Software Engineering and Applications 软件工程与应用, 2023, 12(6), 895-907
Published Online December 2023 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2023.126087

Hans汉斯

基于等值测试的多方隐私集合求交方案

高　钦，李子臣

北京印刷学院信息工程学院，北京

收稿日期：2023年10月30日；录用日期：2023年12月12日；发布日期：2023年12月21日

摘　要

隐私集合求交(Private Set Intersection, PSI)技术可以在保护参与者私有数据集合隐私性的前提下计算出所有参与者的集合交集。作为隐私计算的关键技术，已经在云计算和数据挖掘等领域有了广泛的应用。密文等值测试(Equality Test)技术可以判断不同公钥加密下数据的异同。文中设计了基于密文等值测试技术的多方隐私集合求交方案，实现了不同公钥加密下私有数据集合的隐私求交，并利用等值测试的授权陷门技术将方案拓展为多方。此外，考虑到PSI的计算和存储代价，文中引入了云服务器来分担用户的计算和存储开销，并在半诚实模型下证明了方案的安全性。与现有方案相比，所提方案通信和计算代价较低，使用范围更广。

关键词

隐私集合求交，数据集合，隐私性，集合交集

Multi-Party Privacy Set Intersection Scheme Based on Equivalence Testing

Qin Gao, Zichen Li

School of Information Engineering, Beijing Institute of Graphic Arts, Beijing

Received: Oct. 30th, 2023; accepted: Dec. 12th, 2023; published: Dec. 21st, 2023

Abstract

Private Set Intersection (PSI) technology can calculate the set intersection of all participants while protecting the privacy of participants' private data sets. As a key technology for privacy computing, it has been widely used in fields such as cloud computing and data mining. Ciphertext equality test technology can determine the similarities and differences of data encrypted by different public keys. This paper designs a multi-party privacy set intersection scheme based on ciphertext equivalence testing technology, realizes the privacy intersection of private data sets under differ-

文章引用: 高钦, 李子臣. 基于等值测试的多方隐私集合求交方案[J]. 软件工程与应用, 2023, 12(6): 895-907.
DOI: 10.12677/sea.2023.126087

ent public key encryption, and uses the authorization trapdoor technology of equivalence testing to extend the scheme to multiple parties. In addition, considering the computing and storage costs of, the paper introduces a cloud server to share the user's computing and storage overhead, and proves the security of the scheme under the semi-honest model. Compared with existing solutions, the proposed solution has lower communication and calculation costs and a wider range of applications.

Keywords

Privacy Set Intersection, Data Set, Privacy, Set Intersection

1. 引言

随着信息技术的飞速发展、网络数据的爆炸性增长、数据泄露问题的不断凸显。社会大众更加注重个人隐私信息的保护，但是因噎废食的行为往往不可取。许多有益的数据处理往往需要个人隐私数据的支持。设想一下，针对某种特异性的传染疾病，需要跟据用户的基因序列、个人病例以及家族遗传史来进行易感人群排查。这就需要将每个用户的具体信息与特异性疾病的致病特征进行相等性判断，得到二者数据的交集，以推断该用户的易感程度。此类方案往往不可避免地涉及大量的数据存储和计算开销。云服务商可以为用户提供云存储和云计算的技术支持，将数据外包到云服务商显然成为一种切实可行的解决方案。然而，为保护个人隐私，用户在上传数据的时候往往需要进行加密操作。云服务商如何判断不同密钥加密下的明文数据是否相同成为一个挑战。如何构造一个适用于多用户的 PSI 方案以减轻通信和计算代价同样也是一个挑战。

隐私集合求交(PSI)允许各个持有隐私集合的参与方在不泄露任何集合信息的前提下计算出参与方的集合交集。Meadows 首次利用公钥加密体制实现 PSI [1]，其核心是利用 Diffie-Hellman 密钥交换协议将两个明文集合转化为会话密钥实现 PSI 的功能。Huang 等利用 Yao 电路构造了半诚实敌手模型下的 PSI 协议[2]。为了实现更加安全高效的 PSI 协议、基于哈希函数[3]、布隆过滤器[4]、不经意传输[5] [6]等技术的 PSI 协议被相继提出。但是，大部分的协议是基于双方的，双方的协议往往无法满足现实中的应用，就开篇提到的应用场景而言就需要多个用户进行参与。Freedman 等人首次提出了基于同态加密技术的多方 PSI 方案[7]。此外，基于椭圆曲线[8]、交互式零知识证明[9]、不经意线性函数估值技术的多方 PSI 协议被相继提出。然而这些协议往往都是用户服务器模式下，多方交互式进行的。由于庞大的信息量和计算量，该模式不可避免地限制了中小型企业开展 PSI 服务，如果能够将数据外包到云服务商，利用其存储和计算能力将使得小中型企业有能力提供 PSI 服务。Kamara 等人提出了借助云服务器来分摊参与方的计算开销、借助 BF 来降低通信开销的多方 PSI 方案[10]。Debnath 等人基于 BF 和 ElGamal 同态加密技术实现了在半诚实敌手模型下的安全多方 PSI-CA 协议。尽管如此，在传统的大部分 PSI 方案中，云服务商直接将用户存储到云端的所有数据与所有的参与方进行 PSI，这样势必会存在恶意参与方利用 PSI 来推断其他参与方数据的问题。另一方面，[11]指出由于健康信息大部分是电子共享的，这极大地提高了基因数据滥用的可能性，这种完全无需用户授权就可以肆意操作用户所有数据的模式给用户的隐私保护带来了巨大的挑战。一个用户可以指定操作数据集、求交参与方的模式是用户所期待的。密文等值测试的技

术可以解决这个挑战，等值测试可以解决多公钥加密数据比较问题，即使加密使用的公钥不同，用户也可以在不进行解密的前提下，比较两段密文对应的明文是否相同[12]。Yang 等首次提出了密文等值测试公钥加密技术(PKEET, public key encryption with equality test) [13]，此后[14] [15] [16]对等值测试技术的进行改进，在方案中加入了授权生成算法，并将授权模式拓展到 6 种。实现了用户指定求交数据、求交参与方的目的[17]。利用双线性对技术构造 APSI 来实现可授权外包 PSI，[15]利用 SPHF (smooth projective hash function)构造 PKE-OET 方案实现可授权外包 PSI 求交在物联网场景下的应用。但是二者均没有多方的 PSI 方案，无论是在物联网多用户数据求交，还是在医院多用户基因序列比对都需要多方的 PSI 方案。

因此，本文基于[17]，利用密文等值测试的方法构造一个多方可授权的外包 PSI 方案。本文的贡献如下：

(1) 提出了一个基于密文等值测试的多方隐私集合求交方案 MAPSI (Multi-party Authorized Private Set Intersection over Outsoursed Encrypted Datasets)，该方案利用密文等值测试中不同授权模式的特性将方案拓展为多方。不同于传统基于同态的多方 PSI 方案，本方案可以实现不同公钥加密下数据的隐私集合求交。

