doi:10.11959/j.issn.1000-436x.2015172

隐私保护的可验证多元多项式外包计算方案

任艳丽[1]，谷大武[2]，蔡建兴[1]，黄春水[1]
(1.上海大学 通信与信息工程学院，上海 200444;
2.上海交通大学 电子信息与电气工程学院，上海 200240）

摘 要：随着云计算的发展和大数据时代的到来，如何对隐私数据进行外包计算且有效验证计算结果具有重要的现实意义。基于多线性映射和同态加密方案，提出了可验证的多元多项式外包计算方案，用户可准确验证外包计算结果的正确性。方案在标准模型中可证安全，且多项式函数和用户输入对于服务器都是保密的。分析表明，用户计算量远小于服务器的计算代价以及直接计算多项式函数。
关键词：云计算；大数据；多元多项式；可验证外包计算；多线性映射
中图分类号：TP309 **文献标识码**：A

Verifiably private outsourcing scheme for multivariate polynomial evaluation

REN Yan-li[1], GU Da-wu[2], CAI Jian-xing[1], HUANG Chun-shui[1]
(1.School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China;
2.School of Electronic Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai 200240,China)

Abstract: With the development of cloud computing and big data, it had important practical significance for how to outsource private data and verify the computing result efficiently. A verifiably outsourcing scheme for multivariate polynomial evaluation based on multilinear maps and homomorphic encryption was proposed where the user could verify the computing result exactly. The proposed scheme is provably secure without random oracles and the multivariate polynomial itself and the input of the function are private for the server. Moreover, the cost of the user is much smaller than that of the server, and it is much smaller than that of computing the multivariate polynomial directly.
Key words: cloud computing; big data; multivariate polynomial; verifiable outsourcing computing; multilinear map

1 引言

随着云计算的发展和移动设备的普及，普通用户更倾向于将复杂运算外包给计算能力很强的服务提供者（如云服务器等），由服务提供者返回最终的计算结果，这将为用户节省大量的时间和计算开销。但是，服务提供者并不是完全可信的，有可能泄漏用户数据，或故意返回错误的计算结果。因此，如何不泄露用户隐私并验证外包服务的计算结果具有重要的理论价值与现实意义。

可验证计算（VC, verifiable computation）方案[1]很好地解决了上述问题，用户将需要计算的函数和输入数据加密后发给服务提供者，由服务提供者返回计算结果及对结果的证明。用户可验证计算结果的正确性，且验证的计算量远远小于直接计算函数，如图 1 所示。在可验证计算方案中，函数本身和用户输入都要求是保密的，服务提供者不能提供错误的计算结果，而且还能通过验证[2]。可验证计

收稿日期：2014-08-25；**修回日期**：2014-10-24
基金项目：国家自然科学基金资助项目（61202367）；教育部高校博士点基金资助项目（20120073110094）；上海市自然科学基金资助项目（12ZR1443700）；上海市教委创新基金资助项目（14YZ020）
Foundation Items: The National Natural Science Foundation of China (61202367); The Doctoral Fund of Ministry of Education of China (20120073110094); The Natural Science Foundation of Shanghai (12ZR1443700); The Innovation Program of Shanghai Municipal Education Commission (14YZ020)

算一般分为2类：1)一般函数的可验证外包计算[3~5]，适合于任何函数的计算；2)特殊函数的外包计算，例如模指数运算[6,7]、多项式计算[8]、基于属性解密运算[9,10]等。其中，可验证的多项式外包计算在信息安全、线性代数和信号处理等领域均有广泛应用，因此引起了人们的广泛关注。

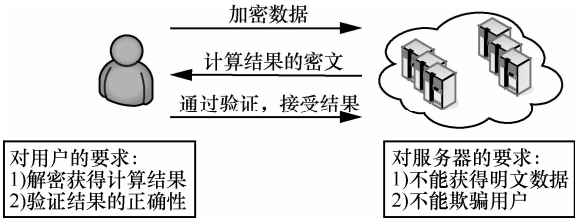

图1 可验证计算方案

Benabbas 等提出了基于大数据的多项式外包计算，方案实现了多项式函数保密，但不能实现用户输入的隐私性[11]。Fiore 等构造了可公开验证的多项式外包计算方案，任何人均可验证计算结果的正确性，但该方案不能实现函数和用户输入的隐私性[12]。Ma 等基于中国剩余定理，实现了多项式函数保密的外包计算方案，可以进行单元和多元多项式函数的外包计算，但不能实现用户的输入隐私性[13]。Zhang 等基于同态加密方案[14]，提出新的可验证多项式外包计算方案，同时实现用户输入和多项式函数的保密性[8]。但是，该方案只能计算单元多项式，没有考虑多元多项式的外包计算。目前，尚未实现多项式函数和用户输入均保密，且能有效验证计算结果的多元多项式外包计算方案。

基于多线性映射，提出了多元多项式的外包计算方案，用户可准确地验证计算结果，同时实现用户输入的保密性。方案基于 SDA 和 MSDHS 困难问题假设，在标准模型中可证明输入隐私性及可验证性。分析表明，所提方案中用户的计算量远小于直接计算多元多项式函数，且小于服务器的计算量。最终，将方案推广到多项式函数和用户输入均保密的可验证计算方案。

2 基础知识

本节介绍一些基础知识，包括多线性映射、可验证计算方案、所提方案基于的困难问题假设及安全模型。

2.1 多线性映射

输入安全参数 λ，群生成器 $G(1^{\lambda},k(n+1))$ 输出一组阶为 N，生成元为 $g_1,g_2,...,g_{k(n+1)}$ 的循环群 $G_1,G_2,\cdots,G_{k(n+1)}$，其中，$N=pq$，$p$ 和 q 为2个 λ bit 的素数。

定义多线性映射[14,15]：$e_i:G_1\times\cdots\times G_1\to G_i$，$i\in\{2,\cdots,k(n+1)\}$，对于任意 $a_1,\cdots a_i\in Z_N$，$e_i(g_1^{a_1},\cdots,g_1^{a_i})=g_i^{a_1\cdots a_i}$。所以，对于任何 $g_i^{a_i}\in G_i$，$g_j^{a_j}\in G_j$，可以计算 $e(g_i^{a_i},g_j^{a_j})=g_{i+j}^{a_ia_j}$。

2.2 困难问题假设

方案的安全性和隐私性基于 SDA 和 MSDHS 困难问题假设可证安全，现对这2个问题做简要介绍。

SDA(subgroup decision assumption)问题[8]：对于任意概率多项式时间敌手 A，如果

$$|\Pr[A(\Gamma_{k(n+1)},u)=1]-\Pr[A(\Gamma_{k(n+1)},u^q)=1]|<\varepsilon$$

称 SDA_i 假设成立，其中 $\Gamma_{k(n+1)}=(N,G_1,\cdots,G_{k(n+1)},e,g_1,\cdots,g_{k(n+1)})\leftarrow G(1^{\lambda},k(n+1))$，$u\leftarrow_R G_i$，$\lambda$ 代表安全参数，ε 代表关于 λ 的可忽略值。

