

doi :10.3969/j.issn.1005-3026.2016.05.003

基于双线性映射的公共可验证外包计算方案

李福祥¹,霍建秋²,林慕清¹,周福才²

(1. 东北大学 计算机科学与工程学院,辽宁 沈阳 110819 ;2. 东北大学 软件学院,辽宁 沈阳 110819)

摘 要 : 已有可验证计算方案存在以下不足 :一是只有计算委托方才可以对计算结果进行验证 ;二是即使计算委托方可以授权其他用户进行验证 ,但也将自身验证密钥交给授权用户 .针对上述不足 ,提出一个支持公共验证的外包计算模型 ,给出其算法形式化定义及安全模型 ,并利用双线性映射提出了一个包含三方实体的公共可验证外包计算方案 ,给出了方案算法的具体描述、实体间的通信协议以及效率分析 ,方案验证无需私钥参与 ,实现了公共可验证性 .在可证安全模型下证明该方案具有不可伪造性 ,其安全性可归约于 l -SBDH 问题的困难性 .

关 键 词 : 双线性映射 ;公共可验证 ;外包计算 ;不可伪造性 ;可验证计算

中图分类号 : TP 309.2 **文献标志码 :** A **文章编号 :** 1005-3026(2016)05-0619-05

Bilinear Map-based Public Verifiable Outsourced Computation Scheme

LI Fu-xiang¹,HUO Jian-qiu¹,LIN Mu-qing¹,ZHOU Fu-cai²

(1. School of Computer Science & Engineering ,Northeastern University ,Shenyang 110819 ,China ;2. School of Software ,Northeastern University ,Shenyang 110819 ,China. Corresponding author :ZHOU Fu-cai ,professor ,E-mail :fczhou@mail.neu.edu.cn)

Abstract : There are two shortcomings for the existing verifiable computation schemes. One is that only the owner who outsourced the computation can verify the result , and the other is when the owner authorizes other users to verify the result , he has to send his secret key to all the authorized users. In order to overcome the problems , an outsourced computation model was proposed which supports the public verification. The description and security model were formalized and a publicly verifiable outsourced computation scheme , which is based on the bilinear map and contains three entities , was also presented. The algorithm implementation and the communication protocol were also described in details. The verification phase in the scheme does not need the owner 's secret key so it can be publicly verifiable. The scheme can be proved to satisfy unforgeability in the security model , and the security can be reduced to the hardness of the l -SBDH problem.

Key words : bilinear map ; public verifiable ; outsourced computation ; unforgeability ; verifiable computation

外包计算^[1-2]也称为委托计算 ,是一种客户-服务器的服务模式 ,它允许客户将函数计算过程交付给服务器 ,而客户自身仅产生待计算变量及接收返回的计算结果 .这种模式对终端计算能力要求低 ,使其可以完成对复杂计算问题的处

理 .因此外包计算不但使计算资源得到更加充分的利用 ,同时也促进了移动便携设备的发展 .但由于计算过程外包于网络服务器 ,所以如何保证计算被正确地执行 ,同时满足验证效率要优于计算效率 ,即实现高效的可验证外包计算成为保障网

络信息安全研究领域的热点之一。

Anderson 等在 SETI@home 实现计划^[3]中实现了利用全球空闲计算机资源进行计算,开始了对外包计算的研究。Gennaro 等^[4]介绍了可验证计算的概念,并且给出了包含源客户以及服务器两方实体的可验证计算(verifiable computation, VC)模型。该模型规定验证信息私钥必须由源客户保留,不可分发给其他客户,因此只有该客户可以验证计算结果。以 VC 模型为基础, Gennar 等利用加密的布尔电路^[5]和全同态加密技术^[6]实现了一个函数外包方案。Parno 等在 VC 中扩展了公共授权^[7]和公共可验证性^[8]两个重要性质,弥补了 VC 模型^[9]无法进行验证授权的不足。他们通过建立 VC 和属性基加密的关联,给出了具有公共授权的 VC 模型,即公共可验证计算(public verifiable computation, PVC)模型^[10]。该模型中有初始客户、服务端和授权客户三类实体,与 VC 不同的是, PVC 授权用户可以是多个,满足公共可验证性。靳方元等^[11]提出了一个基于全同态加密的委托计算方案,从猜测私钥信息、伪造输入输出和得到对应明文 3 方面证明了方案的安全性。此外,还有学者针对集合的交、并、差集等运算的可验证性进行了研究^[12-13],但其研究仍没有考虑到公共可验证性。其他研究还包括针对密文数据的交集运算^[14]以及针对私有大数据集合的交集运算等^[15]。现有模型大多存在实现公共授权时,需要将验证密钥交给授权用户的不足,因此并不完全符合公共可验证性的要求。

