计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2017.20170463
54(10):2215-2223,2017

一个单服务器辅助的高效 n 取 k 茫然传输协议

赵圣楠[1] 蒋 瀚[1] 魏晓超[2] 柯俊明[1] 赵明昊[1]
[1](山东大学计算机科学与技术学院 济南 250101)
[2](山东师范大学信息科学与工程学院 济南 250358)
(yucheng_zhao@163.com)

An Efficient Single Server-Aided k-out-of-n Oblivious Transfer Protocol

Zhao Shengnan[1], Jiang Han[1], Wei Xiaochao[2], Ke Junming[1], and Zhao Minghao[1]
[1](*School of Computer Science and Technology, Shandong University, Jinan* 250101)
[2](*School of Information Science and Engineering, Shandong Normal University, Jinan* 250358)

Abstract Oblivious transfer (OT) is a cryptographic primitive used for choice information hiding for the receiver. As a basic tool for high-level multi-party cryptographic protocol construction, it plays an important role in numerous specific applications. In the k-out-of-n OT(OT_n^k), the receiver acquires k selections among the n choice in an oblivious manner. Generally, the construction of the OT_n^k involves lots of group exponential operations, which brings a heavy burden for embedded devices with limited computational capabilities. With the proliferation of cloud computing, it is feasible to implement complex cryptographic primitives with the support of powerful computing recourse and high-speed dedicated network provided by the cloud service provider (CSP). In this paper, we propose a service-assisted k-out-of-n OT protocol in single server architecture, which outsources the vast majority of exponentiation operations to the cloud. This scheme is constructed with secret sharing and other fundamental public-key primitives, and it achieves provable security on none-collusion semi-honest model under the decisional Diffie-Hellamn (DDH) hard problem; meanwhile it ensures data privacy against the cloud server. Besides, a detailed description of scheme construction and security proof is presented in the context. As a basic cryptographic primitive in cloud environment, the single server-added oblivious transfer protocol will play an important role in designs of general cloud-assisted multi-party computation protocol as well as developments of secure and efficient cloud service software.

Key words oblivious transfer (OT); outsourcing computing; decisional Diffie-Hellamn (DDH) assumption; semi-honest model; secure multi-party computation

摘 要 茫然传输(oblivious transfer, OT)是一种用于隐藏数据接收者选择信息的密码学原语,作为构建高层多方密码协议的基本工具,在诸多具体问题中都有着重要应用.在 k-out-of-n OT(OT_n^k)中,接收者能够以茫然的方式,在 n 个数据中有选择地取得其中的 k 个.通常 OT_n^k 的构造需要大量的群指数操作,

收稿日期:2017-06-11;**修回日期**:2017-07-29
基金项目:国家自然科学基金项目(61572294);国家自然科学基金青年科学基金项目(61602287);国家自然科学基金重点项目(61632020);山东大学基本科研业务费专项资金项目(2017JC019)
This work was supported by the National Natural Science Foundation of China (61572294), the National Natural Science Foundation of China for Young Scientists (61602287), the Key Program of the National Natural Science Foundation of China (61632020), and the Fundamental Research Funds of Shandong University (2017JC019).
通信作者:蒋瀚(jianghan@sdu.edu.cn)

对于计算能力受限的嵌入式设备而言依然是极大的负担.随着云计算的发展,可以利用云服务提供商的计算能力和高速专属网络来辅助复杂密码原语的实现.在此提出了一个高效单服务器辅助的 n 取 k 茫然传输协议,将主要群指数操作外包给云服务器来实现.该方案利用秘密分享等基础密码学原语构建,其安全性基于判定性 Diffie-Hellamn(decisional Diffie-Hellamn, DDH)困难问题,在非合谋半诚实模型下可证明安全,同时可以保证云服务器的数据隐私性,给出该方案的具体描述及其详细的安全性证明.作为云环境下的一种基础密码学原语,所提出的云服务器辅助的茫然传输协议,在设计云辅助的通用安全计算协议及构建高效安全云服务应用软件等方面将起到重要作用.

关键词 茫然传输;外包计算;判定性 Diffie-Hellamn 假设;半诚实模型;安全多方计算

中图法分类号 TP301

外包计算作为当今新型的计算模式,日渐受到研究者和企业界的关注.物联网、移动计算、无线传感器网络等技术的兴起和普及极大地方便了我们的生活、沟通与信息获取,但是出于效率和成本等因素的考虑,这类设备通常采用轻量级硬件架构,计算和存储能力受到极大的限制.

云计算的兴起为能力受限的移动设备提供了完美的后端支持.当前的服务计算架构中,一方面云服务器为移动设备提供视频分发、数据分析以及广告推送等服务;另一方面移动设备会将复杂的计算任务外包到云端,云端完成计算任务后将计算结果返回给移动设备.云计算和外包计算模式,极大地丰富了移动计算的内容[1].

出于安全和隐私性考虑,移动设备(如手机客户端、RFID 芯片、传感器网络节点等)之间往往需要执行认证和加密等多种密码学原语,其中以公钥密码原语居多.在公钥密码中通常需要涉及大量的群上的指数运算,这对于移动设备来说是极大的负担.目前虽然可以设计专用的嵌入式芯片来高速实现密码原语操作[2-3],但这一定程度上会增加设备成本.因此,目前一个很理想的解决方案是将公钥加密方案的部分复杂运算外包到云端来进行.

茫然传输(oblivious transfer, OT)是密码学中一个基础工具.传统的茫然传输协议中有 2 个参与方:发送方 S 和接收方 R,协议的目的是让接收方从发送方的输入中获取自己选择的输入,且满足:1)发送方不知道接收方的选择;2)接收方只能获得自己选择的输入且无法获得额外的输入信息.茫然传输在隐私保护的数据查询、安全多方计算等多个领域有重要应用[4].

传统的 OT 协议通常利用公钥密码原语来构建,尤其在 k-out-of-n OT (OT_n^k)中需要大量的群指数运算.因而在理想情况下,我们希望利用云服务器的计算能力和专属网络来提高该种密码原语的客户端执行效率.