对于任意 $i\in\{1,2,\cdots,k\}$，如果所有的 SDA_i 假设都成立，就称 SDA 假设成立。

$(k(n+1),d)-$MSDHS 问题：对于任意概率多项式时间敌手 A，如果

$$\Pr[A(p,q,\Gamma_{k(n+1)},g_1,g_1^{s},\cdots,g_1^{s^d})=g_{k(n+1)}^{p/s}]<\varepsilon$$

其中，$\Gamma_{k(n+1)}=(N,G_1,G_2,\cdots,G_{k(n+1)},e,g_1,g_2,\cdots,g_{k(n+1)})\leftarrow G(1^{\lambda},k(n+1))$，$s\leftarrow Z_N$。

2.3 可验证计算方案

可验证计算方案包含4个算法，即 $VC=(KeyGen,ProbGen,Compute,Verify)$ [1,2]。

$(pk,sk)\leftarrow KeyGen(1^{\lambda},f)$：输入安全参数 λ，函数 f，输出公钥 pk 和私钥 sk。

$(\sigma,\tau)\leftarrow ProbGen(sk,x)$：输入私钥 sk，函数输入 x，输出加密输入 σ 和验证密钥 τ。

$(\rho,\pi)\leftarrow Compute(pk,\sigma)$：输出公钥 pk，加密后的输入 σ，输出加密函数值 ρ 和计算正确性证明 π。

$\{f(x),\perp\}\leftarrow Verify(sk,\tau,\rho,\pi)$：输入私钥 sk、验证密钥 τ、加密函数值 ρ 和计算正确性证明 π，若验证结果正确，输出函数计算结果 $f(x)$，否则输出 $\perp$。

2.4 安全模型

本节定义可验证的 n 元多项式外包计算方案的安全模型，包括可验证性与输入隐私性。

定义 1　可验证性通过下列游戏进行，游戏有 2 个参与者：敌手 A 和挑战者 B。

游戏开始前，敌手 A 输出最终攻击的输入 $(a_1^*,a_2^*,\cdots,a_n^*)$。

Setup　B 执行 *Setup* 算法，并把公钥 pk 发给 A。

Phase1　A 对 B 进行输入的加密询问：A 发送函数输入 $(a_1,a_2,\cdots,a_n)$ 给 B，B 返回 $\sigma=(\sigma_{a_1},\sigma_{a_2},\cdots,\sigma_{a_n})$ 给 A。该询问可重复多次。

Challenge　询问结束后，B 将 $(a_1^*,a_2^*,\cdots,a_n^*)$ 的加密结果 $\sigma^*=(\sigma_{a_1}^*,\sigma_{a_2}^*,\cdots,\sigma_{a_n}^*)$ 发送给 A。A 返回 σ^* 的计算及证明结果 $(\overline{\rho},\overline{\pi_1},\overline{\pi_2},\cdots,\overline{\pi_m})$，且解密 $\overline{\rho}$ 可得 $\overline{y}\neq f(a_1^*,a_2^*,\cdots,a_m^*)$。

如果所有 t 时间的敌手经过 q 次询问后，都不能以大于 ε 的概率赢得上述游戏，则多元多项式外包计算方案是 (t,ε,q)-可验证的。

定义 2　输入隐私性通过下列游戏进行，包括 3 个参与者：敌手 A，模拟器 S，挑战者 B。

Setup　S 与 B 执行 *Setup* 算法，并把公钥 pk 发给 A。

Phase 1　A 对 S 进行输入的加密询问：A 发送函数输入 $(a_1,a_2,\cdots,a_n)$ 给 S，S 返回 $\sigma=(\sigma_{a_1},\sigma_{a_2},\cdots,\sigma_{a_n})$ 给 A。该询问可重复多次。

Challenge　询问结束后，A 选择 2 个输入 $a_0=(a_{01},a_{02},\cdots,a_{0n})$，$a_1=(a_{11},a_{12},\cdots,a_{1n})$，$S$ 选择 $i\leftarrow\{0,1,\cdots,k-1\}$，计算 $\beta_0=a_0^{2^i}$，$\beta_1=a_1^{2^i}$，并发给挑战者 B。

B 选择 $b\leftarrow\{0,1\}$，发送 $Enc(\beta_b)$ 给 S。S 计算

$$Z=(Enc(a_1),\cdots,Enc(a_1^{2^{i-1}}),Enc(\beta_b),Enc(a_0^{2^{i+1}}),\cdots,Enc(a_0^{2^{k-1}}))$$

并发送给 A。A 返回 $b'\leftarrow\{0,1\}$ 给 S。

Guess　如果 $b'=1$，S 输出 $\hat{b}=1$；否则输出 $\hat{b}=0$。如果 $\hat{b}=b$，S 赢得游戏。

在上述游戏中，定义 A 的优势为 $|\Pr[\hat{b}=b]-\frac{1}{2}|$。

如果所有 t 时间的敌手经过 q 次询问后，都不能以大于 ε 的优势赢得上述游戏，则多元多项式外包计算方案达到 (t,ε,q)-输入隐私性。

3　可验证的多元多项式外包计算方案

本节提出可验证的多元多项式外包计算方案，且能实现输入隐私性。本方案建立在 BGN 加密方案[8]基础上，因此首先简介 BGN 加密方案，然后详细介绍提出的方案。

3.1　BGN 加密算法

对于任意 $k\in Z_N$ 且 $k\geqslant 2$，$BGN_k=(Gen,Enc,Dec)$，具体算法如下。

$Gen(1^\lambda,k)$：输入安全参数 λ，群生成器 $G(1^\lambda,k)$ 输出一组阶为 N，生成元为 $g_1,g_2,\cdots,g_k$ 的循环群 $G_1,G_2,\cdots,G_k$，其中，$N=pq$，p 和 q 为 2 个 λ bit 的素数。公钥 $pk=(g_1,h)$ 和私钥 $sk=p$，其中，$h=u^q$，$u\leftarrow G_1$。

$Enc(pk,m)$：输入明文 m 和公钥 pk，输出密文 $c=g_1^m h^r\in G_1$，其中，$r\in Z_N$。

$Dec(sk,c)$：输入密文 c 和私钥 sk，计算 $c^p=(g_1^p)^m h^{rp}=g_1^{pm}u^{rpq}=g_1^{pm}$，并求离散对数问题得到明文 m。