(2) 针对外包密文等值测试的性质，本文提出了一种新的多方 PSI 模型。降低了多方隐私集合求交的通信和计算代价。

2. 预备知识

(1) 授权模式：

在 MAPSI 中，通过确定不同的授权模式，用户可以指定参与集合求交的数据集和用户。用户可以使用以下四种授权方式来指定云服务器的求交模式。本方案中，我们称第一个向云服务器发送求交请求的参与方为 *leader*，其他的统称为参与方。

(1) 用户级别授权：(多对多)该授权模式下，云服务器可以将用户的所有数据和任意参与方进行集合求交操作。

(2) 密文级别授权：(一对多)该授权模式下，用户会指定部分外包数据，参与方也指定部分数据，云服务器只能对用户指定的数据和任意的参与方指定的数据进行集合求交操作。

(3) 用户指定级别授权：(指定多对多)在该授权模式下，用户不限制求交的数据集合，而是指定参与方，云服务器只能对用户的数据和指定的参与方数据进行集合求交操作。

(4) 密文指定级别授权：(指定一对一)在该授权模式下，用户不仅限制求交的数据集合同时指定参与方，(另一方面)参与方也对参与求交的数据进行限制，云服务商只能对用户指定的外包数据与指定参与方的特定数据进行集合求交操作。

(2) 跨授权类型比较：

用户使用的授权类型可能是不同的。例如，一个用户希望限定自己进行求交的数据集，对求交的对象没有特殊的要求，也就是所谓的密文级别授权。而其他用户不仅会限制求交的数据集也会指定求交的参与方，也就是所谓的密文指定级别授权。只有当参与方相互满足授权要求时，云服务器才能对各参与方进行集合求交。因为针对不同用户而言，信息的价值不同，正是因为参与方的多样性使得跨授权类型比较是必须的。

(3) OPSI (Outsourced PSI)：

数据拥有者限于自身条件，将数据外包到云服务器上，当需要进行集合求交操作时，数据拥有者向云服务器发送请求，并将自己数据的授权陷门发送给云服务器。云服务器对数据进行计算后，将计算结果发送给所有参与方。参与方在本地执行解密操作得到集合求交的结果。

(4) Bilinear Pairing：

设 G_1,G_2,G_T 是以 p 为素数阶的三个循环群，对于所有的 $g_1 \in G_1, g_2 \in G_2, a,b \in Z_p$，双线性映射 $\hat{e}: G_1 \times G_2 \to G_T$ 满足以下特性：

- 双线性：$\hat{e}\left(g_1^a, g_2^b\right) = \hat{e}\left(g_1, g_2\right)^{ab}$；
- 非退化性：$\hat{e}\left(g_1, g_2\right) \neq 1$；
- 可计算性：$\hat{e}\left(g_1, g_2\right)$ 是可以进行高效计算的。

(5) Bilinear Diffie-Hellman Assumption

设 $G_1 \le g_1 >, G_2 \le g_2 >, G_T$ 是以 p 为素数阶的循环群，$\hat{e}: G_1 \times G_2 \to G_T$ 是一个双线性映射，*BDH* 假设：给定一个元组 $\left(g_1, g_1^a, g_1^b, g_2, g_2^c\right)$，其中 $(a,b,c) \overset{R}{\leftarrow} Z_p^*$，对于任意多项式时间的算法 A 能够成功计算 $\hat{e}\left(g_1, g_2\right)^{abc}$ 概率是可以忽略的。

(6) Decisional Bilinear Diffie-Hellman Assumption

设 $G_1 \le g_1 >, G_2 \le g_2 > G_T$ 是以 p 为素数阶的循环群，$\hat{e}: G_1 \times G_2 \to G_T$ 是一个双线性映射，*DBDH* 假设：给定一个元组 $\left(g_1, g_1^a, g_1^b, g_2, g_2^c, Z\right)$，其中 $(a,b,c) \overset{R}{\leftarrow} Z_p^*$，对于任意多项式时间的算法 A 能够成功判断 $\hat{e}\left(g_1, g_2\right)^{abc} = Z$ 的概率是可以忽略的。

(7) Yuanhao Wang 的 APSI 方案介绍

我们的工作基于 Yuanhao Wang 的双方 APSI 方案[17]，每个用户拥有自己的公私钥对。

$$(pk, sk) = \left(\left(g_1^x, g_1^y, g_1^z, g_2^z,\right),(x,y,z)\right)$$

同时利用三个哈希函数来保证云服务器进行隐私计算时的信息安全问题。方案利用双线性对幂指数可以交换的特性实现云服务器在不清楚明文信息的情况下进行数据计算，通过将密文参数和求交方私钥添加到授权陷门的产生函数中，实现用户指定求交密文和求交对象的功能。云服务器利用数据拥有者的授权陷门和密文信息可以还原出明文的哈希值或含哈希值的双线性对值。通过比较两个用户的明文哈希或含哈希值的双线性对值就可以确定双方的明文值是否相等。

方案的核心是利用双线性对幂指数可以交换的特性确保了私钥和授权陷门的区别，使得云服务器可以进行密文数据的处理，但无法解密得到用户明文。通过授权陷门之间的差别实现不同的授权模式。

3. MAPSI 方案设计

在这个部分，我们提出了 MAPSI 的系统模型，介绍了该方案的具体构造流程，并利用一个实例介绍整个方案的执行过程。

方案的总体思路：利用密文等值测试可以实现不同密钥加密的明文进行隐私比较的特性，确保不同的参与方在利用自己的私钥加密明文的情况下可以进行隐私比较。此外，利用等值测试可以针对密文进行访问控制，确保求交集合结果有记忆性，实现下一次隐私求交的结果是在上次结果集合的基础上进行的，从而实现多个参与方进行隐私集合求交的操作。此外，该方案中对参与求交的用户进行访问控制给予了用户更多的数据控制权限，用户可以指定隐私求交的对象进一步提高了用户的隐私安全。正因为以上特性，用户仅使用授权陷门就可以控制数据进行隐私集合求交。反复对中间求交结果和新的数据进行集合求交，最终返回所有参与方的公共部分。

利用密文等值测试方案判断不同公钥加密的密文对应的明文是否相同。传统的基于同态的 PSI 方案往往需要使用相同的公钥进行加密，该场景下，往往需要共同协商一个公私钥对或者使用指定用户的公私钥对，造成一定的通信和计算开销，然而，现实中的应用场景往往是用户利用自己的私钥进行数据加密，然后利用云服务器的计算和存储能力来进行隐私集合求交。利用授权陷门，不仅能实现对用户数据的访问控制，同时也为多用户求解 集合交集提供了技术支持。