在文献[9-10]提出的 VC 模型基础上,针对公共可验证性要求,本文提出了一个公共可验证计算外包模型(public verifiable outsourced computation model, PVOCM),并对其正确性和安全性给出了形式化定义。设计了满足公共验证性的三方外包计算通信协议,以双线性映射理论为基础,针对多项式求值外包,给出了一个新型公共可验证外包计算方案,该方案能被证明具有在选择消息攻击下的不可伪造性,其安全性可归约为 l -strong bilinear Diffie-Hellman(l -SBDH)问题假设。通过对其计算效率分析,方案满足客户查询和验证开销小于函数计算开销的外包计算效率要求。

1 相关知识

1.1 双线性映射

设 G 和 G_T 为 p 阶循环群, g 为 G 生成元,则

有 $e: G \times G \rightarrow G_T$, 且 e 满足条件 $e(g^a, g^b) = e(g, g)^{ab}$ 以及 $e(g, g) \neq 1$, 即 e 满足双线性、非退化性和可计算性, 则称 e 为双线性映射。

1.2 计算 l -SBDH 问题及困难性假设

定义 1 计算 l -SBDH 问题。设 λ 为安全参数, G 和 G_T 为 p 阶循环群, g 为 G 生成元, $e: G \times G \rightarrow G_T$ 为双线性映射, 给定 $g, g^t, \dots, g^{t^l} \in G$, 其中 $t \in Z_p^*$, 随机选取 $c \in Z_p^*$, 计算 $e(g, g)^{\frac{1}{t+c}}$ 。

定义 2 如果找不到概率多项式时间算法在 $\text{poly}(\lambda)$ 时间内解决 l -SBDH 问题, 即找不到满足 l -SBDH 问题的二元组, 使得 $(c, e(g, g)^{\frac{1}{t+c}}) \in Z_p^* \setminus \{-t\} \times G_T$, 则称 l -SBDH 问题是困难的。

1.3 多项式的表达形式

定义 3 多重集合。集合 S 是全集 U 的一个子集, S 中的每一个元素都可以出现多次, 这样的集合 S 称为多重集合, 元素 x 的阶就是该元素在多重集合 S 中出现的次数, 表示为 $\mathcal{X}(x)$ 。

定义 4 多项式的表达形式。设 $S_{d,n}$ 表示满足 n 元 d 次的多项式的所有项的集合, $\mathcal{J}(x_1, x_2, \dots, x_n)$ 表示 n 元 d 次多项式求值函数, 则函数 $\mathcal{J}(x_1, x_2, \dots, x_n)$ 可以表示为

$$\mathcal{J}(x_1, x_2, \dots, x_n) = \sum_{S \in S_{d,n}} C_S \cdot \prod_{i \in S} x_i^{\mathcal{X}(i)}. \quad (1)$$

2 PVOCM

2.1 系统模型

PVOCM 由 5 个概率多项式时间算法组成, $\text{PVOCM} = \{\text{KeyGen}, \text{Setup}, \text{Compute}, \text{Verify}, \text{Update}\}$, 各算法的具体描述为

1) $(pk, sk) \leftarrow \text{KeyGen}(\lambda, F)$ 密钥生成算法。输入一个秘密参数 λ 和一个函数簇 F , 输出公私密钥对 (pk, sk) 。

2) $\text{sign}(f) \leftarrow \text{Setup}(sk, pk, f)$ 初始化算法。将 (pk, sk) 及函数簇 F 中某具体函数 f 作为输入, 输出为函数 f 的签名值 $\text{sign}(f)$ 。

3) $(r, p) \leftarrow \text{Compute}(pk, f, a)$ 函数求值算法。将公钥 pk 和某具体函数 f 以及计算变量 a 作为输入, 输出结果 $r = \mathcal{J}(a)$ 和证据 p 。