1 相关工作和主要结论

1.1 相关工作

茫然传输协议 Rabin[5] 在 1981 年首次提出了茫然传输的概念并给出雏形.该协议的安全性在于保证发送者不能获知接受者是否收到了他所发送的消息;Even 等人在文献[6]中给出 OT 的一种改进形式,即发送者有 2 条消息,接收者有选择性地获取接收其中的一条,其安全性保证发送者不可获知接收者的选择.这一模式被后续研究者所采纳,成为现行的 1-out-of-2 OT(OT_2^1);Brassard 等人[7] 提出了更具一般性的 1-out-of-n OT(OT_n^1); Stern[8],Cramer 等人[9] 和 Kalai[10] 分别给出了不同的 OT_n^1 构造框架;文献[11-12]均提出不同的单边模拟的唯隐私保障(privacy only)的高效 OT 方案;魏晓超等人[13] 首次在标准恶意敌手模型下,基于判定性 Diffie-Hellamn(decisional Diffie-Hellamn, DDH)假设构造了一个高效、可完全模拟的 OT_n^1 协议;Lindell[14] 将 cut-and-choose 技术运用到 OT 协议的构造中去,首次实现了全模拟等 OT 协议;Peikert 等人[15] 提出双模式加密的概念,并利用此技术给出 UC 框架下 OT 构造;冯涛等人[16] 利用可否认加密体制和可验证平滑投影 Hash 函数,对 UC 安全的 OT 协议作出了效率优化;UC 安全的 OT 协议方还有文献[17-18]等;k-out-of-n OT(OT_n^k)是 1-out-of-n 的一个拓展,接收者可以以茫然的模式从 n 个数据中获取其中的 k 个,OT_n^k 构造方案主要有文献[19-22]等;OT 拓展(OT extension)是一种以混合处理的方式用少量 OT 操作来实现大量 OT 功能的密码技术,关于 OT 拓展的研究主要有文献[23-26]等.

双向 OT 使得每个参与方都兼有发送者和接受者的双重身份,该原语可以大大降低基于 cut-and-choose 的安全多方计算协议的轮复杂度. Zhao 等人[27]和 Wei 等人[28]在双向 OT 领域做出了创举性的研究;抗量子 OT 有文献[29-30]等.

密码学原语的部分外包. 通常来讲群指数运算是密码算法中最为耗时的运算. Dijk 等人[31]首先考虑将指数运算外包给一个不可信的服务器;Ma 等人[32]考虑了将指数操作外包到 2 个非合谋的不可信服务器;Hohenberger 等人[33]和 Chen 等人[34]将变基和变幂的指数算法外包到(双服务器模型下的)单一恶意服务器,即要求 2 个服务器中仅有一个服务器为恶意的,且不与另一个诚实服务器合谋. 文献[34]同时考虑了安全与高效的并发指数(simultaneous exponentiations)操作外包;Wang 等人[35]实现了一个单一非可信服务器模型下的指数运算外包方案;Chevalier 等人[36]对该方案做了细致的密码学分析,并给出了一个优化的构造方案;Tsang 等人[37]首先考虑了以批量的方式将椭圆曲线上的点对运算(pairing)外包到云服务器上来实现;Canard 等人[38]给出了一个更为高效的安全点对运算外包方案;Xu 等人[39]考虑将复杂运算外包到计算服务器 P,并将外包计算结果的正确性验证外包到验证服务器 V,其隐私性保证不向计算服务器和验证服务器泄漏数据信息;Gennaro 等人[40]首次提出非交互的可验证外包计算;Carter 等人[41]将基于 cut-and-choose 安全多方计算中的混乱电路求值外包给云服务器来做;文献[42-43]则将混乱电路的生产任务外包给移动设备;服务器辅助的通用两方计算协议方案有文献[44-45]等. 和本文工作最为相关的是 Wei 等人[45]提出的云服务器辅助茫然传输协议,该方案涉及 2 个非合谋的服务器且在半诚实模型下达到可证明安全.

1.2 本文工作

我们设计了一种新的基于单个云服务器的 k-out-of-n OT 协议. 文献[45]将群指数运算外包到 2 个诚实但好奇且相互独立的云服务器上,云服务器诚实但好奇的性质说明该协议是在半诚实模型下执行,云服务器之间相互独立意味着 2 个云服务器之间不能合谋. 该方案虽然降低了通信双方的本地计算量,却存在 2 个明显的弊端:首先外包计算目前仍需要支付大量的费用,将群指数运算外包到 2 个云服务器更是要有足够的经济支撑,出于预算考虑在某些应用场景下不得不放弃使用该协议;再者,2 个云服务器不能合谋的安全假设过于理想,2 个云服务器一旦合谋就可以知道接收方的输入选择,或者云服务器与接收方合谋来获得发送方的全部输入,这样协议的安全性就完全得不到保证. 因此,我们希望在保证安全和效率的基础上将云服务器的个数减少至一个,从而避免 2 个云服务器合谋的严重弊端. 我们具体的贡献有 3 方面:

1) 将 RAND 函数的计算过程分散到发送方和云服务器,并以一种新的形式嵌入发送方的输入值.

2) 将群指数运算外包到一个云服务器,设计出基于单个云辅助的高效 n 取 k 茫然传输协议,该协议是安全三方计算的实例,并依据诚实方大多数原则证明其安全性.

3) 相对于其他基于 DDH 假设的 n 取 k 茫然传输协议,我们将协议的计算和通信复杂度降至 $O(n)$ 量级.

2 基本知识和安全定义

2.1 DH 元组

令 G 为一个以 g 为生成元的 q 阶群,q 为素数. 将形如 $(g^{\alpha},g^{\beta},g^{\alpha\beta})$ 的三元组称为 DH 元组,这里的 α,β 随机取自 $\mathbb{Z}_q$. 也就是说,对于任意群 G 上的三元组 $(g^{\alpha},g^{\beta},g^{\gamma})$,$\alpha,\beta,\gamma\in\mathbb{Z}_q$,若 $\gamma\equiv\alpha\times\beta(\bmod q)$ 则称该元组为 DH 元组,否则为非 DH 元组.

2.2 DDH 假设

DDH 问题定义:给定上述群 G 上的四元组 $(g,g^{\alpha},g^{\beta},g^{\gamma})$,$\alpha,\beta,\gamma\in\mathbb{Z}_q$,判定 $\gamma\equiv\alpha\times\beta(\bmod q)$ 是否成立. 若等式成立,由于该元组的第 1 个元素为生成元,所以该四元组与上述所定义的三元组构成的 DH 元组并不矛盾,四元组 $(g,g^{\alpha},g^{\beta},g^{\gamma})$ 仍称为 DH 元组. 将生成元看作一个公开参数,那么任意三元组都可以写成四元组的形式,因此从现在起所有的 DH 元组均默认为四元组. 那么,DDH 假设是指下述 2 种总体分布是计算不可区分的:

1) $X_1=(g,g^{a},g^{b},g^{ab})$,这里 $a,b\in\mathbb{Z}_q$;

2) $X_2=(g,g^{a},g^{b},g^{c})$,这里 $a,b\in\mathbb{Z}_q$,但 $ab\neq c$.