3.1. 方案具体流程

一：云服务器执行等值测试(双方)流程：

1. 初始化和密钥的产生：

(1) $Setup\left(1^{\lambda}\right)$:输入安全参数$1^{\lambda}$，该算法输出系统的公共参数，如下：

－选择阶数为 p 的群 G_1, G_2, G_T (p 是一个素数)，其中 g_1 是 G_1 的生成元，g_2 是 G_2 的生成元，一个双线性运算 $\hat{e}$：$G_1 \times G_2 \to G_T$，随机选择 $u, w \xleftarrow{R} G_2$。

－选择三个哈希函数 $H_1:\{0,1\}^* \to G_1, H_2:G_T \to \{0,1\}^{l_1}, H_3:\{0,1\}^* \to \{0,1\}^{l_m+l_z}$，其中 l_1 表示 G_1 的元素长度，l_m 表示数据的最大长度，l_z 代表 Z_p^* 中元素的长度。

－设置公共参数 $pp=\left(G_1, G_2, G_T, p, g_1, g_2, u, w, \hat{e}, H_1, H_2, H_3\right)$。

(2) $KeyGen(pp)$：输入公共 pp，该算法选择随机数 $x, y, z \xleftarrow{R} Z_p^*$，输出 $(pk, sk)=\left(\left(g_1^x, g_1^y, g_1^z, g_2^z\right),(x, y, z)\right)$。

2. 加解密流程：

(1) $Enc(pk, D)$：输入数据集合 $D=\left(D_1, \cdots, D_n\right)$ 和公钥 pk，其中 $D_i \in M$，输出的密文集合 C 结果如下：

－对于 $D_i \in M$，随机选择 $r_{i,1}, r_{i,2} \xleftarrow{R} Z_p^*$，随后产生密文集合。

$$C_i=\left(C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}\right):$$

$$C_{i,1}=g_1^{r_i,1}, \quad C_{i,2}=g_1^{r_i,2},$$

$$C_{i,3}=\left(D_i \| r_{i,1}\right) \oplus H_3\left(\hat{e}\left(g_1^x, u\right)^{r_{i,2}} \| C_{i,1} \| C_{i,2} \| C_{i,4}\right),$$

$$C_{i,4}=H_1\left(D_i\right) \cdot\left(g_1^z\right)^{r_{i,1}} \oplus H_2\left(\hat{e}\left(g_1^y, w\right)^{r_{i,2}}\right)$$

－ $Set \quad C=\left(C_1, \cdots, C_n\right)$

(2) $Dec(sk, C)$：输入一个密文集合 $C=\left(C_1, \cdots, C_n\right)$ 和私钥 sk，算法输出数据集 D 如下：

－对于任意的 $C_i\left(1 \le i \le n\right)$，计算 $\left(D_i \| r_{i,1}\right)=C_{i,3} \oplus H_3\left(\hat{e}\left(C_{i,2}, u\right)^x \| C_{i,1} \| C_{i,2} \| C_{i,4}\right)$。

当 $g_1^{i,1}=\left(C_{i,1}\right)$ 且 $H_1\left(D_i\right) \cdot\left(C_{i,1}^z\right) \oplus H_2\left(\hat{e}\left(C_{i,2}, w\right)^y\right)=C_{i,4}$，输出结果 D_i 否则返回 $\perp$ 表示解密失败。

－返回结果集合 $Set \quad D=\left(D_1, \cdots D_n\right)$。

3. 陷门的产生：

用户可以产生四种类型的授权陷门来管理自己的数据，云服务器利用授权陷门对相应的授权数据进行集合求交操作。授权陷门有如下四种。

1) 第一种授权陷门：所有的用户密文可以和任意用户的密文进行比较。

$Aut_1\left(sk_A\right)$:输入用户 A 的私钥 sk_A，算法输出第一种类型的陷门。

$$td_{1,A}=\left(td_{1,A}^1, td_{1,A}^2\right)=\left(w^{y_A}, z_A\right)$$

注：下标 $_1$ 表示第一种类型的授权陷门，下标 $_A$ 指明哪个用户的授权陷门。

2) 第二种授权陷门：所有的用户密文可以和指定用户的密文进行比较。

$Aut_2\left(sk_A, pk_B\right)$:输入用户 A 的私钥 sk_A 和用户 B 的公钥 pk_B，算法输出第二种类型的陷门。

$$td_{2,A \to B}=\left(td_{2,A \to B}^1, td_{2,A \to B}^2, td_{2,A \to B}^3\right)=\left(w^{y_A},\left(g_2^{Z_B}\right)^{Z_A},\left(g_2^{Z_B}\right)^{Z_A \cdot Z_A}\right)$$

注：下标 $A \to B$ 表示用户 A 对用户 B 的授权陷门。

1) 第三种授权陷门：用户可以指定某些密文，这些密文允许和任意用户的密文进行比较。

$Aut_3\left(sk_A, C_i\right)$：输入用户 A 的私钥 sk_A 和密文 C_i，输出第三种类型的陷门。

$$td_{3,A,C_i} = \left(td^1_{3,A,C_i}, td^2_{3,A,C_i}\right) = \left(C_{i,2}{}^{y_A}, z_A\right) = \left(g_1^{r_i \cdot 2 \cdot y_A}, z_A\right)$$

注：下标 A, C_i 表示用户 A 针对密文 C_i 的授权陷门。

2) 第四种授权陷门：用户可以指定某些密文，这些密文只能和指定用户的密文进行比较。

$Aut_4\left(sk_A, pk_B, C_i\right)$：输入用户 A 的私钥 sk_A 和用户 B 的公钥 pk_B，算法输出第四种类型的陷门。

$$\begin{aligned} td_{4,A\to B,C_i} &= \left(td^1_{4,A\to B,C_i}, td^2_{4,A\to B,C_i}, td^3_{4,A\to B,C_i}\right) \\ &= \left(C_{i,2}{}^{y_A}, \left(g_2^{z_B}\right)^{z_A}, \left(g_2^{z_B}\right)^{z_A \cdot z_A}\right) \\ &= \left(g_1^{r_{i,2} \cdot y_A}, \left(g_2^{z_B}\right)^{z_A}, \left(g_2^{z_B}\right)^{z_A \cdot z_A}\right) \end{aligned}$$

云服务器在接收到用户陷门后查找陷门和测试算法的对照表，选择合适的测试算法进行等值测试，对照表见表 1。

Table 1. Authorization trapdoor and equivalence test algorithm comparison table
表 1. 授权陷门与等值测试算法对照表

		用户 A 的授权方式			
		第一种陷门	第二种陷门	第三种陷门	第四种陷门
用户 B 的授权方式	第一种陷门	Test1	$Test_2$	$Test_1$	$Test_2$
	第二种陷门	Test2	$Test_2$	$Test_2$	$Test_2$
	第三种陷门	Test1	$Test_2$	$Test_1$	$Test_2$
	第四种陷门	Test2	$Test_2$	$Test_2$	$Test_2$

