化的数据发给服务器 U 。

(3) 计算

收到盲化的数据后, 服务器 U 计算相应的模指数结果, 并将计算结果返回给用户 T 。

(4) 验证

收到服务器 U 计算的结果后, 用户 T 恢复模



指数计算的值。然后用户 T 自己对云服务器返回结果的正确性进行验证。

2.2 RandN 子程序

为了加快计算速度、提高计算效率, 参考文献[14]中提出了名为 RandN 的子程序。其输入为 $N = p \cdot q$ 和 $e \in Z_N^*$ (p, q 是素数), 输出为随机的、独立的数对 (x, x^e) ($x \in Z^*$ 且长度为 n)。

2.3 安全定义

通常来说, 用户 T 将一些工作安全地外包给服务器 U , 并且 (T, U) 是一个关于加密算法 Alg 安全外包的实现, 如果 T 和 U 共同实现 Alg, 也就是 $\text{Alg} = T^U$, 且假设 T 可以做攻击者 U' (而不是 U) 的随机预言机询问, 其中记录着 U 所有的计算以及它试图恶意的行为, U' 不能获取任何有关 T^U 的输入、输出信息。下文将介绍一个加密算法的安全外包^[9]。

定义 1 (algorithm with outsource-I/O) 一个算法 Alg 是符合外包输入/输出规范的, 如果其接受 5 个输入及产生 3 个输出。前 3 个输入是由可信实体产生的, 并且按照攻击者 $A = (E, U')$ 对他们了解的信息多少进行分类。第一个输入是诚实而秘密的, E 和 U' 都不能获取该值; 第二个输入是诚实而受保护的, E 可能会获取, 但 U' 不能获取; 第三个输入是诚实而不受保护的, E 和 U 都可能获取该值。此外, 还有两个由环境 E 产生的敌对选择输入: 敌对的受保护的输入, E 能够知道该值, 但 U' 不能够知道; 敌对的不受保护的输入, E 和 U 都能够获取该值。同样地, 第一个输出是秘密的, E 和 U' 都不知道; 第二个输出是受保护的, 可能被 E 知道, 但 U' 不能知道; 第三个输出是不受保护的, 攻击者 A 的双方都可能知道。

定义 2 (外包安全) 假定 Alg 是一个外包 I/O 的算法, 一个算法对 (T, U) 被称为 Alg 的安全外包实现。需满足如下两个要求。

正确性: T^U 是 Alg 的一个正确实现。

安全性: 对所有概率多项式时间 PPT 攻击者 $A = (E, U')$, 都存在预期的概率多项式模拟器 (S_1, S_2) , 使得下述随机变量对在计算上是不可区分的。

对 1: $\text{EVIEW}_{\text{real}} \sim \text{EVIEW}_{\text{ideal}}$ 。

对 2: $\text{UVIEW}_{\text{real}} \sim \text{UVIEW}_{\text{ideal}}$ 。

3 方案设计

本文提出了一个新的基于单个不可信服务器模型的复合模指数外包计算方案, 在该方案中, 用户 T 可以通过利用子程序 RandN 将模指数的计算外包给云服务器 U , 提出的算法能够保证攻击者 A 获取不到任何关于输入和输出的隐私信息。使用子程序 RandN 产生的随机数对来对指数和底数进行隐藏, 底数 u 由随机数 v_1, g_1, w_1 和随机数 v_2, g_3, w' 进行隐藏, 而指数 d 则由随机数 l_1, t_1 和 l_2, t_2 进行隐藏。在整个交互过程中, 服务器 U 能得到的数据为 $(r_1, c_1), (r_2, c_2), (l_1, w), (k_1, w_1), (l_2, w'), (k_2, w_2)$ 。而数据 $v_1, g_1, v_2, g_2, t_1, t_2$ 是服务器 U 无法获取的, 从而保证底数及指数的隐私性。与参考文献[9]相似, $U(x, y) \rightarrow y^x \bmod N$ 表示 U 以数对 (x, y) 作为输入, 输出 $y^x \bmod N$ 。

在本文方案中, 输入为 $u \in Z_N^*$ 和 $d \in Z_N^*$, 输出为 $u^d \bmod N$ 。其中, p, q 是两个大素数, 并且满足 $N = p \cdot q$ 。本文方案包含的子算法具体如下。

(1) 预处理

T 调用 4 次 RandN 子程序, 生成 4 个随机数对 $(g_1, g_1^e), (g_2, g_2^e), (g_3, g_3^e)$ 和 (g_4, g_4^e) , 并定义 $v_1 = g_1^e \bmod N$ 、 $w_1 = g_2^e \bmod N$ 、 $v_2 = g_3^e \bmod N$ 以及 $w_2 = g_4^e \bmod N$ 。