由文献[8]可知，BGN_k 支持无限次的加法同态和 $k-1$ 次的乘法同态。因此，已知 $Enc(m_1),\cdots,Enc(m_k)$，可以计算 $Enc(m_1+\cdots+m_k)$ 和 $Enc(m_1m_2\cdots m_k)$，详细过程如下。

$$Enc(m_1+\cdots+m_k)=Enc(m_1)\cdots Enc(m_k)$$

$$Enc(m_1m_2\cdots m_k)=e_k(Enc(m_1),Enc(m_2),\cdots,Enc(m_k))$$

其中，Enc 表示 BGN_k 加密。

所以，可以使用 $Enc(m_1),\cdots,Enc(m_k)$ 计算 $Enc(f(m_1,\cdots,m_k))$，其中，$f(x_1,\cdots,x_k)$ 代表 k 元多项式。

3.2　外包计算方案

下面给出可验证的多元多项式外包计算的具体方案。假设客户端外包 n 元 d 阶多元多项式 $f(x_1,x_2,\cdots,x_n)=\sum_{i_1+i_2+\cdots+i_n\leqslant d}f_{i_1,i_2,\cdots,i_n}x_1^{i_1}x_2^{i_2},\cdots,x_n^{i_n}$ 给服务器。在本节方案中，只对用户输入加密，函数是公开的，后面会介绍如何对函数和输入都加密。

3.2.1　$KeyGen(1^\lambda,f(x_1,x_2,\cdots,x_n))$

选取 $\Gamma=(N,G_1,\cdots,G_{k(n+1)},e,g_1,\cdots,g_{k(n+1)})\leftarrow G(1^\lambda,k(n+1))$，$k=\lceil\log(d+1)\rceil$。

选取 $s=(s_1,s_2,\cdots,s_n)\leftarrow(Z_N)^n$，并计算 $t=g_1^{f(s_1,s_2,\cdots,s_n)}$，

$\sigma_{s_1}=(\sigma_{s_{11}},\sigma_{s_{12}},\cdots\sigma_{s_{1k}})=(g^{s_1},g^{s_1^2},\cdots,g^{s_1^{2^{k-1}}})$，…，

$\sigma_{s_n}=(\sigma_{s_{n1}},\sigma_{s_{n2}},\cdots\sigma_{s_{nk}})=(g^{s_n},g^{s_n^2},\cdots,g^{s_n^{2^{k-1}}})$。

选取 $u \leftarrow G_1$，不妨设 $u = g_1^{\delta}$，$\delta \in Z_N$，并计算 $h = u^q$。

输出私钥 $sk = (p,q,s,t)$，公钥 $pk = (\Gamma, g_1, h; \sigma_{s_1}, \sigma_{s_2}, \cdots, \sigma_{s_n}; f)$。

3.2.2 *ProbGen*$(sk, a_1, a_2, \cdots, a_n)$

设 $(s_1, s_2, \cdots, s_n)$ 对应的 BGN 密文为 $(\sigma_{s_1}, \sigma_{s_2}, \cdots, \sigma_{s_n})$。

假设函数输入为 $(a_1, a_2, \cdots, a_n)$，选取 $r_{ij} \leftarrow Z_N$，计算 $\sigma_{a_{ij}} = g_1^{a_i^{2^{j-1}}} h^{r_{ij}}$，$i \in \{1,2,\cdots,n\}$，$j \in \{1,2,\cdots,k\}$，$k = \lceil \log(d+1) \rceil$。

输出 $\sigma_{a_i} = (\sigma_{a_{i1}}, \sigma_{a_{i2}}, \cdots, \sigma_{a_{ik}})$，$i \in \{1,2,\cdots,n\}$，最终 $\sigma = (\sigma_{a_1}, \sigma_{a_2}, \cdots, \sigma_{a_n})$。

3.2.3 *Compute*(pk, σ)

服务器收到 $\sigma = (\sigma_{a_1}, \sigma_{a_2}, \cdots, \sigma_{a_n})$ 后，计算加密函数值 ρ 和验证值 π，并返回给用户。

1）计算加密函数值 ρ

假设 $f(x_1, x_2, \cdots, x_n) = \sum_{i_1+i_2+\cdots+i_n \leqslant d} f_{i_1,i_2,\cdots,i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$。

将指数 i_b 表示为二进制形式：$(i_{b1}, \cdots, i_{bk})$，则 $i_b = \sum_{j=1}^{k} 2^{j-1} i_{bj}$，$b \in \{1,2,\cdots,n\}$。

当 $i_{bj} = 1$ 时，$\phi_{a_{bj}} = \sigma_{a_{bj}}$；当 $i_{bj} = 0$，$\phi_{a_{bj}} = g_1$，则 $\rho_{a_b^{i_b}} = e_k(\phi_{a_{b1}}, \phi_{a_{b2}}, \cdots, \phi_{a_{bk}}) = g_k^{\mu_{a_b}} = g_k^{a_b^{i_b}} h_k^{r}$ 是 $a_b^{i_b}$ 的对应密文，其中，$\mu_{a_b} = \prod_{j=1}^{k} (a_b^{2^{j-1}} + q\delta r_{bj})^{i_{bj}}$，$r = \frac{1}{q\delta}(\mu_{a_b} - a_b^{i_b})$，$b \in \{1,2,\cdots,n\}$。

$a_1^{i_1} a_2^{i_2} \cdots a_n^{i_n}$ 的对应密文为 $\rho_{a_1^{i_1} a_2^{i_2} \cdots a_n^{i_n}} = e_n(\rho_{a_1^{i_1}}, \rho_{a_2^{i_2}}, \cdots, \rho_{a_n^{i_n}}) = g_{kn}^{\mu_{a_1}\mu_{a_2}\cdots\mu_{a_n}}$，因此 $f(a_1, a_2, \cdots, a_n) = \sum_{i_1+i_2+\cdots+i_n \leqslant d} f_{i_1,i_2,\cdots,i_n} a_1^{i_1} a_2^{i_2} \cdots a_n^{i_n}$ 的对应密文 $\rho = \prod_{i_1+i_2\cdots+i_n \leqslant d} \cdot \rho_{a_1^{i_1} a_2^{i_2} \cdots a_n^{i_n}}^{f_{i_1 i_2 \cdots i_n}}$。

2）计算验证值 π

对于 n 元 d 阶多项式 $f(x_1, x_2, \cdots, x_n)$，存在唯一的 $c_1(x_2, \cdots, x_n), \cdots, c_n(x_n)$，使下列式成立。

$$\begin{cases} f(x_1,x_2,\cdots,x_n) - f(a_1,x_2,\cdots,x_n) = (x_1 - a_1)c_1(x_1,x_2,\cdots,x_n) \\ f(a_1,x_2,\cdots,x_n) - f(a_1,a_2,\cdots,x_n) = (x_2 - a_2)c_2(x_2,\cdots,x_n) \\ \vdots \\ f(a_1,a_2,\cdots,a_{n-1},x_n) - f(a_1,a_2,\cdots,a_n) = (x_n - a_n)c_n(x_n) \end{cases}$$