4. 等值测试算法：

针对不同的参与方的数据授权方式，云服务器执行不同的等值测试算法。方法主要有以下两类：

- $Test_1\left(C_A, td_A, C_B, td_B\right)$：输入两个密文集合 C_A, C_B 和其相应的授权陷门 td_A, td_B，针对不同的授权陷门输出的结果如下：

对于任何密文 $C_i \in C_A\left(1 \le i \le n\right)$：

- 如果 td_A 是第一种类型的授权陷门 $td_{1,A}$：

输出结果：

$$T_i = \frac{C_{i,4} \oplus H_2\left(\hat{e}\left(C_{i,2}, td^1_{1,A}\right)\right)}{C_{i,1}{}^{td^2_{1,A}}} = H_1\left(D_i\right)$$

- 如果 td_A 是第三种类型的授权陷门 $td_{3,A} = \left\{td_{3,A,C_i} \middle| 1 \le i \le n\right\}$：

输出结果：

$$T_i = \frac{C_{i,4} \oplus H_2\left(\hat{e}\left(td^1_{3,A,C_i}, w\right)\right)}{C_{i,1}{}^{td^2_{3,A,C_i}}} = H_1\left(D_i\right)$$

对于任何密文 $C_j \in C_B\left(1 \le j \le m\right)$：

- 如果 td_B 是第一种类型的授权陷门 $td_{1,B}$：

输出结果：

$$T_i = \frac{C_{j,4} \oplus H_2\left(\hat{e}\left(C_{j,2}, td_{1,B}^1\right)\right)}{C_{j,1}^{td_{1,B}^2}} = H_1\left(D_j\right)$$

- 如果 td_A 是第三种类型的授权陷门 $td_{3,B} = \left\{td_{3,B,C_j} \middle| 1 \le j \le m\right\}$：

输出结果：

$$T_j = \frac{C_{j,4} \oplus H_2\left(\hat{e}\left(td_{3,B,C_j}^1, w\right)\right)}{C_{j,1}^{td_{3,B,C_j}^2}} = H_1\left(D_j\right)$$

设 $T_A = \left\{T_i \middle| 1 \le i \le n\right\}, T_B = \left\{T_j \middle| 1 \le j \le m\right\}$ 分别为用户 A 和用户 B 授权陷门所产生的测试结果集，计算 $T_A \cap T_B$，返回结果集 $L_A = \left\{C_i \middle| C_i \in C_A, T_i \in T_A \cap T_B\right\}$ 给 $User_A$，返回结果集 $L_B = \left\{C_j \middle| C_j \in C_B, T_j \in T_A \cap T_B\right\}$ 给 $User_B$。

- $Test_2\left(C_A, td_A, C_B, td_B\right)$：输入两个密文集合 C_A, C_B 和其相应的授权陷门 td_A, td_B，针对不同的授权陷门输出的结果如下：

对于任意 $C_i \in C_A \left(1 \le i \le n\right)$：

- 如果 td_A 是第一种类型的授权陷门 $td_{1,A}$：

$$S_i = \frac{C_{i,4} \oplus H_2\left(\hat{e}\left(C_{i,2}, td_{1,A}^1\right)\right)}{C_{i,1}^{td_{1,A}^2}} = H_1\left(D_i\right),$$

$$T_i = \hat{e}\left(S_i, g_2^{Z_B}\right)^{td_{1,A}^2} = \hat{e}\left(H_1\left(D_i\right), g_2\right)^{Z_A \cdot Z_B}$$

- 如果 td_A 是第二种类型的授权陷门 $td_{2,A\to B}$：

$$S = C_{i,4} \oplus H_2\left(\hat{e}\left(C_{i,2}, td_{2,A\to B}^1\right)\right) = H_1\left(D_i\right) \cdot \left(g_1^{Z_A}\right)^{r_{i,1}},$$

$$T_i = \frac{\hat{e}\left(S_i, td_{2,A\to B}^2\right)}{\hat{e}\left(C_{i,1}, td_{2,A\to B}^3\right)} = \hat{e}\left(H_1\left(D_i\right), g_2\right)^{Z_A \cdot Z_B}$$

- 如果 td_A 是第三种类型的授权陷门 $td_{3,A} = \left\{td_{3,A,C_i} \middle| 1 \le i \le n\right\}$：

$$S_i = \frac{C_{i,4} \oplus H_2\left(\hat{e}\left(td_{3,A,C_i}^1, w\right)\right)}{C_{i,1}^{td_{3,A,C_i}^2}} = H_1\left(D_i\right),$$

$$T_i = \hat{e}\left(S_i, g_2^{Z_B}\right)^{td_{3,A}^2} = \hat{e}\left(H_1\left(D_i\right), g_2\right)^{Z_A \cdot Z_B}$$

- 如果 td_A 是第四种类型的授权陷门 $td_{4,A\to B} = \left\{td_{4,A\to B,C_i} \middle| 1 \le i \le n\right\}$：

$$S_i = C_{i,4} \oplus H_2\left(\hat{e}\left(td_{4,A\to B,C_i}^1, w\right)\right) = H_1\left(D_i\right) \cdot \left(g_1^{Z_A}\right)^{r_{i,1}},$$

$$T_i = \frac{\hat{e}\left(S_i, td_{4,A\to B,C_i}^2\right)}{\hat{e}\left(C_{i,1}, td_{4,A\to B.C_i}^3\right)} = \hat{e}\left(H_1\left(D_i\right), g_2\right)^{Z_A \cdot Z_B}$$

对于任意 $C_j \in C_B \left(1 \le j \le m\right)$：

- 如果 td_B 是第一种类型的授权陷门 $td_{1,B}$：

$$S_j=\frac{C_{j,4}\oplus H_2\left(\hat{e}\left(C_{j,2},td_{1,B}^1\right)\right)}{C_{j,1}^{td_{1,B}^2}}=H_1\left(D_j\right),$$

$$T_j=\hat{e}\left(S_j,g_2^{Z_A}\right)^{td_{1,B}^2}=\hat{e}\left(H_1\left(D_j\right),g_2\right)^{Z_A\cdot Z_B}$$

- 如果 td_A 是第二种类型的授权陷门 $td_{2,A\to B}$：

$$S_j=C_{j,4}\oplus H_2\left(\hat{e}\left(C_{j,2},td_{2,B\to A}^1\right)\right)=H_1\left(D_j\right)\cdot\left(g_1^{Z_B}\right)^{r_{j,1}},$$

$$T_j=\frac{\hat{e}\left(S_j,td_{2,B\to A}^2\right)}{\hat{e}\left(C_{j,1},td_{2,B\to A}^3\right)}=\hat{e}\left(H_1\left(D_j\right),g_2\right)^{Z_A\cdot Z_B}$$