2.3 RAND 函数

RAND 函数定义在群 G 构成的 DH 元组 (w,x,y,z) 上,$w,x,y,z\in G$,函数 $\mathrm{RAND}(w,x,y,z)=(u,v)$,其中 $u=w^{s}\times y^{t}$,$v=x^{s}\times z^{t}$,这里 $s,t\in\mathbb{Z}_q$. 注意,对任意的 DH 元组 (w,x,y,z),必然存在 $a\in\mathbb{Z}_q$,满足 $y=w^{a}$,$z=x^{a}$. 因此,RAND 函数有性质:

1) 如果(w,x,y,z)是 DH 元组且满足 $y=w^a$, $z=x^a$,那么对于 $u,v\leftarrow$RAND(w,x,y,z)满足 $u^a=v$.

2) 如果(w,x,y,z)为非 DH 元组,那么$(w,x,y,z,$RAND$(w,x,y,z))$,(w,x,y,z,g^a,g^b)在分布上是等价的,这里 α,β 随机取自$\mathbb{Z}_q$.

显然,RAND 函数的计算过程能够分为 w^s, y^t, x^s 和 z^t 这 4 个部分,我们可以将这 4 个部分分发给不同参与方,然后联合计算出函数值.

2.4 安全性定义

本文中我们考虑基于单服务器辅助的 n 取 k 茫然传输协议. 为了定义基于云服务器外包协议的安全性,我们参考了文献[45-46]中提出的安全多方计算在外包环境中安全定义. 在文献[45]中协议是基于 2 个相互独立的云服务器,我们的协议将云服务器个数减少为一个,相当于发送方在协议执行过程中扮演了原方案中一个云服务器的角色. 协议共涉及 3 个参与方,因此我们将协议的执行过程看作是安全三方计算的实例,遵循诚安全多方计算里诚实方大多数原则,即在该协议中敌手只能腐化至多一个参与方,因此不考虑任意 2 方合谋的情况. 我们设计的协议中,仍然定义云服务器是诚实但好奇的,且与协议双方保持独立. 诚实但好奇意味着云服务器不仅根据指令诚实地执行协议的每一步操作,并且想要依据接收到的消息副本来试图获得发送方或接收方的输入. 但云服务器不能与发送方或接收方中的任意一方合谋,也就是说云服务器只负责提供额外的计算能力或者依照协议指令传送特定内容.

现在我们分别定义发送方安全和接收方安全.

1) 针对云服务器的接收方安全定义. 这里利用不可区分性来定义针对云服务器的接收方安全. 对于接收方 2 个不同的输入集合,$C=\{\sigma_1,\sigma_2,\cdots,\sigma_k\}$ 和 $C'=\{\sigma'_1,\sigma'_2,\cdots,\sigma'_k\}$,接收方发送给云服务器的 2 个消息副本具有计算不可区分性.

2) 针对发送方的接收方安全定义. 针对发送方的接收方安全定义与针对云服务器的安全定义相同. 也就是说,针对接收方 2 个不同的选择 $C=\{\sigma_1,\sigma_2,\cdots,\sigma_k\}$, $C'=\{\sigma'_1,\sigma'_2,\cdots,\sigma'_k\}$,发送方不能在多项式时间内将对应的消息副本区分出来.

3) 针对接收方的发送方安全定义. 对于接收方的任意选择集合 $C=\{\sigma_1,\sigma_2,\cdots,\sigma_k\}$,发送方对接收方并未选择地输入值的加密结果应与随机计算不可区分.

3 单服务器辅助的 n 取 k 茫然传输协议

3.1 单服务器辅助的 OT_n^k 协议

在本节中,我们给出一个半诚实敌手模型下的单服务器辅助的 n 取 k 茫然传输协议的具体过程.

协议所设计的系统模型如图 1 所示. 协议共涉及 3 个参与方:发送方 S、接收方 R 和云服务器 Server. 协议一共需要进行 3 轮交互. 首先,R 根据自己的选择构造出多项式,并将多项式的系数分割成 2 部分分别发送给 S 和 Server;接着,S 接收到的部分系数后计算 RAND 函数的部分结果,并将自己的输入嵌入到其中;最后,Server 接收到来自 S 和 R 的消息后完成 RAND 函数的全部运算,将函数值发送给 R,R 在本地进行解密获得自己所选的值.

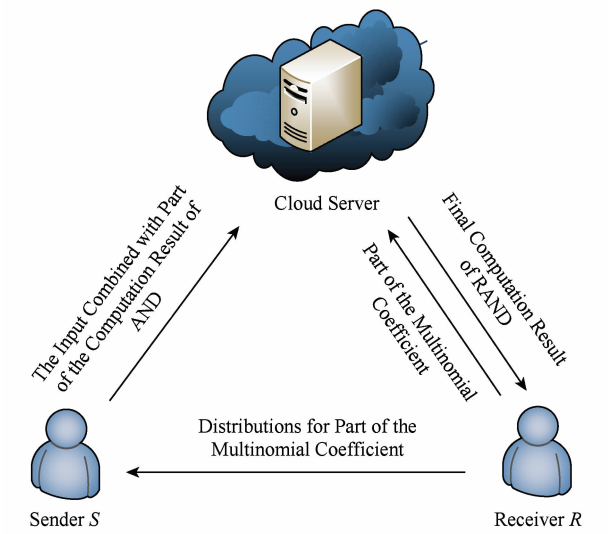


Fig. 1 The system model

图 1 系统模型

协议. 单服务器辅助的高效 n 取 k 茫然传输协议.

输入:S 输入 n 个值$(x_1,x_2,\cdots,x_n)\in\{0,1\}^n$、$R$ 输入 k 个值$(\sigma_1,\sigma_2,\cdots,\sigma_k)\in\{1,2,\cdots,n\}$、云服务器 Server 没有输入;

辅助输入:安全参数 n 以及一个 q 阶群 G,其生成元为 g_0;

输出:R 输出 $x_{\sigma_1},x_{\sigma_2},\cdots,x_{\sigma_k}$.

协议执行过程:

步骤 1. R 随机选择一个 k 阶多项式 $f(x)=(x-\sigma_1)(x-\sigma_2)\cdots(x-\sigma_k)+\beta=a_0+a_1x+\cdots+a_{k-1}x^{k-1}+x^k \bmod q$,其中 $\beta\in\mathbb{Z}_q$. 针对任意 $i=0,1,\cdots,k-1$,R 计算 $c_i=(g_0)^{a_i}$,并将(c_0,c_1)发送给 S,

将$(c_2,c_3,\cdots,c_{k-1})$发送给云服务器 Server. 然后,R 随机选择 $y_1,y_2,\cdots,y_n\in\mathbb{Z}_q$,并将这 n 个值分别发送给 S 和 Server. 此外,R 随机选择一个 $\sigma_m\in\{\sigma_1,\sigma_2,\cdots,\sigma_k\}$以及 $r\in\mathbb{Z}_q$,并计算 $g=(g_0)^{y_{\sigma_m}r}$和 $h=g^\beta$,然后将(g,h)发送给 Server.