把以上 n 个等式相加得

$$f(x_1,\cdots,x_n) - f(a_1,\cdots,a_n) = (x_1 - a_1)c_1(x_1,x_2,\cdots,x_n) + \cdots + (x_n - a_n)c_n(x_n)$$

设

$$c_1(s_1,\cdots,s_n) = \sum_{j_{1,1}+\cdots+j_{1,n+1} < d} c_{j_{1,1},\cdots j_{1,n+1}} s_1^{j_{1,1}} \cdots s_n^{j_{1,n}} a_1^{j_{1,n+1}}$$

$$c_2(s_2,\cdots,s_n) = \sum_{j_{2,1}+\cdots+j_{2,n+1} < d} c_{j_{2,1},\cdots j_{2,n+1}} s_2^{j_{2,1}} \cdots s_n^{j_{2,n-1}} a_1^{j_{2,n}} a_2^{j_{2,n+1}}$$

$$c_n(s_n) = \sum_{j_{n,1}+\cdots+j_{n,n+1} < d} c_{j_{n,1},\cdots,j_{n,n+1}} s_n^{j_{n,1}} a_1^{j_{n,2}} \cdots a_n^{j_{n,n+1}}$$

对于任意一个 $j_{u,v}$，$u \in \{1,\cdots,n\}$，$v \in \{1,2,\cdots,n+1\}$，可以表示为 $j_{u,v} = \sum_{t=1}^{k} j_{u,v,t} 2^{t-1}$。因此，

$$s_i^{j_{u,v}} = s_i^{j_{u,v,1}} (s_i^2)^{j_{u,v,2}} \cdots (s_i^{2^{k-1}})^{j_{u,v,k}}$$

$$a_i^{j_{u,v}} = a_i^{j_{u,v,1}} (a_i^2)^{j_{u,v,2}} \cdots (a_i^{2^{k-1}})^{j_{u,v,k}}$$

对于每个 $t \in \{1,2,\cdots,k\}$，当 $j_{u,v,t} = 1$ 时，令 $\phi_{s_{it}} = g^{s_i^{2^{t-1}}}$，$\phi_{a_{it}} = \sigma_{a_{it}} = g_1^{a_i^{2^{t-1}}} h^{r_t}$；

当 $j_{u,v,t} = 0$ 时，令 $\phi_{s_{it}} = \phi_{a_{it}} = g_1$，其中 $r_t \in Z_N$。

可以写出明文 $s_i^{j_{u,v}}$、$a_i^{j_{u,v}}$ 在 BGN_{2kn+1} 下的密文

$$\pi_{s_i}^{j_{u,v}} = e_k(\phi_{s_{i1}}, \phi_{s_{i2}}, \cdots, \phi_{s_{ik}}) = g_k^{\lambda_{u,v}}$$

$$\pi_{a_i}^{j_{u,v}} = e_k(\phi_{a_{i1}}, \phi_{a_{i2}}, \cdots, \phi_{a_{ik}}) = g_k^{\tau_{u,v}}$$

其中，

$\lambda_{u,v} = \prod_{t=1}^{k} (s_i^{2^{t-1}})^{j_{u,v,t}}$，$\tau_{u,v} = \prod_{t=1}^{k} (a_i^{2^{t-1}} + q\delta r_t)^{j_{u,v,t}}$

因此，$c_1(s_1, s_2, \cdots, s_n)$ 的对应密文

$$\pi_1 = \prod_{j_{1,1}+\cdots+j_{1,n+1} < d} (e_{n+1}(\pi_{s_1}^{j_{1,1}}, \cdots, \pi_{s_1}^{j_{1,n}}, \pi_{a_1}^{j_{1,n+1}}))^{c_{j_{1,1}\cdots j_{1,n+1}}}$$

$$= \prod_{j_{1,1}+\cdots+j_{1,n+1} < d} g_{k(n+1)}^{\lambda_{1,1}\cdots\lambda_{1,n}\tau_{1,n+1} c_{j_{1,1}\cdots j_{1,n+1}}}$$

$c_2(s_2, \cdots, s_n)$ 的对应密文

$$\pi_2 = \prod_{j_{2,1}+\cdots+j_{2,n+1} < d} g_{k(n+1)}^{\lambda_{2,1}\cdots\lambda_{2,n-1}\tau_{2,n}\tau_{2,n+1} c_{j_{2,1}\cdots j_{2,n+1}}}$$

$c_n(s_n)$ 的对应密文

$$\pi_n = \prod_{j_{n,1}+\cdots+j_{n,n+1} < d} g_{k(n+1)}^{\lambda_{n,1}\tau_{n,2}\cdots\tau_{n,n+1} c_{j_{n,1}\cdots j_{n,n+1}}}$$

3.2.4 *Verify*(sk, ρ, π)

用户使用 $\rho^p = (g_{kn}^p)^y$ 求得函数值 $y = f(a_1, a_2, \cdots, a_n)$，并进行以下验证

$$e(t / g_1^{\,y}, g_{k(n+1)}{}^{p}) = e(g_1^{s_1} / g_1^{a_1}, \pi_1^{\,p})$$

$$e(g_1^{s_2} / g_1^{a_2}, \pi_2^{\,p}) \cdots e(g_1^{s_n} / g_1^{a_n}, \pi_n^{\,p}) \qquad (1)$$

如果以上验证式成立输出 y，否则输出 $\perp$。

4　方案分析

本节对所提外包计算方案进行分析，包括正确性、可验证性和输入隐私性，最后对方案进行了性能分析。

4.1　正确性

通过以下引理证明方案的正确性，即只要服务器是诚实的，通过上述方案总能输出正确的函数值 $y=f(a_1,a_2,\cdots,a_n)$ 且式(1)成立。

引理 1　如果服务器是诚实的，则 $y=f(a_1,a_2,\cdots,a_n)$ 且式(1)成立。

证明　由于 $\mu_{a_b}=\prod_{j=1}^{k}(a_b^{2^{j-1}}+q\sigma r_j)^{i_{bj}}$，

$$p\mu_{a_1}\mu_{a_2}\cdots\mu_{a_n}\equiv pa_1^{\sum_{j=1}^{k}i_{1j}\cdot 2^{j-1}}\cdots a_n^{\sum_{j=1}^{k}i_{nj}\cdot 2^{j-1}}$$

$$\equiv pa_1^{i_1}a_2^{i_2}\cdots a_n^{i_n}\pmod N$$

$$\rho^p=\prod_{i_1+\cdots+i_n\le d}\rho_{a_1^{i_1}\cdots a_n^{i_n}}^{pf_{i_1\cdots i_n}}=\prod_{i_1+\cdots+i_n\le d}g_{kn}^{p\mu_{a_1}\cdots\mu_{a_n}f_{i_1\cdots i_n}}$$

$$=\prod_{i_1+\cdots+i_n\le d}g_{kn}^{pf_{i_1\cdots i_n}a_1^{i_1}\cdots a_n^{i_n}}=(g_{kn}^{\ p})^{f(a_1,\cdots,a_n)}$$