- 如果 td_A 是第三种类型的授权陷门 $td_{3,B}=\left\{td_{3,B,C_j}\,\middle|\,1\le j\le m\right\}$：

$$S_j=\frac{C_{j,4}\oplus H_2\left(\hat{e}\left(td_{3,B,C_j}^1,w\right)\right)}{C_{j,1}^{td_{3,B,C_j}^2}}=H_1\left(D_j\right),$$

$$T_j=\hat{e}\left(S_j,g_2^{Z_A}\right)^{td_{3,B}^2}=\hat{e}\left(H_1\left(D_j\right),g_2\right)^{Z_A\cdot Z_B}$$

- 如果 td_A 是第四种类型的授权陷门 $td_{4,B\to A}=\left\{td_{4,B\to A,C_j}\,\middle|\,1\le j\le m\right\}$：

$$S_j=C_{j,4}\oplus H_2\left(\hat{e}\left(td_{4,B\to A,C_j}^1,w\right)\right)=H_1\left(D_j\right)\cdot\left(g_1^{Z_B}\right)^{r_{j,1}},$$

$$T_j=\frac{\hat{e}\left(S_j,td_{4,B\to A,C_j}^2\right)}{\hat{e}\left(C_{j,1},td_{4,B\to A,C_j}^3\right)}=\hat{e}\left(H_1\left(D_j\right),g_2\right)^{Z_A\cdot Z_B}$$

设 $T_A=\left\{T_i\,\middle|\,1\le i\le n\right\},T_B=\left\{T_j\,\middle|\,1\le j\le m\right\}$ 分别为用户 A 和用户 B 授权陷门所产生的测试结果集，计算 $T_A\cap T_B$，返回结果集 $L_A=\left\{C_i\,\middle|\,C_i\in C_A,T_i\in T_A\cap T_B\right\}$ 给 $User_A$，返回结果集 $L_B=\left\{C_j\,\middle|\,C_j\in C_B,T_j\in T_A\cap T_B\right\}$ 给 $User_B$。

二：多方隐私集合求交流程：

该方案主要涉及以下两类实体：一个云服务器，用来存储用户数据以及执行密文等值测试操作，m 个参与方，表示为集合 $\left(User_1,\cdots User_m\right)$，每个用户拥有 n 个数据集合 $D=\left(D_1,\cdots D_n\right)$。

1. 初始化和密文产生。

1) 使用 $Setup\left(1^\lambda\right)$ 产生系统的公共参数 $pp=\left(G_1,G_2,G_T,p,g_1,g_2,u,w,\hat{e},H_1,H_2.H_3\right)$。

2) 用户 i 使用 $KeyGen\left(pp\right)$ 产生自己的公私钥对 $\left(pk_i,sk_i\right)=\left(\left(g_1^{x_i},g_1^{y_i},g_1^{z_i},g_2^{z_i}\right),\left(x_i,y_i,z_i\right)\right)$。

3) 用户 i 用自己的私钥 pk_i 将明文 D_i 加密得到密文 $c_i=Enc\left(D_i,pk_i\right)$，并将自己的密文上传到云服务器。

2. 等值测试流程。

为了简化方案流程，我们假设用户依次向云服务器发送陷门来进行等值测试操作，并且不考虑用户对密文数据的授权，假设用户不会对求交的角色进行授权限制(对授权角色进行限制仅仅增加了无效等值测试次数，即不使用第二类和第四类授权陷门)。此外，本方案中，我们将第一个参与方 $User_1$ 作为 $leader$，$leader$ 相较于其他参与方会多次发送陷门给云服务器，以帮助云服务器反复进行等值测试操作，其他参与方只需进行一次等值测试操作，只需要给云服务器发送一次授权陷门。为了方便区分，我们使用下标 $A,B\ldots$ 来代替 $User_i$ 的下标 $1,2\ldots$。

1) 首先$User_1$利用陷门产生算法产生第一类授权陷门$ltd_1 = td_{1,A} = \left(td_{1,A}^1, td_{1,A}^2\right) = \left(w^{y_A}, z_A\right)$，其中$ltd_i$的下标表示*leader*第$i$次发送的授权陷门。同理，$User_2$也产生其对应的第一类授权陷门$td_{1,B} = \left(td_{1,B}^1, td_{1,B}^2\right) = \left(w^{y_B}, z_B\right)$，发送给云服务器。云服务器接收到*leader*和$User_2$的授权陷门后进行等值测试操作，返回密文等值测试的结果给*leader*。

2) *leader*接收到云服务器返回的测试结果之后，针对不同的密文数据产生相应的第三类授权陷门$\left\{ltd_2\right\}_{C_i} = td_{3,A,C_i} = \left(td_{3,A,C_i}^1, td_{3,A,C_i}^2\right) = \left(C_{i,2}{}^{y_A}, z_A\right) = \left(g_1^{r_i \cdot 2 \cdot y_A}, z_A\right)$，同时，$User_3$产生第一类授权陷门$td_{1,C} = \left(td_{1,C}^1, td_{1,C}^2\right) = \left(w^{y_C}, z_C\right)$并发送给云服务器，云服务器接收到来自*leader*和$User_3$的授权陷门后，对两者的数据进行密文等值测试操作。由于*leader*仅会对求交集合的结果进行授权，这就使得最终的返回结果为*leader*,$User_1$,$User_2$的共同结果集合。

3)以此类推，$User_i$依次将自己的授权陷门上传给云服务器，同时*leader*依次产生相应的第三类授权陷门给云服务器，云服务器对数据进行等值测试操作，直到最后一个用户完成等值测试操作，云服务器返回的等值测试结果为所有参与方共同的数据集合。

3.2. 方案数据流

首先，云服务器的数据流流入方面：为了利用云服务器的存储和计算能力。如图 1，2，在方案的初始化阶段，用户需要将自己的数据进行加密，将加密后的密文数据流传输到云服务器上。方案的执行阶段，用户将自己的授权陷门数据流上传给云服务器。云服务器的数据流流出方面，云服务器完成等值测试操作后，将等值测试的结果数据流返回给用户。用户数据流流出方面，该方案中，在目前的假设前提之下，有两类用户，一类用户为*leader*，*leader*流出数据流有首次用于隐私集合求交的第一类授权陷门，和以后用于求交的其他种类的授权陷门。另一类用户为普通用户$\left\{User_2、User_3……\right\}$，该类型用户仅会向用户上传一次第一种类型的授权陷门。用户数据流流入方面，对于*leader*而言，云服务器会将其每次等值测试的结果集合返回给*leader*。对于普通用户而言，云服务器会将最终等值测试的结果集合返回给所有的用户。