4) $\{0, 1\} \leftarrow \text{Verify}(pk, \text{sign}(f), a, r, p)$ 验证算法。将公钥 pk , 签名值 $\text{sign}(f)$, 变量 a 及结果 $r = \mathcal{J}(a)$ 和证据 p 作为输入, 输出结果 1 或 0, 其中 1 代表验证结果正确, 0 代表验证结果错误。

5) $\text{sign}(f') \leftarrow \text{Update}(sk, pk, \text{sign}(f), f')$ 更新算法。将公私密钥对 (pk, sk) , $\text{sign}(f)$ 和新选函数 f' 作为输入, 输出的是 f' 的签名值 $\text{sign}(f')$ 。

2.2 安全模型

2.2.1 正确性

设 λ 是初始化选定的安全参数, \mathcal{F} 是函数簇, 定义算法 check 为多项式时间算法, 其以函数输入 x , 计算结果 r 和函数 f 为输入, 当结果确实为 f 计算结果时, 返回 accept , 否则返回 reject , 即

$\{\text{accept}, \text{reject}\} \leftarrow \text{check}(x, r, f_i)$

对于任何 $i = 0, \dots, \text{poly}(\lambda)$, 变量 $x \in \text{domain}(f_i)$, PVOCM 是正确的, 则要求各算法正确执行后, 算法 Compute 结果满足:

$$\Pr \left[\begin{array}{l} 0 \leftarrow \text{Verify}(pk, \text{sign}(f_i), x, r, p) \\ \text{accept} \leftarrow \text{check}(x, r, f_i) \end{array} \right] \leq \text{neg}(\lambda),$$

其中 $\text{neg}(\lambda)$ 为可忽略函数。

2.2.2 安全性

设 λ 为安全参数, 通过敌手 Adv 和挑战者 Chal 的挑战实验给出安全性定义。

挑战实验:

定义敌手 Adv 对于选定变量 $b = (b_1, b_2, \dots, b_n)$, 它试图通过以下步骤来伪造一个结果通过验证:

1) 挑战者 Chal 执行算法 KeyGen , 输出公私密钥对 (pk, sk) , 将公钥 pk 给 Adv , 私钥 sk 保留。执行算法 Setup , 生成初始函数签名值 $\text{sign}(f_0)$ 。

2) Adv 向 Chal 进行查询, Chal 向 Adv 返回 $\text{sign}(f_0)$ 。之后 Adv 可以向 Chal 进行 $\text{poly}(\lambda)$ 次的查询, 每一次查询 Chal 进行一次 Update 算法, 并将更新的函数签名值 $\text{sign}(f_i)$ 发送至 Adv 。

3) Adv 对于特定具体函数 f_i 输出一个伪造结果 r' 及伪造的证据 $p' = (p'_1, p'_2, \dots, p'_n)$ 。

4) 若 $\text{Verify}(pk, \text{sign}(f_i), b, r', p')$ 输出为 1, 且有 $f_i(b) \neq r'$, 则 Adv 伪造成功, 否则 Adv 失败。

定义 5 PVOCM 安全性。令 $\text{PVOCM} = \{\text{KeyGen}, \text{Setup}, \text{Compute}, \text{Verify}, \text{Update}\}$ 是公共可验证外包计算模型, 在上述挑战实验过程中, 如果敌手 Adv 伪造成功的概率满足:

$$\Pr \left[\begin{array}{l} \{x, r, p, i\} \leftarrow \text{Adv}\{1^k, pk\}; \\ 1 \leftarrow \text{Verify}(pk, \text{sign}(f_i), x, r, p); \\ \text{reject} \leftarrow \text{check}(x, r, f_i). \end{array} \right] \leq \text{neg}(\lambda),$$

则称 PVOCM 满足不可伪造性。

3 具体方案

针对多项式求值计算, 基于双线性映射提出包含源端 Sou (函数拥有者), 服务端 Ser (计算执行者) 以及客户端 Cl (计算发起和验证者) 三方实

体的公共可验证外包计算方案 (bilinear map-based public verifiable outsourced computation scheme, BPVOCS), 并证明该方案满足不可伪造性, 其安全性可归约于 l -SBDH 问题困难性假设。

