隐私计算联盟
Privacy Preserving Computing Alliance

# 联　　盟　　规　　范

PPCA 5 —2022

# 隐私计算 跨平台互联互通 开放协议 第 1 部分：ECDH-PSI

Privacy-preserving computation cross-platform interconnection—

Open protocol Part1: ECDH-PSI

2022-12-28 发布　　2022-12-28 实施

隐私计算联盟 发 布

# 目　次

目 次 ..... I
前　言 ..... II
引　言 ..... III
1 范围 ..... 5
2 规范性引用文件 ..... 5
3 术语和定义 ..... 5
4 缩略语 ..... 6
5 概述 ..... 6
5.1 ECDH-PSI 算法概述 ..... 6
5.2 算法流程 ..... 7
6 算法协议 ..... 8
6.1 算法协商握手 ..... 8
6.2 算法主体运行 ..... 10
7 传输层实现参考 ..... 11
7.1 通信框架 ..... 11
7.2 Protobuf 消息 ..... 11
7.3 通信模式 ..... 13
参　考　文　献 ..... 14

# 前　言

本文件是《隐私计算 跨平台互联互通》系列文件之一，该系列文件名称如下：

——开放协议 第1部分：ECDH-PSI。

# 引　　言

当前多方安全计算、联邦学习等隐私计算技术快速发展，越来越多的产品从试点部署阶段转入落地应用，市场竞争火热。但是，不同技术厂商提供的产品和解决方案在设计原理和功能实现之间存在较大差异，使得部署于不同平台的隐私计算参与方之间无法跨平台完成同一计算任务，为实现与部署于不同平台的多个合作方之间的数据融合，用户往往不得不部署多套产品以逐一适配。作为促进跨机构间数据共享融合的关键技术，隐私计算有望成为支撑数据流通产业的基础设施，但高额的应用成本不利于隐私计算技术的推广应用。因此，解决不同产品之间的技术壁垒，实现计算任务在跨平台间的互联互通已成为产业内的迫切需求。

# 版权声明



邮箱：ppca@caictyds.cn

# 隐私计算 跨平台互联互通
# 开放协议 第1部分：ECDH-PSI

## 1 范围

本文件规定了异构隐私计算平台间的ECDH-PSI互联互通的算法协议和传输层参考实现。

## 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

## 3 术语和定义

GB/T 25069-2022界定的以及下列术语和定义适用于本文件。

3.1

**隐私计算 privacy-preserving computation**

在保证数据提供方不泄露原始数据的前提下，对数据进行分析计算的一系列信息技术，保障数据在流通与融合过程中的“可用不可见”。

3.2

**开放协议 open protocol**

通过规范算法执行流程中交互信息，各平台独立开发算法来实现平台间算法的互通。

3.3

**节点 node**

各隐私计算技术平台的部署实例，是互联互通网络的基本组成单元，对外提供交互接口。

3.4

**算法 algorithm**

为解决问题严格定义的有限的有序规则集。

[来源：GB/T 25069—2022, 3.581]

3.5

**组件 component**

独立执行隐私计算任务的模块单元，其经过封装、符合开放接口规范、可以完成某个特定计算或算法，可独立部署。

3.6

**任务** task
组件运行实例的载体。

## 4 缩略语

下列缩略语适用于本文件。

| | | |
|---|---|---|
| ECDH | 椭圆曲线迪菲-赫尔曼 | Elliptic Curve Diffie–Hellman |
| PSI | 隐私集合求交 | Private Set Intersection |
| RPC | 远程过程调度 | Remote Procedure Call |