(2) 分割

本文方案对 u 和 d 进行逻辑分割处理, 使 U 能够简易地计算出结果, 更重要的是保证 U 不能获取 T 的任何敏感信息。

第一次分割:

$$u^d = (g_1 w)^d = v_1 g_1^{r_1} w^d \quad (1)$$

其中, $w = u/g_1$, $r_1 = d - e$ 。

通过第一次分割, 底数 u 已经被隐藏, 且将底数 u 分割成了随机数 v_1 、 g_1 、 w_1 。接下来要进一步实现对指数 d 的隐藏。

第二次分割:

$$\begin{aligned} u^d &= v_1 g_1^{r_1} w^d \\ &= v_1 (g_2 c_1)^{r_1} w^d \\ &= v_1 c_1^{r_1} w_1^{k_1} g_2^{t_1} w^d \\ &= v_1 c_1^{r_1} w_1^{k_1} g_2^{t_1} w^{l_1+t_1} \\ &= v_1 c_1^{r_1} w_1^{k_1} (w_1)^{k_1} (wg_2)^{t_1} \end{aligned} \quad (2)$$

其中, $c_1 = g_1/g_2$, $t_1 = r_1 - k_1 e$, $d = l_1 + t_1$ 。

通过第二次分割, 实现了对指数 d 的隐藏, 将指数 d 分割成了随机数 l_1 、 t_1 。

同时, T 使用相似的方法分割 u^{rd} :

$$u^{rd} = (g_3 w')^{rd} = v_2 g_3^{r_2} (w')^{rd} \quad (3)$$

其中, $w' = u/g_3$, $r_2 = rd - e$:

$$\begin{aligned} u^{rd} &= v_2 g_3^{r_2} (w')^{rd} \\ &= v_2 (g_4 c_2)^{r_2} (w')^{rd} \\ &= v_2 c_2^{r_2} w_2^{k_2} g_4^{t_2} (w')^{rd} \\ &= v_2 c_2^{r_2} w_2^{k_2} g_4^{t_2} (w')^{l_2+t_2} \\ &= v_2 c_2^{r_2} (w')^{l_2} (w_2)^{k_2} (w' g_4)^{t_2} \end{aligned} \quad (4)$$

其中, $c_2 = g_3/g_4$, $t_2 = r_2 - k_2 e$, $rd = l_2 + t_2$ 。

(3) 计算

利用 T 上传的数据 (r_1, c_1) 、 (r_2, c_2) 、 (l_1, w) 、 (k_1, w_1) 、 (l_2, w') 、 (k_2, w_2) , U 计算如下模幂运算并将相应的结果返回给 T 。 $U(r_1, c_1) \rightarrow c_1^{r_1}$, $U(r_2, c_2) \rightarrow c_2^{r_2}$, $U(l_1, w) \rightarrow w^{l_1}$, $U(k_1, w_1) \rightarrow (w_1)^{k_1}$, $U(l_2, w') \rightarrow (w')^{l_2}$, $U(k_2, w_2) \rightarrow (w_2)^{k_2}$ 。

(4) 验证

T 检测 U 是否产生了正确的响应, 即验证式(5)是否成立:

$$(v_1 c_1^{r_1} w^{l_1} (w_1)^{k_1} (wg_2)^{t_1})^r = v_2 c_2^{r_2} (w')^{l_2} (w_2)^{k_2} (w' g_4)^{t_2} \quad (5)$$

如果式(5)不成立, 则 T 输出“error”; 否则, T 自己计算出最终结果 $u^d = v_1 c_1^{r_1} w^{l_1} (w_1)^{k_1} (wg_2)^{t_1}$ 。

为了降低用户的计算负担, 这里的 r 应该选取一个尽可能小的数值, 从而方便用户计算 u^r 。例如, r 可以在一个小范围[2,11]中选取。为了进一步增强输入的安全性, 随机盲化因子 t_1 和 t_2 至少为 64 bit。