方案构成:

BPVOCS 包括 5 个多项式时间算法, 即 $\text{BPVOCS} = \{\text{KeyGen}, \text{Setup}, \text{Compute}, \text{Verify}, \text{Update}\}$, 其具体算法描述为

1) $(pk, sk) \leftarrow \text{KeyGen}(\lambda, \mathcal{F})$ 密钥生成算法。函数簇 \mathcal{F} 是满足 n 元 d 次多项式函数的集合。 KeyGen 先生成一个 λ 位的素数 p , 再生成 p 阶循环群 G 和 G_T , 及双线性映射函数 e 。取 G 中生成元 g 和 n 个随机值 $t_1, t_2, \dots, t_n \in \mathbb{Z}_p^n$ 。计算签名生成集 $W_{n,d} = \{g^{\prod_{i \in S} t_i^{f(i)}} : \forall S \in S_{n,d}\}$ 。其中 $W_{n,d}$ 中的元素个数为 C_{n+d}^n 。

最终公钥 $pk = \{g, W_{n,d}, G, G_T, e\}$, 私钥 sk 包括 n 个随机值 t_1, t_2, \dots, t_n 。

2) $\text{sign}(f) \leftarrow \text{Setup}(sk, pk, f)$ 函数签名值初始化算法。设 $f(t)$ 表示 n 元 d 次多项式函数, 总共有 k 项, 由多重集合 S_1, S_2, \dots, S_k 来表示。对于 $i = 1, 2, \dots, k$, 有 $\forall S_i \in S_{n,d}$, 每项系数为 c_1, c_2, \dots, c_k 。使用 $W_{n,d}$ 和 pk 中椭圆曲线参数计算多项式签名值:

$$\text{sign}(f) = g^{f(t)} = \prod_{j=1}^k (g^{\prod_{i \in S_j} t_i^{f(i)}})^{c_j}. \quad (2)$$

3) $(r, p) \leftarrow \text{Compute}(pk, f, a)$ 函数计算算法。首先计算 $r = f(a)$, 再产生证据 p 。

若结果 r 是正确计算得到的结果, 则可找出 n 个多项式 $q_1(x), q_2(x), \dots, q_n(x)$, 使得 $f(x) - r = \sum_{i \in [n]} (x_i - a_i) q_i(x)$ 。证据 p 包含 n 个元素 p_1, p_2, \dots, p_n , 其中 $p_i = g^{q_i(t)}$ 。

4) $\{0, 1\} \leftarrow \text{Verify}(pk, \text{sign}(f), a, r, p)$ 结果验证算法。利用 $p = (p_1, p_2, \dots, p_n)$, pk , 函数签名值 $\text{sign}(f)$ 和变量 a 以及双线性映射 e 来验证是否有式 (3) 成立。

$$e(\text{sign}(f), g^{-r} \cdot g) \stackrel{?}{=} \prod_{i=1}^n e(g^{t_i - a_i}, p_i). \quad (3)$$

又由式 (3) 可以转化为验证:

$$f(t) - r \stackrel{?}{=} \sum_{i=1}^n (t_i - a_i) q_i(t). \quad (4)$$

如果式 (4) 成立, 则该算法接受结果 r , 输出 1, 反之拒绝结果 r 并且输出 0。

5) $\text{sign}(f') \leftarrow \text{Update}(sk, pk, \text{sign}(f), f')$ 函数签名值的更新。 f 表示当前计算函数, f' 表示新计

算函数,则

sign(f') = sign(f)g^{(c's-cs)\prod_{i\in S}q^{(i)}}. (5)

4 安全性证明和性能分析

4.1 正确性证明

BPVOCS 是正确的,表示如果方案步骤都是正确执行,产生的结果都是按照正确步骤执行得到的,没有被恶意篡改的,则客户端将以极大概率接受结果,即客户端执行验证算法 1←Verify(pk, sign(f) a, r, p). 证明过程略.

4.2 安全性证明

BPVOCS 满足不可伪造性,即要证明敌手 Adv,对于选定变量 b=(b1, b2, ..., bn)试图伪造出正确结果和证据(r', p')是不可行的.