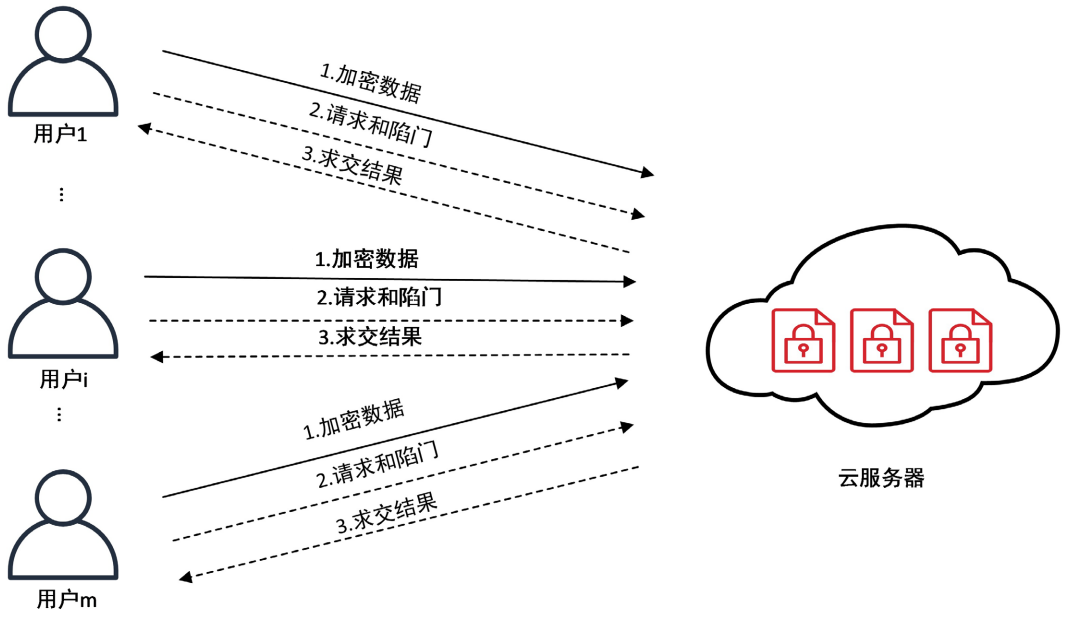

Figure 1. System model of MAPSI
图 1. MAPSI 的系统模型

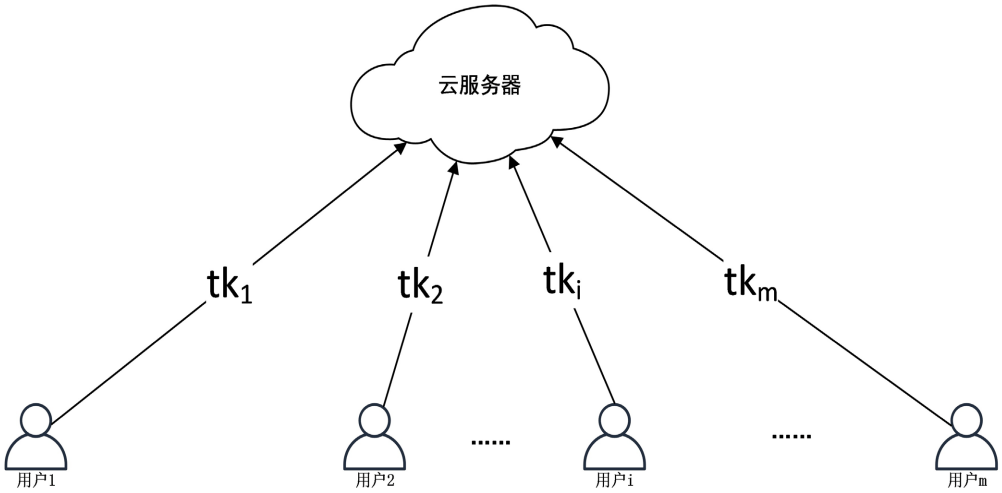

Figure 2. MAPSI communication model
图 2. MAPSI 的通信模型

本方案是一种外包的隐私集合求交协议(OPSI)协议，在进行多方隐私集合求交之前，用户会先将数据用自己的私钥加密后存储到云服务器。由于存储的数据是经过私钥加密过的密文形式，云服务器没有用户私钥，就无法获得用户的明文信息。该操作实现了在利用云服务器的存储功能的同时保障了用户的隐私。另一方面，用户使用授权陷门的方式来实现对数据访问权限的管理，保证云服务器无法随意地(利用任意用户数据与任意参与方执行集合求交操作)对用户数据进行集合求交。同时，在保证用户对外包到云服务器的数据进行授权管理的同时，也赋予了云服务器利用授权陷门进行等值测试的能力。

综上所述，该方案利用了云服务器的存储和计算能力，保证了在隐私集合求交的过程中，用户仅需向服务器发送一次数据，而且用户无需在本地承担任何隐私计算的压力。极大地降低了隐私集合求交的通信和参与方本地的计算代价。此外，多种的授权陷门也为用户的隐私数据管理提供了更大的灵活性。

4. 安全性证明

1. 安全模型

本文在半诚实模型下证明方案的安全性。*leader* 按照方案规定的流程依次产生授权陷门 ltd_i 发送给云服务器，并针对云服务器返回的结果产生新的授权陷门。同时，*leader* 对其他用户的私有数据保持好奇，并试图推断其他用户的私有数据。普通用户 $User_i$ 同样按照方案规定的流程产生针对自身数据的授权陷门 *td* 并发送给云服务器，但同时对其他用户和 *leader* 的私有数据保持好奇，并试图推断他们的私有数据。云服务器按照方案规定的流程；利用用户上传的授权陷门对用户密文数据进行密文等值测试，并将测试结果发送给 *leader* 。但同时对用户求交结果的明文数据和用户私有数据保持好奇。

2. 安全性分析

A. 内部攻击者：

定理 1：如果 *BDH* 假设和 *DBDH* 假设成立， *MAPSI* 方案针对内部攻击者是安全的。

内部攻击者主要有两类，普通用户和 *leader* ，首先证明方案针对 *leader* 是安全的。本方案中，为了降低通信代价，要求 *leader* 进行多次授权陷门的上传操作，这使得 *leader* 相较于普通用户而言拥有更大的权限，同样使得本方案针对 *leader* 的安全性相对较低，由于 *leader* 可以获得每次等值测试的求交结果。