## 5 概述

### 5.1 ECDH-PSI 算法概述

ECDH-PSI的算法流程如图1所示，包括五个步骤：

第一步：参与方在本地计算原始数据（如$a_i$）的杂凑值，并将杂凑值映射到椭圆曲线上的点，然后加密[注1]得到数据（如$P1_i$）；

第二步：参与方将加密后的数据的传输给其它数据提供方，如参与方A将$P1_i$传输给数据提供方B；

第三步：参与方对在本地使用自己的私钥对步骤二中接收到的数据进行二次加密[注1]；

第四步：如果结果对另一个参与方可见，将步骤三中加密后的数据传输给另外一个参与方；

第五步：参与方本地计算集合求交的结果。

**注1：加密指基于椭圆曲线的点乘算法和本地的密钥（如$key_A$），对数据完成加密。**

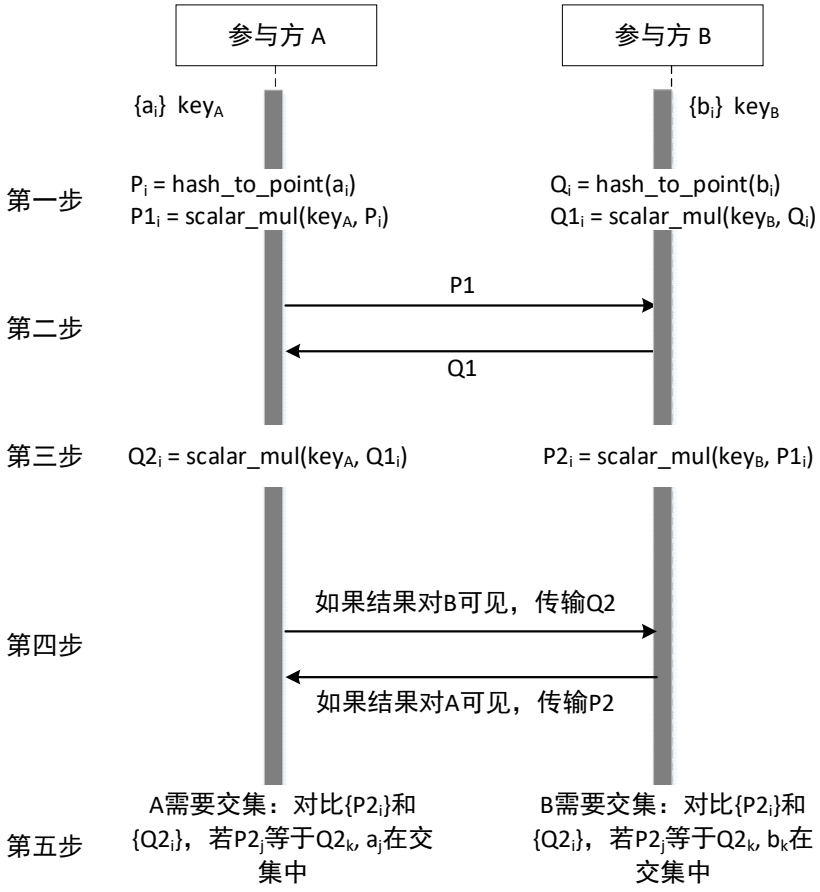


图1 ECDH-PSI算法

## 5.2 算法流程

算法流程包含两个阶段，如图2。第一阶段为算法协商握手阶段，第二阶段为算法主体运行阶段：

a) 算法协商握手阶段，确定算法版本、PSI算法类型、PSI算法参数、待求交集合的大小等算法运行所需的信息；

b) 算法主体运行阶段，实现隐私集合求交。

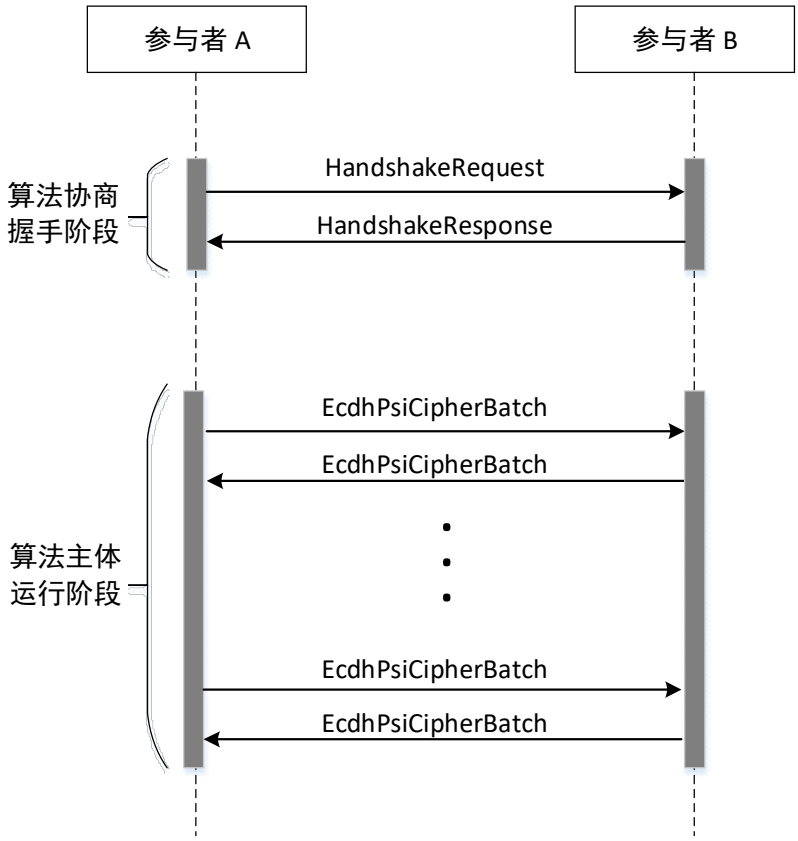


图2 ECDH-PSI协议流程

# 6 算法协议

## 6.1 算法协商握手

### 6.1.1 HandshakeRequest 消息

HandshakeRequest包括算法协商握手请求的基本信息，其数据结构如表1所示。

表 1 HandshakeRequest 数据结构

| 属性名称 | 数据类型 | 数据说明 | 示例 | 数据备注 |
|---|---|---|---|---|
| version | int32 | 握手请求版本号 | 1 | 必选 |