步骤 2. 记 $g_1=(g_0)^{y_1},g_2=(g_0)^{y_2},\cdots,g_n=(g_0)^{y_n}$. 发送方 S 首先计算$(c_j)^{y_i}=(g_0)^{a_jy_i}=(g_i)^{a_j}$,这里 $i=1,2,\cdots,n,j=0,1$. 令 $f_1(x)=a_0+a_1x$,然后发送方就可以计算出 $h_i^{(Sen)}=(g_i)^{f_1(i)}=(g_i)^{a_0+a_1i},i=1,2,\cdots,n$. 随后,$S$ 计算出 $v_i^{(Sen)}=(h_i^{(Sen)})^{t_i}\times x_i$,这里 $t_i\in\mathbb{Z}_q,x_i$ 是 S 的第 i 个输入. 最后,S 将$(v_1^{(Sen)},t_1),(v_2^{(Sen)},t_2),\cdots,(v_n^{(Sen)},t_n)$发送给云服务器 Server.

步骤 3. 同样记 $g_1=(g_0)^{y_1},g_2=(g_0)^{y_2},\cdots,g_n=(g_0)^{y_n}$,云服务器 Server 在接收到$(c_2,c_3,\cdots,c_{k-1})$后首先检查元素个数是否为 $k-2$,若不是则停止协议,若是则首先计算$(c_j)^{y_i}=(g_0)^{a_jy_i}=(g_i)^{a_j}$,这里 $i=1,2,\cdots,n,j=2,3,\cdots,k-1$. 令 $f_2(x)=a_2x^2+a_3x^3+\cdots+a_{k-1}x^{k-1}+x^k$,然后对 $i=1,2,\cdots,n$,可以计算得到:$h_i^{(Ser)}=(g_i)^{f_2(i)}=(g_i)^{a_2i^2+a_3i^3+\cdots+a_{k-1}i^{k-1}+i^k}$. 随后 Server 根据接收自 S 的信息,计算 $v_i=v_i^{(Sen)}\times(h_i^{(Ser)})^{t_i}=((g_i)^{f_1(i)+f_2(i)})^{t_i}\times x_i=(g_i)^{f(i)\times t_i}\times x_i$. 最后,对任意 $i=1,2,\cdots,n$,Server 随机选择 $s_i\in\mathbb{Z}_q$,计算 $u_i=(g)^{s_i}\times(h)^{t_i},w_i=(g_i)^{s_i}\times v_i$,并将$(u_1,w_1),(u_2,w_2),\cdots,(u_n,w_n)$发送给 R.

步骤 4. R 输出:

1) 针对步骤 1 中 R 随机选择的σ_m,R 计算:

$$x_{\sigma_m}=\frac{w_{\sigma_m}}{(u_{\sigma_m})^{r^{-1}}};$$

2) 针对每一个 $\sigma_j\in\{\sigma_1,\sigma_2,\cdots,\sigma_k\}$且 $\sigma_j\notin\sigma_m$,R 计算:

$$x_{\sigma_j}=\frac{w_{\sigma_j}}{(u_{\sigma_j})^{r^{-1}\times y_{\sigma_m}^{-1}\times y_{\sigma_j}}}.$$

3.2 协议正确性分析

在证明协议安全性之前,我们首先分析协议的正确性. 假设所有参与方都诚实地执行该协议,在协议执行完成后接收方 R 能够获得与自己的选择相对应的 k 个值. 首先从 R 构造的 k 阶多项式中我们可以看出:多项式 $f(x)-\beta=(x-\sigma_1)(x-\sigma_2)\cdots(x-\sigma_k)$有$(\sigma_1,\sigma_2,\cdots,\sigma_k)$共 k 个根,这也就意味着 $f(\sigma_1)=f(\sigma_2)=\cdots=f(\sigma_k)=\beta$. 因此,对于每一个 $i\in\{\sigma_1,\sigma_2,\cdots,\sigma_k\}$ 都有 $(g_i)^{f(i)}=(g_i)^\beta$,元组 $(g_i,(g_i)^r,(g_i)^{f(i)},(g_i)^{r\times\beta})$为 DH 元组. 注意到,$g=(g_0)^{y_{\sigma_m}r}=(g_{\sigma_m})^r,h=g^\beta=(g_{\sigma_m})^{r\times\beta}$,且对于 $\sigma_j\in\{\sigma_1,\sigma_2,\cdots,\sigma_k\}$,有$(g_{\sigma_m})^{y_{\sigma_m}^{-1}\times y_{\sigma_j}}=g_{\sigma_j}$,因此 R 可得:

1) 针对 $i=\sigma_m$,有 $f(i)=\beta$,

$$\frac{w_i}{(u_i)^{r^{-1}}}=\frac{(g_{\sigma_m})^{s_i}\times(g_{\sigma_m})^{f(i)\times t_i}\times x_i}{((g_{\sigma_m})^{r\times s_i}\times(g_{\sigma_m})^{r\times\beta\times t_i})^{r^{-1}}}=x_i;$$

2) 针对每一个 $i\in\{\sigma_1,\sigma_2,\cdots,\sigma_k\}$但 $i\neq\sigma_m$,同样有 $f(i)=\beta$,计算得到:

$$\frac{w_i}{(u_{\sigma_j})^{r^{-1}\times y_{\sigma_m}^{-1}\times y_{\sigma_j}}}=\frac{(g_i)^{s_i}\times(g_i)^{\beta\times t_i}\times x_i}{((g_{\sigma_m})^{r\times s_i}\times(g_{\sigma_m})^{r\times\beta\times t_i})^{r^{-1}\times y_{\sigma_m}^{-1}\times y_i}}=\frac{(g_i)^{s_i}\times(g_i)^{\beta\times t_i}\times x_i}{((g_i)^{r\times s_i}\times(g_i)^{r\times\beta\times t_i})^{r^{-1}}}=x_i;$$

3) 当 $i\in\{1,2,\cdots,n\}$但 $i\notin\{\sigma_1,\sigma_2,\cdots,\sigma_k\}$,此时 $f(i)\neq\beta$,则计算为

$$\frac{w_i}{(u_{\sigma_j})^{r^{-1}\times y_{\sigma_m}^{-1}\times y_{\sigma_j}}}=\frac{(g_i)^{s_i}\times(g_i)^{f(i)\times t_i}\times x_i}{((g_{\sigma_m})^{r\times s_i}\times(g_{\sigma_m})^{r\times\beta\times t_i})^{r^{-1}\times y_{\sigma_m}^{-1}\times y_i}}=\frac{(g_i)^{s_i}\times(g_i)^{f(i)\times t_i}\times x_i}{(g_i)^{s_i}\times(g_i)^{\beta\times t_i}}\neq x_i.$$

综上可以看出,接收方 R 能且仅能获得与自己选择相对应的 k 个值.