如果在一次等值测试完成后， *leader* 交集元素相较于以前减少了，则 *leader* 可以推断本次与之求交的用户是否拥有减少的元素，但是 *leader* 在没有云服务器的辅助之下无法确定与之比较的用户。即使在云服务器的辅助下， *leader* 能够确定与之求交的用户及求交后的结果，但是对于 *leader* 不拥有，而其他用户拥有的数据。 *leader* 需要进行明文的穷举来推测与之比较的用户数据，如果 *BDH* 假设和 *DBDH* 假设成立， *leader* 能够区分随机二进制字符流 $W_3=\{0,1\}^{l_m+l_z}$ (l_m+l_z 等于 $C_{i,3}$ 的二进制长度) 和密文 $C_{i,3}=\left(D_t\|r_{t,1}\right)\oplus H_3\left(\hat{e}\left(g_1^{x_s},u\right)^{r_{t,2}}\|C_{t,1}\|C_{t.2}\|C_{t,4}\right)$ 的可能性可以忽略不计，详细证明参考[17]。 *leader* 推断此类数据相当于挑战 *APSI* 的 *OW* − *CCA* 安全性。

证明方案针对普通用户是安全的。即普通用户无法推测 *leader* 和其他用户的非公共数据。虽然普通用户可以穷举数据进行密文等值测试，但是其无法得到等值测试的结果，如果普通用户之间不共谋，所有普通用户同时猜测相同明文数据的可能性是可以忽略不计的，最终集合求交的结果中含有猜测明文数据的可能性同样是可以忽略不计的。即使在所有其他用户共谋的情况下，该场景下普通用户相当于拥有了 *leader* 的权限，能够在云服务器的辅助下得到集合求交的结果，证明同 *leader* 推测其他用户的非公共数据。所以即使在该场景下普通用户推测 *leader* 的非公共数据相当于挑战 *APSI* 的 *OW* − *CCA* 安全性。

普通用户无法推测其他用户的非公共数据，由于方案执行的过程中，普通用户无法得到等值测试的中间结果，只有与 *leader* 共谋才可确定中间结果。即使在共谋的情况下，普通用户推测其他用户的非公共数据也是一种穷举攻击，相当于挑战 *APSI* 的 *OW* − *CCA* 安全性。

B. 外部攻击者

定理 2：如果 *BDH* 假设和 *DBDH* 假设成立， *MAPSI* 方案针对外部攻击者是安全的。

没有授权陷门，攻击者无法执行 *Test* 来区分两个挑战密文。就外部攻击者而言，在密文中，$C_{i,1}$ 和 $C_{i,2}$ 都是随机元素。如果 *DBDH* 问题是困难的，就外部攻击者而言，$C_{i,3}$ 中的 $\hat{e}\left(g_1^x,u\right)^{r_{i,2}}$ 和 $C_{i,4}$ 中的 $\hat{e}\left(g_1^y,w\right)^{r_{i,2}}$ 都是随机的，所以 $C_{i,3}$ 和 $C_{i,4}$ 也是随机的。因为以上密文都是信息异或结果或者随机数据的哈希值，外部攻击者无法得到任何信息来辅助他区分两个挑战密文。所以 MAPSI 方案针对外部攻击者是 *OW* − *CCA* 安全的。

C. 半诚实云服务器

定理 3：如果 *BDH* 假设和 *DBDH* 假设成立， *MAPSI* 方案针对半诚实云服务器是安全的。

云服务器无法通过已知的信息推断出任何参与方的数据，云服务器只能得到参与比较数据的哈希值。因为云服务器可以执行密文等值测试来验证数据的哈希值，这使得云服务器可能通过暴力破解的方式来推测参与方的明文数据。但是，我们通常认为信息空间足够大(安全参数达到指数级，信息空间的数据均匀分布)，这使得云服务器利用暴力破解的方式挑战该方案是不可行的。如果 *BDH* 假设和 *DBDH* 假设成立，云服务器区分随机二进制数据流 $\{0,1\}^{l_n}$ (l_n 等于 $H_i\left(D_i\right)$ 的二进制长度)和 $H_i\left(D_i\right)$ 的可能性是可以忽略不计的。所以该方案对于半诚实云服务器而言是 *OW* − *CCA* 安全的。

5. 性能分析

本方案中，用户的数据以密文的形式存储在云服务器中，在进行隐私集合求交的过程中，只涉及到授权陷门和求交结果的通信，此外，方案中大部分计算代价由云服务器承担。

表 2 中将本方案与目前常见的几种基于公钥的多方 PSI 方案进行了比较。从表中可以看出，由于二进制授权陷门大小 $\log_2|t|$ 和二进制求交结果大小 $\log_2|r|$ 远小于二进制密文大小 $\log_2|x|$，所以 *MAPSI* 方案在复杂度对比方面 ，无论是 *leader* 方，还是 *Client* 方，在通信复杂度和计算复杂度方面相较于目前半诚实安全模型下的公钥多方 PSI 中有较好的性能。

Table 2. Comparison of multi-party protocols based on public keys
表 2. 基于公钥的多方 PSI 协议比较

协议	安全性	抗共谋	通信复杂度		计算复杂度	
			Leader	*Client*	*Leader*	*Client*
文献[18]	半诚实	√	$O\left(n^2m^2\lambda\right)$	$O\left(n^2m^2\lambda\right)$	$O\left(n^2m+n\lambda m^2\right)$	$O\left(n^2m+n\lambda m^2\right)$
文献[19]	半诚实	√	$O\left(nm\lambda\right)$	$O\left(m\lambda\right)$	$O\left(nm\log_2 mk\right)$	$O\left(mk\right)$
文献[20]	半诚实	√	$O\left(dn\log_2\lvert x\rvert\right)$	$O\left(d\log_2\lvert x\rvert\right)$	$O\left(d\right)$	$O\left(d\right)$
文献[8]	半诚实	√	$O\left(nd\right)$	$O\left(d\right)$	$O\left(nd\right)$	$O\left(d\right)$
本方案	半诚实	√	$O\begin{pmatrix}nm\log_2\lvert t\rvert\\+nm\log_2\lvert r\rvert\end{pmatrix}$	$O\left(m\log_2\lvert t\rvert\right)$	$O\left(nm\right)$	$O\left(m\right)$

注：在复杂度对比中，其中 n 为参与方的数目，m 为集合大小，λ 为安全参数，d 为域的大小，$\log_2\lvert x\rvert$ 为二进制密文 x 的大小，$\log_2\lvert t\rvert$ 为二进制授权陷门 t 的大小，$\log_2\lvert r\rvert$ 为二进制求交结果 r 的大小。

6. 结论

本文探索了一种利用密文等值测试技术实现多方 PSI 的方法，并基于 *APSI* 方案[17]构造了一个多方 PSI 的方案，不同于传统的基于同态技术实现的多方 PSI，本方案可以对不同公钥加密的密文数据进行多方 PSI 操作，这更适应现实生活中个人用自身公钥加密自身数据的应用场景。另一方面，利用密文等值测试技术实现多方 PSI，用户可以利用授权陷门进一步限制 PSI 操作对数据的访问。增强了用户对自身数据的控制能力，进一步保护了用户的隐私。此外，本方案利用云服务器来承担多方 PSI 带来的计算和存储开销，减轻了用户的计算和存储压力。最后，本文从通信复杂度和计算复杂度的角度比较了该方案与现有的基于公钥的多方 PSI 方案，该方案表现出了较好的性能。