定理 1 BPVOCS 的不可伪造性. 敌手 Adv 伪造结果及证据(b, r', p'),如果最终客户端 C 的验证算法输出为 1 的概率是极小概率 neg(λ),即 Adv 伪造成功的概率是极小概率 neg(λ),则满足不可伪造性.

证明 敌手 Adv 选定 b=(b1, b2, ..., bn). 源端 Sou 随机选择一个系列变量 t=(t1, t2, ..., tn),创建相应的 Wn,d,令 t1=τ,对于 i∈{2, 3, ..., n}, Sou 选择随机的(r_i, s_i),使得 bi=ri·b1+si,计算 ti=ri·τ+si. Sou 记录所有 ri,计算 Wn,d. Sou 选定初始函数 f0,计算出初始 sign(f0).

查询阶段. Adv 向 Sou 查询初始函数签名值 sign(f0)以及后续 sign(fi).

伪造阶段. Adv 指定 Sou 使用过的某函数 fi,伪造 b 对应的错误值 r'和证据 p',其中 r'≠f(b) p'=(p'1, p'2, ..., p'n).

最后客户验证阶段,客户 C 接到 Adv 发过来的(b, r', p'),及 Sou 端 pk 和对应 sign(fi).

假设敌手 Adv 伪造成功,则有 r'≠fi(b)和 Verify(pk, sign(fi) b, r', p')=1 同时成立. 令 δ=r'-fi(b),明显 δ≠0. 由于客户端 C 处验证成功,所以有 e(g, g)^{(t)-r'} = \prod_{i\in [n]} e(g^{ti-bi} p'_i) 成立.

则有 f(t)-f(b) = \sum_{i\in [n]} (ti-bi)qi(t). 因此有 e(g, g)^{\delta} = \prod_{i\in [n]} e(g^{ti-bi} g^{qi(t)} (p'_i)^{-1}). 又因为有 ti-bi=r(\tau-b1),所以有

(\prod_{i\in [n]} e(g^{ti-bi} g^{qi(t)} (p'_i)^{-1}))^{\frac{1}{\tau-b1}} = \prod_{i\in [n]} e(g^{ri} g^{qi(t)} (p'_i)^{-1}).

这样在客户端 C 处可以得到

e(g, g)^{\frac{1}{\tau-b1}} = (\prod_{i\in [n]} e(g^{ri}, g^{qi(t)} (p'_i)^{-1}))^{\delta^{-1}}.

至此,在客户端 C 处已知的 g, g^{\tau}, g^{\tau^2}, ..., g^{\tau^l} 情况下,在多项式时间里得到了 e(g, g)^{\frac{1}{\tau-b1}},而这与 l-SBDH 假设相矛盾,因此假设中敌手成功伪造错误结果通过验证不成立,证明了本文 BPVOCS 满足不可伪造性.

4.3 效率分析

源端 Sou 在 KeyGen 算法计算签名生成集 Wn,d 时, Wn,d 中的元素个数是 C_{n+d}^n,而生成每个元素的复杂度为 O(d),故生成 Wn,d 复杂度为 O(d·C_{n+d}^n),即 O(C_{n+d}^n). 私钥生成的复杂度为 O(n),产生双线性配对参数的复杂度都是低于 O(n),故 KeyGen 算法的计算复杂度为 O(C_{n+d}^n). 对于 Setup 算法,函数签名值 sign(f)要计算 C_{n+d}^n 项,故其复杂度也是 O(C_{n+d}^n). 对于 Update 算法,由于满足 n 元 d 次多项式的函数之间的区别最多是 C_{n+d}^n 个数,所以 Update 算法的计算复杂度为 O(C_{n+d}^n). 由于 Sou 端 KeyGen 算法和 Setup 算法只执行一次,之后执行 Update 算法,所以 Sou 运行的时候计算复杂度为 O(C_{n+d}^n).

在服务端 Ser 执行 Compute 算法. 首先计算函数值 r=f(b),由于函数的最高次幂是 d,所以计算函数值的复杂度为 O(n^d). 之后 Ser 进行证据 p=(p1, p2, ..., pn)的生成,证据 pi=g^{qi(t)},故生成每个证据的计算复杂度和生成函数签名值的复杂度一样,都是 O(C_{n+d}^n),所以生成证据 p 的计算复杂度为 O(C_{n+d}^n). 因此, Ser 运行时计算复杂度为 O(n^d).