因此，$y=f(a_1,a_2,\cdots,a_n)$ 成立。

由于 $\lambda_{u,v}=\prod_{t=1}^{k}(s_i^{2^{t-1}})^{j_{u,v,t}}$，$\tau_{u,v}=\prod_{t=1}^{k}(a_i^{2^{t-1}}+q\delta_t)^{j_{u,v,t}}$，因此，$p\lambda_{1,1}\cdots\lambda_{1,n}\tau_{1,n+1}\equiv ps_1^{j_{1,1}}\cdots s_n^{j_{1,n}}a_1^{j_{1,n+1}}\pmod N$。

$$\pi_1^{\ p}=\prod_{j_{1,1}+\cdots+j_{1,n+1}<d}(\pi_{s_1}^{j_{1,1}}\pi_{s_2}^{j_{1,2}}\cdots\pi_{s_n}^{j_{1,n}}\pi_{a_1}^{j_{1,n+1}})^{pc_{j_{1,1}\cdots j_{1,n+1}}}$$

$$=\prod_{j_{1,1}+\cdots+j_{1,n+1}<d}g_{k(n+1)}^{p\lambda_{1,1}\cdots\lambda_{1,n}\tau_{1,n+1}c_{j_{1,1}\cdots j_{1,n+1}}}$$

$$=\prod_{j_{1,1}+\cdots+j_{1,n+1}<d}g_{k(n+1)}^{ps_1^{j_{1,1}}\cdots s_n^{j_{1,n}}a_1^{j_{1,n+1}}c_{j_{1,1}\cdots j_{1,n+1}}}$$

$$=(g_{k(n+1)}^{\ p})^{c_1(s_1,s_2,\cdots,s_n)}$$

同理，$\pi_2^{\ p}=(g_{k(n+1)}^{\ p})^{c_2(s_2,\cdots,s_n)}$，…，

$\pi_n^{\ p}=(g_{k(n+1)}^{\ p})^{c_n(s_n)}$。

$$e(\frac{g_1^{s_1}}{g_1^{a_1}},\pi_1^{\ p})e(\frac{g_1^{s_2}}{g_1^{a_2}},\pi_2^{\ p})\cdots e(\frac{g_1^{s_n}}{g_1^{a_n}},\pi_n^{\ p})$$

$$=e(g_1^{s_1-a_1},g_{k(n+1)}^{\ p})^{c_1(s_1,s_2,\cdots,s_n)}$$

$$e(g_1^{s_2-a_n},g_{k+1}^{\ p})^{c_2(s_2,\cdots,s_n)}\cdots$$

$$e(g_1^{s_n-a_n},g_{k(n+1)}^{\ p})^{c_n(s_n)}$$

$$=e(g_1,g_{k(n+1)}^{\ p})^{(s_1-a_1)c_1(s_1,\cdots s_n)+\cdots+(s_n-a_n)c_n(a_n)}$$

$$=e(g_1,g_{k(n+1)}^{\ p})^{f(s_1,s_2,\cdots,s_n)-f(a_1,a_2,\cdots,a_n)}$$

$$=(\frac{t}{g_1^{\ y}},g_{k(n+1)}^{\ p})$$

因此，只要服务器是诚实的，则式(1)成立。

4.2　可验证性

通过以下引理证明方案的可验证性，即不可信的服务器不能迫使用户接受 $y\ne f(a_1,a_2,\cdots,a_n)$ 和一个错误的证明。方案可验证性基于 $(k(n+1),d)-$MSDHS 假设。

引理 2　假设 $(k(n+1),d)-$MSDHS 假设成立，方案在选择输入模型下是可验证的。

证明　假设存在 PPT 敌手 A 能以概率 ε 欺骗用户接受不正确的计算结果，可以构建算法 B 以概率 ε 解决 $(k(n+1),d)-$MSDHS 问题。游戏开始前，给定 B 一个向量 $(p,q,g_1,g_1^{\ s},\cdots,g_1^{\ s^d})$，其中 $s\leftarrow Z_N$，B 需要计算 $g_{k(n+1)}^{\ p/s}$。

游戏开始前，敌手 A 输出最终攻击的输入 $(a_1^*,a_2^*,\cdots,a_n^*)$。

然后，模拟器 B 生成系统参数如下：随机选择 n 元 d 阶多元多项式 $f(x_1,x_2,\cdots,x_n)\in Z_N[x]$，$\Gamma=(N,G_1,\cdots,G_{k(n+1)},e,g_1,\cdots,g_{k(n+1)})\leftarrow G(1^\lambda,k(n+1))$，$k=\lceil\log(d+1)\rceil$，$g_1\leftarrow G_1$。

令 $s_i=s+a_i^*,i\in\{1,2,\cdots,n\}$，计算 $t=g_1^{f(s_1,s_2,\ldots,s_n)}$，$\sigma_{s_1}=(\sigma_{s_{11}},\sigma_{s_{12}},\cdots\sigma_{s_{1k}})=(g^{s_1},g^{s_1^{\ 2}},\cdots,g^{s_1^{\ 2^{k-1}}})$，…，$\sigma_{s_n}=(\sigma_{s_{n1}},\sigma_{s_{n2}},\cdots,\sigma_{s_{nk}})=(g^{s_n},g^{s_n^{\ 2}},\cdots,g^{s_n^{\ 2^{k-1}}})$。

由于 $(g_1,g_1^{\ s},\cdots,g_1^{\ s^d})$ 及 $(a_1^*,a_2^*,\cdots,a_n^*)$ 是已知的，因此 B 能计算上述结果。

选取 $u\leftarrow G_1$，并计算 $h=u^q$，将公钥 $pk=(\Gamma,g_1,h;\sigma_{s_1},\sigma_{s_2},\cdots,\sigma_{s_n};f)$ 发送给敌手 A。

接下来，A 对 B 进行输入的加密询问。A 发送函数输入 $(a_1,a_2,\cdots,a_n)$ 给 B，B 选取 $r_j\leftarrow Z_N$，计算 $\sigma_{a_{bj}}=g_1^{a_b^{2^{j-1}}}h^{r_j}$，$j\in\{1,2,\cdots,k\}$，$k=\lceil\log(d+1)\rceil$。