3.3 协议安全性证明

下面我们给出协议的形式化安全证明.

定理 1. 如果云服务器 Server 是诚实但好奇的且与发送方 S 相互独立,那么针对发送方协议满足接收方的安全性要求.

证明. 在协议中,接收方 R 在构造出 k 阶多项式 $f(x)=(x-\sigma_1)(x-\sigma_2)\cdots(x-\sigma_k)+\beta=a_0+a_1x+\cdots+a_{k-1}x^{k-1}+x^k \bmod q$ 后,以加密的形式将多项式的部分系数分别发送给 S 和 Server. 我们将要说明的是,发送方和云服务器在接收到这些消息后均不能推断出原多项式 $f(x)$的根$(\sigma_1,\sigma_2,\cdots,\sigma_k)$. 在协议中,接收方把 k 阶多项式分成 2 部分,$f_1(x)$和 $f_2(x)$,然后把 $f_1(x)=a_0+a_1x$ 的系数加密为$(g_0^{a_0},g_0^{a_1})$发送给 S,把 $f_2(x)=a_2x^2+a_3x^3+\cdots+a_{k-1}x^{k-1}+x^k$ 的系数加密为$(g_0^{a_2},g_0^{a_3},\cdots,g_0^{a_{k-1}})$发送给云服务器 Server. 由于多项式 $f(x)$满足 $f(\sigma_1)=f(\sigma_2)=\cdots=f(\sigma_k)=\beta$,因此在$(g_0^{f(1)},g_0^{f(2)},\cdots,g_0^{f(n)})$中有 k 个值等于 g_0^β. 然而在$(g_0^{f_1(1)},g_0^{f_1(2)},\cdots,g_0^{f_1(n)})$和$(g_0^{f_2(1)},g_0^{f_2(2)},\cdots,g_0^{f_2(n)})$中可能不具备这个性质. 假设在 Server 一方中,存在 $i,j\in\{1,2,\cdots,n\}$,满足

$g_0^{f_2(i)}=g_0^{f_2(j)}$,由于 Server 不知道 $f_1(x)$所以无法确定$g_0^{f_1(i)}\times g_0^{f_2(i)}=g_0^{f_1(j)}\times g_0^{f_2(j)}=g_0^{\beta}$是否成立.因此,对于任意 $i,j\in\{1,2,\cdots,n\}$,Server 只能随机猜测 $i,j\in\{\sigma_1,\sigma_2,\cdots,\sigma_k\}$成立与否.对于发送方 S 来说也只能随机猜测哪些是 R 的输入,证明过程与云服务一方的证明方法类似,这里不再赘述.因此,面对发送方 S 和云服务器 Server,协议保护了 R 的输入隐私. 证毕.

定理 2. 如果在群 G 上 DDH 假设成立,并且发送方 S 与云服务器 Server 是诚实但好奇的且相互独立,那么针对云服务器协议满足接收方的安全性要求.

证明. 在协议执行的第 3 个阶段,云服务器 Server 获得 n 个四元组$(g_i,g,(g_i)^{f(i)},h)$,其中有 k 个为 DH 元组,$n-k$ 个为非 DH 元组.Server 计算得到$(u_i,v_i)=\mathrm{RAND}(g,g_i,h,(g_i)^{f(i)})$,即 $u_i=(g)^{s_i}\times(h)^{t_i}$, $v_i=(g_i)^{s_i}\times(g_i)^{f(i)\times t_i}$,$i=1,2,\cdots,n$.假设云服务器拥有不可忽略的优势知道接收方 R 的选择,由于这 k 个 DH 元组与接收方 R 的选择相关,也就意味着云服务器 Server 能够区分 n 个元组中哪些是 DH 元组而哪些不是.因此我们构造一个运行在多项式时间的区分器 D,它调用 Server 作为自己的子程序来解决 DDH 困难问题.

首先定义 2 组选择集合 C 和 C',这里 $C=\{\sigma_1,\sigma_2,\cdots,\sigma_k\}$,$C'=\{\sigma'_1,\sigma'_2,\cdots,\sigma'_k\}$.将协议中云服务器 Server 的视图定义为发送方 S 和接收方 R 发送给云服务器 Server 消息的副本,用符号 $View^{C}_{\mathrm{Server}}$ 表示,这里 C 为接收方 R 所选的集合.根据协议,$View^{C}_{\mathrm{Server}}=((g,h),(g_1,h_1),(g_2,h_2),\cdots,(g_n,h_n))$,这里 $h_i=g_i^{f(i)}$,$i=1,2,\cdots,n$.假设接收方所选的集合为 C,那么在视图 $View^{C}_{\mathrm{Server}}$ 中一共存在 k 个元组为 DH 元组:$(g_{\sigma_1},g,h_{\sigma_1},h),(g_{\sigma_2},g,h_{\sigma_2},h),\cdots,(g_{\sigma_k},g,h_{\sigma_k},h)$,同理如果所选集合为 C',那么 DH 元组为$(g_{\sigma'_1},g,h_{\sigma'_1},h),(g_{\sigma'_2},g,h_{\sigma'_2},h),\cdots,(g_{\sigma'_k},g,h_{\sigma'_k},h)$.对于 $i=1,2,\cdots,k$,不妨令某一 $\sigma_m\in C$ 且 $\sigma_m\notin C'$,此时元组 $Tuple=(g_{\sigma_m},g,h_{\sigma_m},h)$为 DH 元组当且仅当所选集合为 C.由于元组个数不影响 DDH 假设的计算不可区分性,那么区分视图 $View^{C}_{\mathrm{Server}}$ 和 $View^{C'}_{\mathrm{Server}}$的分布就规约到区分 $Tuple$ 是否为 DH 元组的问题上.如果云服务器能够以一个不可忽略的概率 ε 区分视图 $View^{C}_{\mathrm{Server}}$ 和 $View^{C'}_{\mathrm{Server}}$,那么我们可以构造一个运行在多项式时间的区分器 D,它调用 Server 做为自己的子程序来解决 DDH 问题,那么 D 能区分 $Tuple$ 是否为 DH 元组的概率至少是 $\varepsilon-\frac{2}{q}$,显然与 DDH 假设矛盾,因此 $View^{C}_{\mathrm{Server}}$ 和 $View^{C'}_{\mathrm{Server}}$在计算上不可区分. 证毕.