基金项目

国家自然科学基金(61370188)；北京市教委科研计划(KM202010015009)；北京市教委科研计划资助(No.KM202110015004)；北京印刷学院博士启动金项目(27170120003/020)；北京印刷学院科研创新团队项目(Eb202101)；北京印刷学院校内学科建设项目(21090121021)；北京印刷学院重点教改项目(22150121033/009)；北京印刷学院科研基础研究一般项目(Ec202201)；北京印刷学院博士启动金项目(27170122006)；北京印刷学院基础研究一般项目(Ec202201)；北京市高等教育学会 2022 年立项面上课题(MS2022093)；北京市教育委员会科学研究计划项目资助(KM202310015002)。

参考文献

[1] Meadows, C. (1986) A More Efficient Cryptographic Matchmaking Protocol for Use in the Absence of a Continuously Available Third Party. 1986 *IEEE Symposium on Security and Privacy*, Oakland, 7-9 April 1986, 134-134. https://doi.org/10.1109/SP.1986.10022

[2] Huang, Y., Evans, D. and Katz, J. (2012) Private Set Intersection: Are Garbled Circuits Better than Custom Protocols?

[3] Pinkas, B., Schneider, T., Weinert, C., *et al*. (2018) Efficient Circuit-Based PSI via Cuckoo Hashing. 37*th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Tel Aviv, 29 April-3 May 2018, 125-157. https://doi.org/10.1007/978-3-319-78372-7_5

[4] Dong, C., Chen, L. and Wen, Z. (2013) When Private Set Intersection Meets Big Data: An Efficient and Scalable Protocol. *Proceedings of the* 2013 *ACM SIGSAC Conference on Computer & Communications Security*, Berlin, 4-8 November 2013, 789-800. https://doi.org/10.1145/2508859.2516701

[5] Kolesnikov, V. and Kumaresan, R. (2013) Improved OT Extension for Transferring Short Secrets. 33*rd Annual Cryptology Conference*, Santa Barbara, 18-22 August 2013, 54-70. https://doi.org/10.1007/978-3-642-40084-1_4

[6] Yang, K., Weng, C., Lan, X., *et al*. (2020) Ferret: Fast Extension for Correlated OT with Small Communication. *Proceedings of the* 2020 *ACM SIGSAC Conference on Computer and Communications Security*, 9-13 November 2020, 1607-1626. https://doi.org/10.1145/3372297.3417276

[7] Freedman, M.J., Nissim, K. and Pinkas, B. (2004) Efficient Private Matching and Set Intersection. *Advances in Cryptology-EUROCRYPT* 2004: *International Conference on the Theory and Applications of Cryptographic Techniques*, Interlaken, 2-6 May 2004, 1-19. https://doi.org/10.1007/978-3-540-24676-3_1

[8] Vos, J., Conti, M. and Erkin, Z. (2022) Fast Multi-Party Private Set Operations in the Star Topology from Secure ANDs and ORs. Cryptology ePrint Archive.

[9] Sang, Y. and Shen, H. (2007) Privacy Preserving Set Intersection Protocol Secure against Malicious Behaviors. 8*th International Conference on Parallel and Distributed Computing, Applications and Technologies* (*PDCAT* 2007), Adelaide, 3-6 December 2007. 461-468. https://doi.org/10.1109/PDCAT.2007.59

[10] Kamara, S., Mohassel, P., Raykova, M., *et al*. (2014) Scaling Private Set Intersection to Billion-Element Sets. *Financial Cryptography and Data Security*: 18*th International Conference*, *FC* 2014, Christ Church, 3-7 March 2014, 195-215. https://doi.org/10.1007/978-3-662-45472-5_13

[11] Baldi, P., Baronio, R., De Cristofaro, E., *et al*. (2011) Countering Gattaca: Efficient and Secure Testing of Fully-Sequenced Human Genomes. *Proceedings of the* 18*th ACM Conference on Computer and Communications Security*, Chicago, 17-21 October 2011, 691-702. https://doi.org/10.1145/2046707.2046785

[12] 魏立斐, 刘纪海, 张蕾. 面向隐私保护的集合交集计算综述[J]. 计算机研究与发展, 2022, 59(8): 1782-1799.

[13] Yang, G., Tan, C.H., Huang, Q., *et al*. (2010) Probabilistic Public Key Encryption with Equality Test. *Topics in Cryptology-CT-RSA* 2010: *The Cryptographers' Track at the RSA Conference* 2010, San Francisco, 1-5 March 2010, 119-131. https://doi.org/10.1007/978-3-642-11925-5_9

[14] Tang, Q. (2011) Towards Public Key Encryption Scheme Supporting Equality Test with Fine-Grained Authorization. *Information Security and Privacy*: 16*th Australasian Conference*, *ACISP* 2011, Melbourne, 11-13 July 2011, 389-406. https://doi.org/10.1007/978-3-642-22497-3_25

[15] Ma, S., Zhong, Y. and Huang, Q. (2022) Efficient Public Key Encryption with Outsourced Equality Test for Cloud-Based IoT Environments. *IEEE Transactions on Information Forensics and Security*, **17**, 3758-3772. https://doi.org/10.1109/TIFS.2022.3212203

[16] Xu, Y., Wang, M., Zhong, H., *et al*. (2017) Verifiable Public Key Encryption Scheme with Equality Test in 5G Networks. *IEEE Access*, **5**, 12702-12713. https://doi.org/10.1109/ACCESS.2017.2716971

[17] Wang, Y., Huang, Q., Li, H., *et al*. (2021) Private Set Intersection with Authorization over Outsourced Encrypted Datasets. *IEEE Transactions on Information Forensics and Security*, **16**, 4050-4062. https://doi.org/10.1109/TIFS.2021.3101059

[18] Kissner, L. and Song, D. (2005) Privacy-Preserving Set Operations. 25*th Annual International Cryptology Conference*, Santa Barbara, 14-18 August 2005, 241-257. https://doi.org/10.1007/11535218_15

[19] Hazay, C. and Venkitasubramaniam, M. (2017) Scalable Multi-Party Private Set-Intersection. 20*th IACR International Conference on Practice and Theory in Public-Key Cryptography*, Amsterdam, 28-31 March 2017, 175-203. https://doi.org/10.1007/978-3-662-54365-8_8

[20] Bay, A., Erkin, Z., Alishahi, M., *et al*. (2021) Multi-Party Private Set Intersection Protocols for Practical Applications. *Proceedings of the* 18*th International Conference on Security and Cryptography SECRYPT*, **1**, 515-522. https://doi.org/10.5220/0010547605150522