输出 $\sigma_{a_b}=(\sigma_{a_{b1}},\sigma_{a_{b2}},\cdots,\sigma_{a_{bk}})$，$b\in\{1,2,\cdots,n\}$，最终 B 发送 $\sigma=(\sigma_{a_1},\sigma_{a_2},\cdots,\sigma_{a_n})$ 给 A。

询问结束后，B 将 $(a_1^*,a_2^*,\cdots,a_n^*)$ 的加密结果 $\sigma^*=(\sigma_{a_1}^{\ *},\sigma_{a_2}^{\ *},\cdots,\sigma_{a_n}^{\ *})$ 发送给 A。A 返回 σ^* 的计算及证明结果 $(\overline{\rho},\overline{\pi_1},\overline{\pi_2},\cdots,\overline{\pi_n})$，且解密 $\overline{\rho}$ 可得 $\overline{y}\ne f(a_1^*,a_2^*,\cdots,a_n^*)$。由引理 1，解密 $\overline{\rho}$ 可得 $y^*=f(a_1^*,a_2^*,\cdots,a_n^*)$。如果这 2 个结果都可以通过验证，即

$$e(\frac{t}{g_1^{\ \overline{y}}},g_{k(n+1)}^{\ p})=e(\frac{g_1^{s_1}}{g_1^{a_1^*}},\overline{\pi}_1^{\ p})$$

$$e(\frac{g_1^{s_2}}{g_1^{a_2^*}},\overline{\pi}_2^{\ p})\cdots e(\frac{g_1^{s_n}}{g_1^{a_n^*}},\overline{\pi}_n^{\ p})$$

$$e(\frac{t}{g_1^{y^*}}, g_{k(n+1)}{}^p) = e(\frac{g_1^{s_1}}{g_1^{a_1^*}}, (\pi_1^*)^p)$$
$$e(\frac{g_1^{s_2}}{g_1^{a_2^*}}, (\pi_2^*)^p) \cdots e(\frac{g_1^{s_n}}{g_1^{a_n^*}}, (\pi_n^*)^p)$$

由于 $s_i = s + a_i^*, i \in \{1,2,\cdots,n\}$，因此上述 2 式也可写成

$$e(\frac{t}{g_1^{\bar{y}}}, g_{k(n+1)}{}^p) = e(g_1^s, \overline{\pi_1}^p) \cdot e(g_1^s, \overline{\pi_2}^p) \cdots e(g_1^s, \overline{\pi_n}^p) \tag{2}$$

$$e(\frac{t}{g_1^{y^*}}, g_{k(n+1)}{}^p) = e(g_1^s, (\pi_1^*)^p) e(g_1^s, (\pi_2^*)^p) \cdots e(g_1^s, (\pi_n^*)^p) \tag{3}$$

式(2)、式(3)进行相除，可得

$$e(g_1^{y^*-\bar{y}}, g_{k(n+1)}{}^p) = e(g_1^s, (\frac{\overline{\pi_1}}{\pi_1^*})^p) e(g_1^s, (\frac{\overline{\pi_2}}{\pi_2^*})^p) \cdots e(g_1^s, (\frac{\overline{\pi_n}}{\pi_n^*})^p)$$

则

$$e(g_1, g_{k(n+1)}{}^{p(y^*-\bar{y})}) = e(g_1, (\frac{\overline{\pi_1}}{\pi_1^*})^{sp}) e(g_1, (\frac{\overline{\pi_2}}{\pi_2^*})^{sp}) \cdots e(g_1, (\frac{\overline{\pi_n}}{\pi_n^*})^{sp})$$

所以，$g_{k(n+1)}{}^{p/s} = \left[\left(\frac{\overline{\pi_1}}{\pi_1^*}\right)\left(\frac{\overline{\pi_2}}{\pi_2^*}\right)\cdots\left(\frac{\overline{\pi_n}}{\pi_n^*}\right)\right]^{\frac{p}{y^*-\bar{y}}}$，$B$ 以概率 ε 解决 $(k(n+1),d)-$MSDHS 问题。

4.3 输入隐私性

通过以下引理证明方案的输入隐私性，即不可信服务器不能区分用户的 2 个不同的输入。方案的输入隐私性基于 SDA 假设。

引理 3 如果 SDA 假设成立，方案能达到输入隐私性。

证明 对于任何输入 $(a_1,a_2,\cdots,a_n)$，用户发给服务器的唯一包含输入的信息是 $\sigma = (\sigma_{a_1}, \sigma_{a_2}, \cdots, \sigma_{a_n})$。假设敌手 A 能够以优势 ε 攻击方案的隐私性，即以概率 $\Pr[A] > \frac{1}{2} + \varepsilon$ 区分 2 个不同的输入，可以构造算法 S 以概率 ε/k 区分 2 个不同明文对应的密文。在方案的安全性证明游戏中，模拟器 B 存储私钥 p，将公钥 $pk = (\Gamma, g_1, h)$ 发给 S。然后，S 与敌手 A 执行如下操作。

S 生成系统参数：随机选择 n 元 d 阶多元多项式 $f(x_1,x_2,\cdots,x_n) \in Z_N[x]$，

$\Gamma = (N, G_1, \cdots, G_{k(n+1)}, e, g_1, \cdots, g_{k(n+1)}) \leftarrow G(1^\lambda, k(n+1))$，

$k = \lceil \log(d+1) \rceil$，$g_1 \leftarrow G_1$。

随机选择 $s = (s_1, s_2, \cdots, s_n) \leftarrow (Z_N)^n$，计算

$t = g_1^{f(s_1, s_2, \ldots, s_n)}$

$\sigma_{s_1} = (\sigma_{s_{11}}, \sigma_{s_{12}}, \cdots, \sigma_{s_{1k}}) = (g^{s_1}, g^{s_1{}^2}, \cdots, g^{s_1{}^{2^{k-1}}})$

…

$\sigma_{s_n} = (\sigma_{s_{n1}}, \sigma_{s_{n2}}, \cdots, \sigma_{s_{nk}}) = (g^{s_n}, g^{s_n{}^2}, \cdots, g^{s_n{}^{2^{k-1}}})$

将公钥 $pk = (\Gamma, g_1, h; \sigma_{s_1}, \sigma_{s_2}, \cdots, \sigma_{s_n}; f)$ 发送给敌手 A。

接下来，A 对 S 进行输入的加密询问。A 发送输入 $(a_1,a_2,\cdots,a_n)$ 给 S，S 选取 $r_j \leftarrow Z_N$，计算 $\sigma_{a_{bj}} = g_1^{a_b^{2^{j-1}}} h^{r_j}$，$j \in \{1,2,\cdots,k\}$，$k = \lceil \log(d+1) \rceil$。