定理 3. 如果在群 G 上 DDH 假设成立,并且发送方 S 与云服务器 Server 是诚实但好奇的且相互独立,那么针对接收方协议满足发送方的安全性要求.

证明. 首先我们分析接收方 R 通过恶意行为来获取除自己选择外的值.接收方 R 可以试图构造一个多项式 $f(x)$满足 $f(1)=f(2)=\cdots=f(n)=\beta$,从而可以恢复出发送方 S 的 n 个输入.这必然导致所构造出的多项式的阶大于 k,在协议中云服务器 Server 检查了接收方 R 所发送的系数的个数,利用这个方法可以将多项式 $f(x)$的阶严格限制为 k,因此 n 个元组中 DH 元组的个数也是 k.接下来证明接收方 R 不可能从 $n-k$ 个非 DH 元组中获取额外信息.

根据 RAND 函数的性质,如果(w,x,y,z)为非 DH 元组,那么 $\mathrm{RAND}(w,x,y,z)$的结果是群 G 中的随机值.假设接收方的输入为$(\sigma_1,\sigma_2,\cdots,\sigma_k)$,对于每一个 $j\notin\{\sigma_1,\sigma_2,\cdots,\sigma_k\}$对应的$(g_j,g,h_j,h)$都不是 DH 元组.不妨从$\mathbb{Z}_q$中随机选择 a 和 b,使得 $h_j=(g_j)^a$,$h=g^b$,且 $a\neq b$.为了证明 $\mathrm{RAND}(g_j,g,h_j,h)$的结果在群 G 中是随机的,只需说明 $Pr[\mathrm{RAND}(g_j,g,h_j,h)=(g_0^{\alpha},g_0^{\beta})]=\frac{1}{q^2}$,这里 α,β 随机取自$\mathbb{Z}_q$.由于 $\mathrm{RAND}(g_j,g,h_j,h)=((g_j)^s\times(h_j)^t,(g)^s\times(h)^t)$,这里 s,t 随机取自$\mathbb{Z}_q$,因此上述的概率是来自于 s 和 t 的随机性.令 $g=(g_j)^{\gamma}$,γ 随机取自$\mathbb{Z}_q$,那么元组(g_j,g,h_j,h)可以表示为$(g_j,(g_j)^{\gamma},(g_j)^{a},(g_j)^{\gamma\times b})$,相应的 $\mathrm{RAND}(g_j,g,h_j,h)=((g_j)^{s+a\times t},(g_j)^{s\times\gamma+\gamma\times b\times t})$.这样,对于方程组:

$$\begin{cases}s+a\times t=\alpha,\\ s\times\gamma+\gamma\times b\times t=\beta,\end{cases}$$

由于 $a\neq b$,系数矩阵组成的行列式$\begin{vmatrix}1 & a\\ \gamma & \gamma\times b\end{vmatrix}\neq 0$,那么就一定存在一对 s,t 使上述方程组成立.由于 s 和 t 均以$\frac{1}{q}$的概率从$\mathbb{Z}_q$中独立随机选取,所以方程组有解的概率为$\frac{1}{q^2}$,即非 DH 元组 $\mathrm{RAND}(g_j,g,h_j,h)$的结果在群 G 中是随机的,也就是非 DH 元

组计算 RAND(g_j,g,h_j,h)的结果与群 G 中的随机值计算不可区分.

综上所述,我们完成对协议的安全性证明.

证毕.

3.4 协议效率分析与比较

我们设计的协议中需要交互 3 轮:第 1 轮接收方 R 向同时向发送方 S 和云服务器 Server 发送消息;第 2 轮发送方 S 向云服务器 Server 发送一系列本地计算的结果;第 3 轮云服务器 Server 接收到发送方 S 发送的消息后将四元组"加密"后发送给接收方 R,R 在本地进行解密运算.

每一轮中具体的指数运算如下:

1) 接收方 R 共进行了 $k+1$ 次群指数运算. 其中 $g_0^{a_1},g_0^{a_2},\cdots,g_0^{a_{k-1}}$ 需要 $k-1$ 次,2 次群指数运算分别用来计算 $g=(g_0)^{y_{\sigma_m}r}$,$h=g^{\beta}$.

2) 发送方 S 共进行了 $4n-1$ 次群指数运算. $2n$ 次群指数运算用来计算矩阵:

$$\begin{pmatrix}(c_0)^{y_1} & (c_0)^{y_2} & \cdots & (c_0)^{y_n}\\ (c_1)^{y_1} & (c_1)^{y_2} & \cdots & (c_1)^{y_n}\end{pmatrix},$$

$n-1$ 次群指数运算用来计算矩阵:

$$((c_1)^{y_2\times 2} \quad (c_1)^{y_3\times 3} \quad \cdots \quad (c_1)^{y_n\times n}),$$

n 次群指数运算用来计算$(h_i^{(Sen)})^{t_i}$. 对于形如 $g^a\times g^b$ 的 2 个群指数相乘的计算复杂度可以优化为 1.25 次群指数运算,所以可以将上述运算合并为该形式,即$(c_0)^{y_i}\times(c_1)^{y_i\times i}$,$i=1,2,\cdots,n$,因此发送方的群指数运算次数可以减少为 $3.25n$.

3) 云服务器 Server 共进行 $2(k+1)n+1-k$ 次群指数运算. $(k-1)n$ 次用于计算矩阵:

$$\begin{pmatrix}(c_2)^{y_1} & (c_2)^{y_2} & \cdots & (c_2)^{y_n}\\ (c_3)^{y_1} & (c_3)^{y_2} & \cdots & (c_3)^{y_n}\\ \vdots & \vdots & & \vdots\\ (c_{k-1})^{y_1} & (c_{k-1})^{y_2} & \cdots & (c_{k-1})^{y_n}\\ (g_0)^{y_1} & (g_0)^{y_2} & \cdots & (g_0)^{y_n}\end{pmatrix},$$

$(n-1)(k-1)$次用于计算矩阵:

$$\begin{pmatrix}(c_2)^{y_2\times 2^2} & (c_2)^{y_3\times 3^2} & \cdots & (c_2)^{y_n\times n^2}\\ (c_3)^{y_2\times 2^3} & (c_3)^{y_3\times 3^3} & \cdots & (c_3)^{y_n\times n^3}\\ \vdots & \vdots & & \vdots\\ (c_{k-1})^{y_2\times 2^{k-1}} & (c_{k-1})^{y_3\times 3^{k-1}} & \cdots & (c_{k-1})^{y_n\times n^{k-1}}\\ (g_0)^{2^k} & (g_0)^{3^k} & \cdots & (g_0)^{n^k}\end{pmatrix}.$$

此外$(h_i^{(Sen)})^{t_i}$,$(g)^{s_i}\times(h)^{t_i}$ 和$(g_i)^{s_i}$ 共需 $4n$ 次群指数运算.