4 方案分析

通过下面的引理证明本文方案的正确性。只要云服务器是诚实可信的, 使用本文方案将能够得到正确的计算结果 u^d , 且式(5)成立。

引理 1 如果云服务器是诚实可信的, 则可以得到正确的计算结果且式(5)成立。

证明:

$$\begin{aligned} u^d &= v_1 c_1^{r_1} w^{l_1} (w_1)^{k_1} (wg_2)^{t_1} \\ &= v_1 c_1^{r_1} (w_1)^{k_1} g_2^{t_1} w^{l_1+t_1} \\ &= v_1 c_1^{r_1} g_2^{k_1 e} g_2^{t_1} w^d \\ &= v_1 c_1^{r_1} g_2^{r_1} w^d \\ &= v_1 (c_1 g_2)^{r_1} w^d \\ &= g_1^{e+r_1} w^d \\ &= (g_1 w)^d \\ &= u^d \end{aligned} \quad (6)$$

$$\begin{aligned} u^{rd} &= v_2 c_2^{r_2} (w')^{l_2} (w_2)^{k_2} (w' g_4)^{t_2} \\ &= v_2 c_2^{r_2} (w_2)^{k_2} g_4^{t_2} w'^{l_2+t_2} \\ &= v_2 c_2^{r_2} g_4^{r_2} w'^{rd} \\ &= v_2 (c_2 g_4)^{r_2} w'^{rd} \\ &= g_3^{e+r_2} w'^{rd} \\ &= (g_3 w')^{rd} \\ &= u^{rd} \\ &= (u^d)^r \end{aligned} \quad (7)$$

由式(6)和式(7)可知, 只要云服务器是诚实可信的, 那么式(5)就是成立的, 也就可以计算出 $u^d = v_1 c_1^{r_1} w^{l_1} (w_1)^{k_1} (wg_2)^{t_1}$ 。



引理 2 算法 CME 是安全外包计算的实现, 其中输入 (d, u) 可能是可信任的秘密输入, 或可信任的受保护输入, 或是攻击者的受保护输入。

证明: 令 $A = (E, U')$ 是一个 PPT 攻击者。

首先, 证明 $\text{EVIEW}_{\text{real}} \sim \text{EVIEW}_{\text{ideal}}$ 。模拟器 S_1 和实际执行的方式相同。如果 (d, u) 是诚实且秘密的输入, 那 S_1 操作如下: 当接受第 i 轮输入后, S_1 以 (α_j, β_j) 的形式对 U'_1 和 U'_2 进行 6 次随机查询。 S_1 从每个程序中随机检测两个输出。如果检测出错误, S_1 保存所有状态并输出 $(\text{error}, \phi, 1)$; 如果没有检测出错误, S_1 检测其余的输出。如果所有的检测都通过, S_1 输出 $(\phi, \phi, 0)$; 否则, S_1 选取一个随机数 r 并输出 $(r, \phi, 1)$ 。其他情况下, S_1 保存合适的状态。在实际的实验中, 上述 3 个查询在计算上是没有区别的。在理想的实验中, 输入是随机地、均匀地选择的。因此在实际实验和理想实验中, (U'_1, U'_2) 的输入分布在计算上是没有区别的。接下来, 可以推出 $\text{EVIEW}_{\text{real}}^i \sim \text{EVIEW}_{\text{ideal}}^i$, 也就意味着 $\text{EVIEW}_{\text{real}} \sim \text{EVIEW}_{\text{ideal}}$ 。

第二步, 证明 $\text{UVIEW}_{\text{real}} \sim \text{UVIEW}_{\text{ideal}}$ 。模拟器 S_2 操作如下: 当收到第 i 轮输入后, S_2 以 (α_j, β_j) 的形式对 U'_1 和 U'_2 进行 6 次随机查询。然后 S_2 保存自己的状态以及 U'_1 和 U'_2 的状态。尽管恶意环境 E 能够区分出实际实验和理想实验, E 却不会将这些信息传达给 U'_1 和 U'_2 。因此, 对每一个循环 i , 都有 $\text{UVIEW}_{\text{real}}^i \sim \text{UVIEW}_{\text{ideal}}^i$, 进而可以得到 $\text{UVIEW}_{\text{real}} \sim \text{UVIEW}_{\text{ideal}}$ 。

引理 3 若服务器 U 想要欺骗用户, CME 的结果可验证率为 119/120。

证明: 如果服务器 U 想要欺骗用户, 服务器首先要计算出正确的 $(r_1, c_1), (r_2, c_2), (l_1, w), (l_2, w'), (k_1, w_1), (k_2, w_2)$ (这里为了便于描述, 把 $(r_1, c_1), (r_2, c_2), (l_1, w), (l_2, w'), (k_1, w_1), (k_2, w_2)$ 记为 $(\alpha_i, \beta_i), (\alpha'_i, \beta'_i)$), 并选择一个随机数 R , 然后返回 $R\beta_i^{\alpha_i}, R^r\beta_i^{\alpha'_i}$ 给用户。