输出 $\sigma_{a_b} = (\sigma_{a_{b1}}, \sigma_{a_{b2}}, \cdots, \sigma_{a_{bk}})$，$b \in \{1,2,\cdots,n\}$，最终 B 发送 $\sigma = (\sigma_{a_1}, \sigma_{a_2}, \cdots, \sigma_{a_n})$ 给 A。

询问结束后，A 选择 2 个输入 $a_0 = (a_{01}, a_{02}, \cdots, a_{0n})$，$a_1 = (a_{11}, a_{12}, \cdots, a_{1n})$，$S$ 选择 $i \leftarrow \{0,1,\cdots,k-1\}$，计算 $\beta_0 = a_0{}^{2^i} = (a_{01}^{2^i}, a_{02}^{2^i}, \cdots, a_{0n}^{2^i})$，$\beta_1 = a_1{}^{2^i} = (a_{11}^{2^i}, a_{12}^{2^i}, \cdots, a_{1n}^{2^i})$，并发给模拟器 B。

B 选择 $b \leftarrow \{0,1\}$，发送 $Enc(\beta_b)$ 给 S。S 计算

$$Z = (Enc(a_1), \cdots, Enc(a_1^{2^{i-1}}), Enc(\beta_b), Enc(a_0^{2^{i+1}}), \cdots, Enc(a_0^{2^{k-1}}))$$

并发送给 A。A 返回 $b' \leftarrow \{0,1\}$ 给 S。

如果 $b'=1$，S 输出 $\hat{b}=1$；否则输出 $\hat{b}=0$。如果 $\hat{b}=b$，S 赢得游戏。

对于 $j \in \{0,1,\cdots,k\}$，定义：$Z_j = (Enc(a_1), \cdots, Enc(a_1^{2^{j-1}}), Enc(a_0^{2^j}), \cdots, Enc(a_0^{2^{k-1}}))$，

则 $Z_0 = (Enc(a_0), Enc(a_0^2), \cdots, Enc(a_0^{2^{k-1}}))$，$Z_k = (Enc(a_1), Enc(a_1^2), \cdots, Enc(a_1^{2^{k-1}}))$。

由 $\Pr[A] > \frac{1}{2} + \varepsilon$ 可以推出

$$|\Pr[A(pk, Z_k) = 1] - \Pr[A(pk, Z) = 1]|$$
$$= |\Pr[A(pk, Z_k) = 1] - \frac{1}{2} + \frac{1}{2} - \Pr[A(pk, Z_0) = 1]|$$
$$\geqslant \varepsilon + \varepsilon = 2\varepsilon$$

当 $b=0$ 时，$Z = Z_i$；当 $b=1$ 时，$Z = Z_{i+1}$。因此

$$\Pr[b'=b] = |\sum_{l=0}^{k-1} \Pr[b'=b \mid i=l] \Pr[i=l]|$$
$$= \frac{1}{k} |\sum_{l=0}^{k-1} (\Pr[b'=b \mid i=l, b=0] \Pr[b=0] +$$

$$\Pr[b'=b \mid i=l,b=1]\Pr[b=1])\mid$$
$$=\frac{1}{2k}\mid\sum_{l=0}^{k-1}(\Pr[b'=b \mid i=l,b=0]+$$
$$\Pr[b'=b \mid i=l,b=1])\mid$$
$$=\frac{1}{2k}\mid\sum_{l=0}^{k-1}(1-\Pr[A(pk,Z_l)=1]+\Pr[A(pk,Z_{l+1})=1])\mid$$
$$=\frac{1}{2k}\mid(k+\Pr[A(pk,Z_k)=1]-\Pr[A(pk,Z_0)=1])\mid$$
$$=\mid\frac{1}{2}+\frac{1}{2k}(\Pr[A(pk,Z_k)=1]-\Pr[A(pk,Z_0)=1])\mid$$
$$\geqslant\frac{1}{2}+\frac{\varepsilon}{k}$$

定理 1　若 $(k(n+1),d)-$MSDHS 假设和 SDA 假设成立，所提方案是满足输入隐私性的安全的 VC 方案。

证明　由引理 1~引理 3 可证。

函数保密性：按照文献[8]中的方法，可在所提方案的基础上实现多项式函数和用户输入均保密的外包计算方案。假设 $f(x_1,x_2,\cdots,x_n)=\sum_{i_1+i_2+\cdots+i_n\leqslant d} f_{i_1,i_2,\cdots,i_n}x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}$，用户将函数加密结果 $Enc(f)=\{Enc(f_{i_1,i_2,\cdots,i_n})\mid i_1+i_2+\cdots+i_n\leqslant d\}$ 和输入 $(a_1,a_2,\cdots,a_n)$ 的加密结果 $\sigma=(\sigma_{a_1},\sigma_{a_2},...,\sigma_{a_n})$ 发给服务器，然后服务器可根据 BGN 方案，使用 $Enc(f)$ 和 σ 计算 $f(a_1,a_2,\cdots,a_n)$ 的加密结果 $\rho=Enc(f(a_1,a_2,\cdots,a_n))$ 和正确性证明 $\pi_1=Enc(c_1(s_1,\cdots,s_n))$，$\pi_2=c_2(s_2,\cdots,s_n),\cdots$，$\pi_n=c_n(s_n)$。参考定理 1，可证明新方案的安全性、输入和函数隐私性。

4.4　性能分析

在多项式 $f(x_1,x_2,\cdots,x_n)=\sum_{i_1+i_2+\cdots+i_n\leqslant d} f_{i_1,i_2,\cdots,i_n}x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}$ 中，$i_1,i_2,\cdots,i_n\in[0,d]$，因此多项式最多有 d^n+n 项。若用户直接计算，每项需进行 $O(n)$ 次指数运算，总共需要进行 $O(d^n n)$ 次指数运算。下面分析外包计算方案中用户及服务器的计算量。

由表 1 可知，为了保证输入的隐私性，用户在 *ProbGen* 阶段需进行 $O(nk)$ 次指数运算，在 *Verify* 阶段需进行 $O(n)$ 次指数运算和对运算；而为了计算 $(\rho,\pi_1,\cdots,\pi_n)$，服务器在 *Compute* 阶段须进行 $O(n^2d)$ 次指数运算和 $O(n^2d)$ 次对运算。综上，用户的计算量远小于服务器的计算代价，也小于直接计算多元多项式。

4.5　仿真实现

本节对所提多元多项式外包计算方案进行仿真实现。用户和服务器分别使用 Intel Celeron Processor (1.6 GHz，4 GB 内存)和 Intel i7 Processor(3.0 GHz，8 GB 内存）模拟。