| supported_algos | string list | 支持的 PSI 算法，如 ecdh-psi | [“ECDH-PSI”] | 必选 |
| algo_params | google.protobuf.Any | 相应的 PSI 算法详细握手参数，与 supported_algo 对应，其数据结构如表 2 所示。<br>实际类型随 PSI 类型而变化，ECDH-PSI 的类型是 EcdhPsiParamsProposal | 见表 2 | 必选 |
| item_num | int64 | 待求交的 PSI 数据总量 | 10000 | 必选 |
| result_to_rank | int32 | PSI 结果的获取方，其值域表如 3 所示。 | -1 | 必选 |

其中：

- EcdhPsiParamsProposal包括ECDH-PSI算法参数协商的基本信息，如表2：

表 2 EcdhPsiParamsProposal 数据结构

| 属性名称 | 数据类型 | 数据说明 | 示例 | 数据备注 |
|---|---|---|---|---|
| supported_versions | int32 list | 支持的算法版本列表 | [1] | 必选 |
| curves | string list | 支持的椭圆曲线类型列表，如 curve25519、sm2 | [“SM2”] | 必选 |
| hash_methods | string list | 支持的 HASH 算法列表，如 sha256、sm3 | [“SM3”] | 必选 |

- result_to_rank字段用来确定 PSI 结果获取方，其值域如表3所示。

表 3 result_to_rank 的值域表

| 数值 | 数据说明 |
|---|---|
| -1 | 所有机构都可以拿到交集结果 |
| rank | 指定机构 rank 拿到交集结果。rank 是参与方在传输层的编号 |