最后,我们统计出协议中的各参与方发送的群元素的个数:

1) 接收方 R 发送 2 个群元素$((g_0)^{a_0},(g_0)^{a_1})$给发送方 S,给云服务器 Server 共 k 个群元素:

$((g_0)^{a_2},(g_0)^{a_3},\cdots,(g_0)^{a_{k-1}},g,h)$.

2) 发送方 S 发送给云服务器 Server 共 $2n$ 个元素:$(v_1^{(Sen)},t_1),(v_2^{(Sen)},t_2),\cdots,(v_n^{(Sen)},t_n)$.

3) 云服务器 Sever 把 $2n$ 个元素发送给接收方 R:$(u_1,w_1),(u_2,w_2),\cdots,(u_n,w_n)$.

表 1 是我们所设计的协议与文献[12,45]的对比结果. Chu 等人[19]提出的 n 取 k 茫然传输协议不需要云服务器辅助,因此需要发送方和接收方进行大量群指数运算. Wei 等人提出的协议将大部分群指数运算外包至 2 个云服务器,从而降低发送方和接收方的群指数运算. 我们的协议仅使用一个云服务器且相对高效.

Table 1 Comparison of *k*-out-of-*n* Oblivious Transfer Protocols

表 1 已有的 *k*-out-of-*n* 茫然传输协议比较

Protocol	Round Complexity	Exponentiation Operations	Number of Cloud Sever
Ref[19]	2	$(k+2)n+3k+2$	0
Ref[45]	3	$2.5n+2k+3$	2
Ours	3	$3.25n+k+1$	1

4 结束语

本文主要针对 OT_n^k 问题,基于 DDH 困难问题假设,构造出单服务器辅助下的高效 n 取 k 茫然传输协议. 协议共需要 3 轮并将计算和通信复杂度降至 $O(n)$. 将来的工作包括考虑恶意模型的构造,即允许恶意敌手之间相互合谋的情况下如何设计高效的云辅助 OT_n^k 协议.

参 考 文 献

[1] Li Zhenhua, Dai Yafei, Chen Guihai, et al. Content Distribution for Mobile Internet: A Cloud-based Approach [M]. Singapore: Springer Science Business Media, 2016

[2] Liu Zhe, Weng Jian, Hu Zhi, et al. Efficient elliptic curve cryptography for embedded devices [J]. ACM Trans on Embedded Computing Systems, 2016, 16(2): Article No 53

[3] Liu Zhe, Seo H, Großschädl J, et al. Efficient implementation of NIST-compliant elliptic curve cryptography for 8-bit AVR-based sensor nodes [J]. IEEE Trans on Information Forensics and Security, 2016, 11(7): 1385-1397

[4] Jiang Han, Xu Qiuliang. Advances in key techniques of practical secure multi-party computation [J]. Journal of Computer Research and Development, 2015, 52(10): 2247-2257 (in Chinese)
(蒋瀚, 徐秋亮. 实用安全多方计算协议关键技术研究进展[J]. 计算机研究与发展, 2015, 52(10): 2247-2257)

[5] Rabin M O. How to exchange secrets with oblivious transfer, TR-81 [R]. Cambridge, MA: Harvard University, 1981

[6] Even S, Goldreich O, Lempel A. A randomized protocol for signing contracts [J]. Communications of the ACM, 1985, 28(6): 637-647

[7] Brassard G, Crepeau C, Robert J M. All-or-nothing disclosure of secrets [G] //LNCS 263: Advances in Cryptology (CRYPTO 1986). Berlin: Springer, 1986: 234-238

[8] Stern J P. A new and efficient all-or-nothing disclosure of secrets protocol [G] //LNCS 1514: Advances in Cryptology (Asiacrypt 1998). Berlin: Springer, 1998: 357-371

[9] Cramer R, Shoup V. Universal Hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption [G] //LNCS 2332: Advances in Cryptology (EUROCRYPT 2002). Berlin: Springer, 2002: 45-64

[10] Kalai Y T. Smooth projective hashing and two-message oblivious transfer [G] //LNCS 3494: Advances in Cryptology (EUROCRYPT 2005). Berlin: Springer, 2005: 78-95

[11] Naor M, Pinkas B. Efficient oblivious transfer protocols [C]//Proc of the 12th Annual ACM-SIAM Symp on Discrete Algorithms (SODA). New York: ACM, 2001: 448-457.

[12] Tzeng W G. Efficient 1-out-*n* oblivious transfer schemes [G] //LNCS 2274: Public Key Cryptography. Berlin: Springer, 2002: 159-171

[13] Wei Xiaochao, Jiang Han, Zhao Chuan. An efficient 1-out-of-*n* oblivious transfer protocol with full simulation [J]. Journal of Computer Research and Development, 2016, 53(11): 2475-2481 (in Chinese)
(魏晓超, 蒋瀚, 赵川. 一个高效可完全模拟的 *n* 取 1 茫然传输协议[J]. 计算机研究与发展, 2016, 53(11): 2475-2481)

[14] Lindell A Y. Efficient fully-simulatable oblivious transfer [G] //LNCS 4964: Topics in Cryptology (CT-RSA 2008). Berlin: Springer, 2008: 52-70

[15] Peikert C, Vaikuntanathan V, Waters B. A framework for efficient and composable oblivious transfer [G] //LNCS 5157: Advances in Cryptology (CRYPTO 2008). Berlin: Springer, 2008: 554-571

[16] Feng Tao, Ma Jianfeng, Li Fenghua. Efficient and universally composable security oblivious transfer [J]. Acta Electronica Sinica, 2008, 36(1): 17-23 (in Chinese)
(冯涛, 马建峰, 李凤华. UC 安全的高效不经意传输协议[J]. 电子学报, 2008, 36(1): 17-23)

[17] Green M, Hohenberger S. Universally composable adaptive oblivious transfer [G] //LNCS 5350: Advances in Cryptolog (Asiacrypt 2008). Berlin: Springer, 2008: 179-197

[18] Garay J A, Ishai Y, Kumaresan R, et al. On the complexity of UC commitments [G] //LNCS 8441: Advances in Cryptology (CRYPTO 2014). Berlin: Springer, 2014: 677-694

[19] Chu C K, Tzeng W G. Efficient *k*-out-of-*n* oblivious transfer schemes with adaptive and non-adaptive queries [G] //LNCS 3386: Public Key Cryptography (PKC 2005). Berlin: Springer, 2005: 172-183

[20] Zhang Jianhong, Wang Yumin. Two provably secure *k*-out-of-*n* oblivious transfer schemes [J]. Applied Mathematics and Computation, 2005, 169(2): 1211-1220