表 1　　外包计算方案计算复杂度

运算参与方	运算	用户	服务器
ProbGen 阶段	指数运算	$O(nk)$	0
Compute 阶段	对运算	0	$O(n^2d)$
	指数运算	0	$O(n^2d)$
Verify 阶段	指数运算	$O(n)$	0
	对运算	$O(n)$	0

注：n、d 分别代表多元多项式函数的元数和次数，$k=\lceil\log(d+1)\rceil$

在图 2 和图 3 中，对于三元 63 次多项式和三元 127 次多项式，给出了用户直接计算和外包计算多项式的时间。其中横坐标代表合数群阶 N (单位：bit)，纵坐标代表计算时间(单位：ms)。可以看出，用户外包计算时间远小于直接计算多元多项式时间。

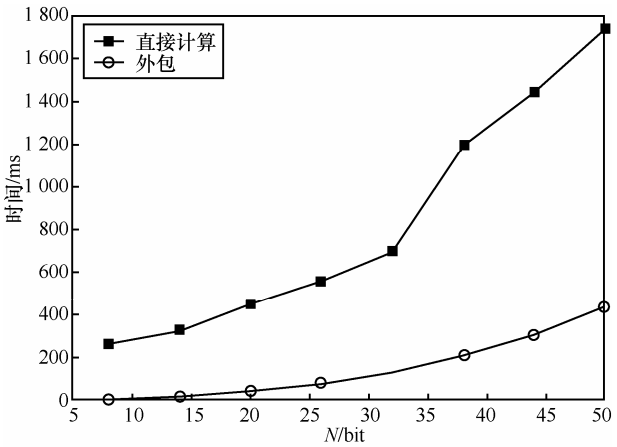

图 2　三元 63 次多项式外包计算（$k=6,d=2^k-1=63$）

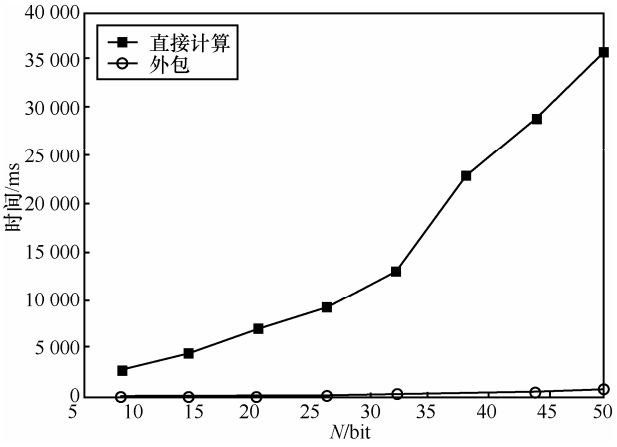

图 3　三元 127 次多项式外包计算（$k=7,d=2^k-1=127$）

5 结束语

基于多线性映射和 BGN 加密方案，提出了一个可验证的多元多项式外包计算方案，用户将加密后的多项式函数和函数输入发给服务器，服务器返回加密后的计算结果，用户解密后可验证计算结果的正确性，且验证的计算量远小于直接计算多元多项式函数。基于 MSDHS 假设和 SDA 假设，方案在标准模型中达到可验证性和输入隐私性。

参考文献：

[1] GENNARO R, GENTRY C, PARNO B. Non-interactive verifiable computing: outsourcing computation to untrusted workers[A]. CRYPTO 2010[C]. 2010.465-482.

[2] CHUNG K M, KALAI Y, VADHAN S P. Improved delegation of computation using fully homomorphic encryption[A]. CRYPTO 2010[C]. 2010. 483-501.

[3] APPLEBAUM B, ISHAI Y, KUSHILEVITZ E. From secrecy to soundness: efficient verification via secure computation[A]. ICALP 2010[C]. 2010. 152-163.

[4] PARNO B, RAYKOVA M, VAIKUNTANATHAN V. How to delegate and verify in public: verifiable computation from attribute-based encryption[A]. TCC 2012, LNCS 7194[C]. 2012. 422-439.

[5] CHOI S, KATZ J, KUMARESAN R, CID C. Multi-client non-interactive verifiable computation[A]. TCC 2013, LNCS 7785[C]. 2013. 499-518.

[6] CHEN X, LI J, MA J, TANG Q, LOU W. New algorithms for secure outsourcing of modular exponentiations[A]. ESORICS 2012, LNCS 7459[C]. 2012. 541-556.

[7] HOHENBERGER S, LYSYANSKAYA A, How to securely outsource cryptographic computations[A]. TCC 2005. LNCS 3378[C]. 2005. 264-282.

[8] ZHANG L, NAINI R S. Private outsourcing of polynomial evaluation and matrix multiplication using multilinear maps[A]. CANS 2013, LNCS 8257[C]. 2013. 329-348.

[9] GREEN M, HOHENBERGER S, WATERS B. Outsourcing the decryption of ABE ciphertexts[EB/OL]. http://static.usenix.org/events/sec11/tech/ full-papers/Green.pdf

[10] LAI J, DENG R H, GUAN C, WENG J. Attribute-based encryption with verifiable outsourced decryption[J]. IEEE Transactions on Information Forensics and Security, 2013, 8(8): 1343-1354.

[11] BENABBAS S, GENNARO R, VAHLIS Y. Verifiable delegation of computation over large datasets[A]. CRYPTO 2011, LNCS 6841[C]. 2011.111-131.

[12] FIORE D, GENNARO R. Publicly verifiable delegation of large polynomials and matrix computations with applications[A]. ACM CCS 2012[C]. 2012. 501-512.

[13] MA X., ZHANG F, LI J. Verifiable evaluation of private polynomials[A]. The Fourth International Conference on Emerging Intelligent Data and Web Technologies[C]. IEEE CPS, 2013. 451-458.

[14] GARG S, GENTRY C, HALEVI S. Candidate multilinear maps from ideal lattices[A]. EUROCRYPT 2013, LNCS 7881[C]. 2013. 1-17.

[15] GARG S, GENTRY C, HALEVI S, SAHAI A, WATERS B. Attribute-based encryption for circuits from multilinear maps[A]. CRYPTO 2013, LNCS 8043[C]. 2013. 479-499.

作者简介：

任艳丽（1982-），女，山西运城人，博士，上海大学副研究员、硕士生导师，主要研究方向为公钥密码学、可验证外包计算、网络安全协议等。

谷大武（1970-），男，河南漯河人，博士，上海交通大学教授、博士生导师，主要研究方向为密码分析与设计、信息分析与密码工程、计算机安全体系结构等。

蔡建兴（1991-），男，浙江温州人，上海大学硕士生，主要研究方向为公钥密码学、可验证外包计算等。

黄春水（1991-），男，湖北黄冈人，上海大学硕士生，主要研究方向为公钥密码学、可验证外包计算等。