根据返回结果可以得出 $R(v_1\beta_1^{\alpha_1}\beta_2^{\alpha_2}(\beta_3)^{\alpha_3}(\beta_2g_2)^{t_1}) = Ru^d, R^rv_2\beta_1^{\alpha'_1}(\beta_2')^{\alpha'_2}(\beta_3')^{\alpha'_3}(\beta_2'g_4)^{t_2} = R^ru^{rd}$ 。

若这个不正确的计算结果可以通过验证, 那么在这个计算过程中, 服务器 U 首先需要任意选择一组正确的数对与 R 相乘, 同时还需要获取 r 的值, 其中 R 是服务器为了欺骗用户选择的随机数, r 是用户为了防止服务器欺骗用户选择的随机数。 U 在计算过程中共获取 6 组用户的随机数对, 因此猜测出 $(\alpha_i, \beta_i), (\alpha'_i, \beta'_i)$ 概率为 $(1/6) \times (1/5) = 1/30$ 。在实验过程中选取 r 的范围为 $[2, 5]$, 因此 U 获取 r 的概率为 $1/4$ 。所以服务器 U 能够欺骗用户的概率为 $(1/30) \times (1/4) = 1/120$ 。即用户的可验证概率为 $1 - 1/120 = 119/120$ 。

5 性能分析

分别对本文提出的方案进行理论分析和性能分析。

5.1 理论分析

将本文提出的方案与参考文献[13]中提出的方案进行对比。 $r \in [2, 5]$, MM 表示模乘, MInv 表示模逆, Invoke 表示调用子程序。忽略一些如模加法的运算。表 1 中给出了方案有效性和结果可验证率的对比。

表 1 方案有效性和结果可验证率的对比

	参考文献[13]方案	本文方案
安全模型	单 UP	单 UP
MM	$12 + 1.5\chi$	$13 + 1.5\lg t_1 + 1.5\lg t_2 + 1.5\lg r$
MInv	4	4
Invoke(Rand)	6	4
Invoke(U)	4	6
可验证概率	1/2	119/120

由表 1 可以直观地看出本文方案的结果可验证率有较大的提高。虽然参考文献[13]中的算法在 MM 和 MInv 方面似乎要比本文方案更好, 但参考

文献[13]中的算法并不是一个实用的算法。当 $n=1$ 时,最大的验证率为 $1/2$ 。如果变成复杂的多模指数情况(比如用户外包 $\prod_{i=1}^n u_i^{a_i}$ ($n > 10$)),那么可验证率将低于 $1/10$ 。这意味着服务器能够随意地返回错误结果。

5.2 实验分析

对本文方案进行实验模拟,实验利用两台计算机进行模拟仿真,计算机的规格都是 Inter Core i5 CPU @ 3.1 GHz, 内存 4 GB, 使用 Java 语言编程。

在计算模指数时,无论外包还是不外包,模指数运算的时间开销都会随着计算次数的增加而增大。但当使用本文提出的外包方案时,其时间开销增长率明显小于不使用外包计算方案时。并且随着计算次数的增加,两种计算方式的时间差越来越明显。图2为模指数外包方案实验结果。

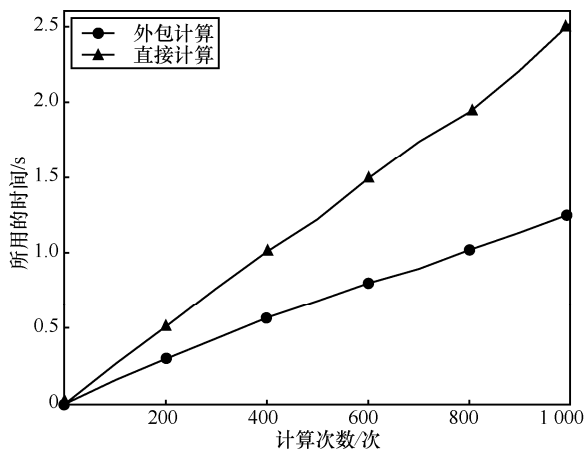


图2 模指数外包方案实验结果

当用户不将模指数运算外包时,模指数的运算将消耗大量的时间,并且其时间开销会随着指数位数的增加而线性增长。当使用本文提出的外包计算方案时,时间开销会大幅减小,并且时间开销基本恒定,不会随着指数位数的增加而改变。图3给出了模指数 u^d 的时间开销实验结果。