[21] Camenisch J, Neven G, Shelat A. Simulatable adaptive oblivious transfer [G] //LNCS 4515: Advances in Cryptology (EUROCRYPT 2007). Berlin: Springer, 2007: 573-590

[22] Zeng Bin, Tartary C, Xu Peng, et al. A practical framework for *t*-out-of-*n* oblivious transfer with security against covert adversaries [J]. IEEE Trans on Information Forensics and Security, 2012, 7(2): 465-479

[23] Harnik D, Ishai Y, Kushilevitz E, et al. OT-combiners via secure computation [G] //LNCS 4948: Theory of Cryptography. Berlin: Springer, 2008: 393-411

[24] Nielsen J B, Nordholt P S, Orlandi C, et al. A new approach to practical active-secure two-party computation [G]//LNCS 7417: Advances in Cryptology (CRYPTO 2012). Berlin: Springer, 2012: 681-700

[25] Asharov G, Lindell Y, Schneider T, et al. More efficient oblivious transfer extensions [J]. Journal of Cryptology, 2017, 30(3): 805-858

[26] Patra A, Sarkar P, Suresh A. Fast actively secure OT extension for short secrets [C] //Proc of the 24th Network and Distributed System Security Symp (NDSS). Reston, VA: Internet Society, 2017

[27] Zhao Chuan, Jiang Han, Wei Xiaochao, et al. Cut-and-choose bilateral oblivious transfer and its application [C] //Proc of the 14th IEEE Int Conf on Trust, Security and Privacy in Computing and Communications. Los Alamitos, CA: IEEE Computer Society, 2015: 384-391

[28] Wei Xiaochao, Jiang Han, Zhao Chuan, et al. Fast cut-and-choose bilateral oblivious transfer for malicious adversaries [C] //Proc of the 15th IEEE Int Conf on Trust, Security and Privacy in Computing and Communications. Los Alamitos, CA: IEEE Computer Society, 2016: 418-425

[29] Plesch M, Pawłowski M, Pivoluska M. 1-out-of-2 oblivious transfer using a flawed bit-string quantum protocol [J]. Physical Review A, 2017, 95(4): 042324

[30] Yang Yugang, Yang Rui, Cao Weifeng, et al. Flexible quantum oblivious transfer [J]. International Journal of Theoretical Physics, 2017, 56(4): 1286-1297

[31] Van Dijk M, Clarke D, Gassend B, et al. Speeding up exponentiation using an untrusted computational resource [J]. Designs, Codes and Cryptography, 2006, 39(2): 253-273

[32] Ma Xu, Li Jin, Zhang Fangguo. Outsourcing computation of modular exponentiations in cloud computing [J]. Cluster Computing, 2013, 16(4): 787-796

[33] Hohenberger S, Lysyanskaya A. How to securely outsource cryptographic computations [G] //LNCS 3378: Theory of Cryptography Conference (TCC 2005). Berlin: Springer, 2005: 264-282

[34] Chen Xiaofeng, Li Jin, Ma Jianfeng, et al. New algorithms for secure outsourcing of modular exponentiations [J]. IEEE Trans on Parallel and Distributed Systems, 2014, 25(9): 2386-2396

[35] Wang Yujie, Wu Qianhong, Wong D S, et al. Securely outsourcing exponentiations with single untrusted program for cloud storage [G] //LNCS 8712: Computer Security (ESORICS 2014). Berlin: Springer, 2014: 326-343

[36] Chevalier C, Laguillaumie F, Vergnaud D. Privately outsourcing exponentiation to a single server: Cryptanalysis and optimal constructions [C] //LNCS 9878: Computer Security (ESORICS 2016). Berlin: Springer, 2016: 261-278

[37] Tsang P P, Chow S S M, Smith S W. Batch pairing delegation [G] //LNCS 4752: Advances in Information and Computer Security. Berlin: Springer, 2007: 74-90

[38] Canard S, Devigne J, Sanders O. Delegating a pairing can be both secure and efficient [G] //LNCS 8479: Applied Cryptography and Network Security. Berlin: Springer, 2014: 549-565

[39] Xu G, Amariucai G T, Guan Y. Delegation of computation with verification outsourcing: Curious verifiers [J]. IEEE Trans on Parallel and Distributed Systems, 2017, 28(3): 717-730

[40] Gennaro R, Gentry C, Parno B. Non-interactive verifiable computing: Outsourcing computation to untrusted workers [G] //LNCS 6223: Advances in Cryptology (CRYPTO 2010). Berlin: Springer, 2010: 465-482

[41] Carter H, Mood B, Traynor P, et al. Secure outsourced garbled circuit evaluation for mobile devices [J]. Journal of Computer Security, 2016, 24(2): 137-180

[42] Carter H, Lever C, Traynor P. Whitewash: Outsourcing garbled circuit generation for mobile devices [C] //Proc of the 30th Annual Computer Security Applications Conf. New York: ACM, 2014: 266-275

[43] Mohassel P, Orobets O, Riva B. Efficient server-aided 2PC for mobile phones [J]. Proceedings on Privacy Enhancing Technologies, 2016, 2016(2): 82-99

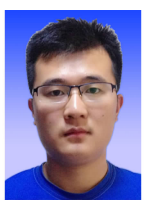
[44] Carter H, Mood B, Traynor P, et al. Outsourcing secure two-party computation as a black box [J]. Security and Communication Networks, 2016, 9(14): 2261-2275

[45] Wei Xiaochao, Zhao Chuan, Jiang Han, et al. Practical server-aided k-out-of-n oblivious transfer protocol [G] //LNCS 9663: Green, Pervasive, and Cloud Computing 2016. Berlin: Springer, 2016: 261-277

[46] Kamara S, Mohassel P, Raykova M. Outsourcing multi-party computation [OL]. [2017-08-25]. https://eprint.iacr.org/2011/272/pdf

Zhao Shengnan, born in 1994. PhD candidate. His main research interests include secure multiparty computation and search on encrypted data.

Jiang Han, born in 1974. PhD and lecturer. His main research interests include theory of cryptography, side-channel attacks and cryptographic protocols.



Wei Xiaochao, born in 1990. PhD and lecturer. His main research interests include secure multiparty computation and search on encrypted data (weixiaochao2008@163.com).

Ke Junming, born in 1994. Master candidate. His main research interests include computer security and cryptocurrencies (Junmingke1994@gmail.com).

Zhao Minghao, born in 1991. PhD candidate in Tsinghua University. Received his MSc degree in Shandong University and bachelor degree in Harbin Engineering University. Student member of ACM, CCF and CACR. His main research interests include cloud computing, storage system, mobile computing, privacy-preserving techniques and applied cryptography.