客户端 C 执行 Verify 算法. \prod_{i=1}^n e(g^{ti-ai} pi) 的复杂度为 O(n),计算 e(sign(f)g^{-r}, g)的复杂度为 O(1),之后将两个结果进行比较,复杂度也为 O(1). 因此 C 的计算复杂度为 O(n).

由此可见客户端 C 的工作量是远低于其他两个实体所需的工作量的,这就满足本文减轻客户端 C 的负担,使得计算能力较弱的设备也可以进行复杂计算任务的设计初衷.

5 结 论

1) 提出公共可验证外包计算模型 PVOCM,介绍了三方通信协议以及模型的正确性和安全性定义.

2) 针对多项式求值计算, 提出基于双线性映射的公共可验证外包计算方案 BPVOCS, 给出方案具体实现算法, 并证明其满足正确性和不可伪造性, 其安全性可归约于 l -SBDH 问题的困难性。

3) 同其他同类方案相比, 验证过程中不需私钥的参与, 满足公共可验证性, 且客户验证代价明显低于服务端计算代价, 符合计算外包的本质要求。

参考文献：

- [1] Green M, Hohenberger S, Waters B. Outsourcing the decryption of ABE ciphertexts[C]//Proceeding of the 20th USENIX Conference on Security. San Francisco, 2011 :34 – 34.
- [2] Chung K M, Kalai Y, Vadhan S. Improved delegation of computation using fully homomorphic encryption[M]. Berlin :Springer 2010 :483 – 501.
- [3] Anderson D P, Cobb J, Korpela E, et al. SETI@ Home :an experiment in public-resource computing [J]. *Communications of the ACM*, 2002, 45(11) :56 – 61.
- [4] Gennaro R, Gentry C, Parno B. Non-interactive verifiable computing :outsourcing computation to untrusted workers [M]. Berlin :Springer 2010 :465 – 482.
- [5] Yao A. Protocols for secure computations[C]//Proceeding of the 23rd Annual Symposium on Foundations of Computer Science. New York, 1982 :160 – 164.
- [6] Gentry C. A fully homomorphic encryption scheme[D]. Stanford :Stanford University, 2009.
- [7] Barbosa M, Farshim P. Delegatable homomorphic encryption with applications to secure outsourcing of computation[M]. Berlin :Springer 2012 :296 – 312.
- [8] Goldwasser S, Kalai Y T, Rothblum G N. Delegating computation :interactive proofs for muggles[C]//Proceeding of the 40th Annual ACM Symposium on Theory of Computing. Victoria, 2008 :113 – 122.
- [9] Bitansky N, Canetti R, Chiesa A, et al. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again[C]//Proceeding of the 3rd Innovations in Theoretical Computer Science Conference. Cambridge, 2012 :326 – 349.
- [10] Parno B, Raykova M, Vaikuntanathan V. How to delegate and verify in public :verifiable computation from attribute-based encryption[M]. Berlin :Springer 2012 :422 – 439.
- [11] 靳方元, 朱艳琴, 罗喜召. 基于可验全同态加密的委托计算方案[J]. *计算机工程*, 2012, 38(23) :150 – 153.
(Jin Fang-yuan, Zhu Yan-qin, Luo Xi-zhao. Delegation of computation scheme based on verifiable fully homomorphic encryption[J]. *Computer Engineering*, 2012, 38(23) :150 – 153.)
- [12] Papamanthou C, Tamassia R, Triandopoulos N. Optimal verification of operations on dynamic sets[M]//Advances in Cryptology-CRYPTO 2011. Berlin :Springer 2011 :91 – 110.
- [13] Canetti R, Paneth O, Papadopoulos D N. Triandopoulos, verifiable set operations over outsourced databases[M]. Berlin :Springer 2014 :113 – 130.
- [14] Zheng Q, Xu S. Verifiable delegated set intersection operations on outsourced encrypted data[C]//Proceeding of the 2015 IEEE International Conference on Cloud Engineering (IC2E). Tempe, 2015 :175 – 184.
- [15] Kamara S, Mohassel P, Raykova M, et al. Scaling private set intersection to billion-element sets [M]//Financial Cryptography and Data Security. Berlin :Springer, 2014 :195 – 215.