6 结束语

本文基于单个不可信服务器模型,提出了一种新的复合模指数安全外包计算方案。本文方案

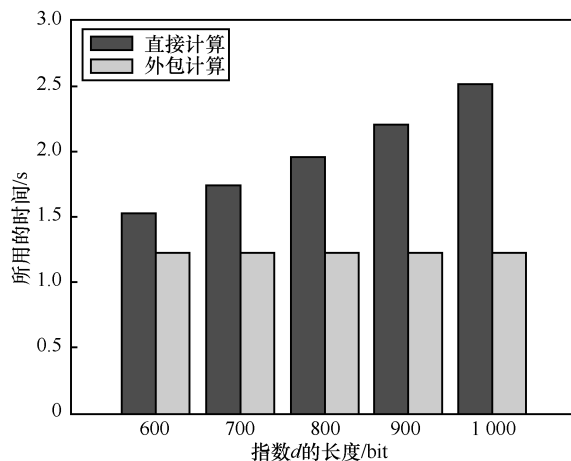


图3 模指数 u^d 的时间开销实验结果

使用新的数学分割方式,能够保证模指数运算底数和指数的隐私性,并且极大地提高了现有方案中外包计算结果的可验证率。如果云服务器不诚实,用户能够以 $119/120$ 的概率检测出错误结果。

参考文献:

- [1] 汪来富, 沈军, 金华敏. 云计算应用安全研究[J]. 电信科学, 2010, 26(6): 67-70.
WANG L F, SHEN J, JIN H M. Research on application security of cloud computing[J]. Telecommunications Science, 2010, 26(6): 67-70.
- [2] 何明, 郑翔, 赖海光, 等. 云计算技术发展及应用探讨[J]. 电信科学, 2010, 26(5): 42-46.
HE M, ZHENG X, LAI H G, et al. Development and application of cloud computing technology[J]. Telecommunications Science, 2010, 26(5): 42-46.
- [3] REN K, WANG C, WANG Q. Security challenges for the public cloud[J]. IEEE Internet Computing, 2012, 16(1): 69-73.
- [4] MEZGAR I, RAUSCHECKER U. The challenge of networked enterprises for cloud computing interoperability[J]. Computers in Industry, 2014, 65(4): 657-674.
- [5] 陈克非, 翁健. 云计算环境下数据安全与隐私保护[J]. 杭州师范大学学报(自然科学版), 2014(6): 561-570.
CHEN K F, WEN J. Data security and privacy protection in cloud computing environment[J]. Journal of Hangzhou Normal University (Natural Science Edition), 2014(6): 561-570.
- [6] 何明. 云计算在公安电子数据取证技术中的研究与应用[J]. 网络安全技术与应用, 2015(7): 88-90.
HE M. Research and application of cloud computing in police electronic data forensics[J]. Network Security Technology and Application, 2015(7): 88-90.
- [7] 农卫涛. 云计算在公安信息化建设中的应用[J]. 数字通信世



- 界, 2016(1): 97, 103.
- NONG W T. Application of cloud computing in police information construction[J]. Digital Communications World, 2016(1): 97, 103.
- [8] CHAUM D, PEDERSEN T P. Wallet databases with observers[C]//International Cryptology Conference on Advances in Cryptology, August 16-20, 1992, London, UK. [S.l.:s.n.], 1992: 89-105.
- [9] HOHENBERGER S, LYSYANSKAYA A. How to securely outsource cryptographic computations[J]. Theory of Cryptography, 2005(3378): 264-282.
- [10] CHEN X, LI J, MA J, et al. New algorithms for secure outsourcing of modular exponentiations[J]. IEEE Transactions on Parallel & Distributed Systems, 2014, 25(9): 2386-2396.
- [11] YE J, CHEN X, MA J. An improved algorithm for secure outsourcing of modular exponentiations[C]//IEEE International Conference on Advanced Information Networking and Applications Workshops, March 24-27, 2015, Gwangju, South Korea. Piscataway: IEEE Press, 2015: 73-76.
- [12] DIJK M, CLARKE D, GASSEND B, et al. Speeding up exponentiation using an untrusted computational resource[J]. Designs Codes & Cryptography, 2006, 39(2): 253-273.
- [13] WANG Y, WU Q, WONG D S, et al. Securely outsourcing exponentiations with single untrusted program for cloud storage[C]//European Symposium on Research in Computer Security, September 7-11, 2014, Wroclaw, Poland. [S.l.:s.n.], 2014: 326-343.
- [14] BOYKO V, PEINADO M, VENKATESAN R. Speeding up discrete log and factoring based schemes via precomputations[C]//International Conference on the Theory and Applications of Cryptographic Techniques, May 31-June 4, 1998, Espoo, Finland. [S.l.:s.n.], 1998: 221-235.

[作者简介]



丁伟杰 (1980—), 男, 浙江警察学院计算机与信息技术系讲师, 浙江工业大学信息工程学院博士生, 主要研究方向为网络安全、公安信息技术应用等。