### 6.1.2 HandshakeResponse 消息

HandshakeResponse消息结构包括算法协商握手响应的基本信息，其数据结构如表4所示。

表 4 HandshakeResponse 数据结构

| 属性名称 | 数据类型 | 数据说明 | 示例 | 数据备注 |
|---|---|---|---|---|
| header | ResponseHeader | 握手请求响应消息头。ResponseHeader 的格式见表 5 | 见表 5 | 必选 |
| algo | string | 决策下来的 PSI 算法 | “ECDH-PSI” | 必选 |
| item_count | int64 | HandshakeResponse 发送方待求交集合的大小 | 20000 | 必选 |
| algo_params | google.protobuf.Any | 决策下来的 PSI 算法详细握手参数。实际类型随 PSI 类型而变化，ECDH-PSI 的类型是 EcdhPsiParamsResult，见表 5 | 见表 5 | 必选 |

其中：

- EcdhPsiParamsResult包括ECDH-PSI算法参数协商的基本信息，如表5：

表 5 EcdhPsiParamsResult 数据结构

| 属性名称 | 数据类型 | 数据说明 | 示例 | 数据备注 |
|---|---|---|---|---|
| version | int32 | 支持的算法版本 | 1 | 必选 |
| curve | string | 支持的椭圆曲线类型，如 curve25519、sm2 | “SM2” | 必选 |
| hash_method | string | 支持的 HASH 算法，如 sha256、sm3 | “SM3” | 必选 |

- ResponseHeader消息结构包括的基本信息如表6所示

表 6 ResponseHeader 数据结构

| 属性名称 | 数据类型 | 数据说明 | 数据备注 |
|---|---|---|---|
| error_code | int | 握手响应的结果，其值域如表 7 所示。 | 必选 |
| error_msg | string | 用户自定的消息字符串 | 可选 |

- error_code的取值和说明如表7所示。

表 7 error_code 的值域表

| 错误码 | 数据说明 |
|---|---|
| 0 | 成功 |
| 31100000 | GENERIC_ERROR，通用错误 |
| 31100001 | UNEXPECTED_ERROR，状态不符合预期错误 |
| 31100002 | NETWORK_ERROR，网络通信错误 |
| 31100100 | INVALID_REQUEST，非法请求 |
| 31100101 | INVALID_RESOURCE，运行资源不满足 |
| 31100200 | HANDSHAKE_REFUSED，握手拒绝 |
| 31100201 | UNSUPPORTED_VERSION，不支持的版本 |
| 31100202 | UNSUPPORTED_ALGO，不支持的算法 |
| 31100203 | UNSUPPORTED_PARAMS，不支持的算法参数 |

## 6.2 算法主体运行

### 6.2.1 EcdhPsiCipherBatch 消息

EcdhPsiCipherBatch的基本信息如表8所示。

表 8 EcdhPsiCipherBatch 数据结构

| 属性名称 | 数据类型 | 数据说明 | | 数据备注 |
|---|---|---|---|---|
| type | string | 标识密文的类型，取值"enc"和“dual.enc”。 "enc":图 1 中算法 | “enc” | 必选 |

| 属性名称 | 数据类型 | 数据说明 | | 数据备注 |
|---|---|---|---|---|
| | | 第二步中交换的密文信息。"dual.enc ":图 1 中算法第四步中交换的密文信息 | | |
| batch_index | int32 | 传输批次的编号。当待求交集合比较大的时候，发送方可以选择分多个批次发送密文 | 0 | 必选 |
| is_last_batch | bool | 是否为最后一个传输批次 | false | 必选 |
| count | int32 | 当前批次包含的密文数量 | 1000 | 必选 |
| ciphertext | bytes | 当前批次包含的密文 | - | 必选 |

## 7 传输层实现参考

### 7.1 通信框架

异构隐私计算技术平台间进行互联互通时，通信框架应满足兼容性、通用性以及安全性的要求，如表9所示。

表 9 通信框架范围

| | |
|---|---|
| **通信协议** | gRPC |
| **编码方式** | ProtoBuf |

#### 7.1.1 初始化通信协议

初始化通信协议在 PSI 任务开始前执行一次。

每个参与方向其它参与者通知自己的存在性，即向他人发送 connect_{self_rank}：

```
For i in 0..word_size [注1]:
  if i == self_rank:
    continue
  P2P send to rank i: {key: connect_{self_rank}, value: ""}
```

每个参与者检查他人的存在性，即依次检查 connect_{rank} 消息已经收到：

```
For i in 0..word_size:
  if i == self_rank:
    continue
  P2P receive on key connect_{i}
```

注 1：word_size 表示参与者数量

### 7.2 Protobuf 消息

不同隐私计算的参与者之间使用 Protobuf 协议传递信息。

```
service ReceiverService
{
```

```
    rpc Push(PushRequest) returns (PushResponse);
}
```

### 7.2.1 PushRequest 消息

PushRequest包括传输的基本信息，其数据结构如表10所示。

表 10 PushRequest 数据结构

| 属性名称 | 数据类型 | 数据说明 | 示例 | 数据备注 |
|---|---|---|---|---|
| sender_rank | uint64 | 发送者的编号，如 0 指 rank 0 | 0 | 必选 |
| key | string | 消息唯一 ID，生成规则见 7.3 节 | “root:P2P-0:0->1” | 必选 |
| value | bytes | 消息体，ECDH-PSI 中的 protobuf 序列化成二进制 string，然后把整个 string 放到 value 中 | 需要传输的实际信息 | 必选 |
| trans_type | TransType | 传输模式，如全量传输、分块传输，其值域如表 11 所示 | 见表 11 | 必选 |
| chunk_info | ChunkInfo | 消息大小，其数据结构如表 12 所示 | 见表 12 | 必选 |

其中：

- TransType 的值域如表 11 所示：

表 11 TransType 的值域表

| 数值 | 数据说明 |
|---|---|
| MONO | 全量传输模式 |
| CHUNKED | 分块传输模式 |

- ChunkInfo 的值域如表 12 所示。

表 12 ChunkInfo 的数据结构

| 属性名称 | 数据类型 | 数据说明 | | 数据备注 |
|---|---|---|---|---|
| message_length | uint64 | 数据总大小，单位是字节 Byte | 1048576 | 必选 |
| chunk_offset | uint64 | 当前分块的偏移量 | 0 | 必选 |

### 7.2.2 PushResponse 消息

PushResponse包括传输的基本信息，其数据结构如表13所示。

表 13 PushResponse 消息的数据结构

| 属性名称 | 数据类型 | 数据说明 | 数据备注 |
|---|---|---|---|
| header | ResponseHeader | 返回消息，如表 6 所示 | 必选 |

## 7.3 通信模式

### 7.3.1 信道

信道是一个逻辑概念，用于区分通信的上下文。每一个信道有一个全局唯一名称，命名规则为：\w+，即信号名称由字母、数字、下划线组成。信道的名字由通信组双方约定，在初始化阶段由用户传入。

信道唯一的作用是影响 message key 的生成，信道名称会作为 message key 一部分，因此，不同信道中的消息一定不会有相同的 key，从而不同信道的消息在逻辑上不会混淆。

当上层算法需要多个信道时，第一个信道称为主信道，其它信道称为子信道。子信道的命名规则为：主信道名称-子信道编号。

**举例：假设主信道名称为 root，则 0 号子信道名称为 root-0，1 号子信道名称为 root-1，以此类推。**

### 7.3.2 P2P 通信

P2P 通信允许在任意两个参与者之间发送信息。P2P 通信 key 的命名规则为：**{信道名称}:P2P-{计数器}:{发送者 RANK}->{接收者 RANK}**，其中每一个信道、每一对 <sender，receiver> 都有一个独立的计数器。

**举例，假设信道名称为 root，以下消息依次发送：**

Rank 0 → 1 发送消息，key 为：root:P2P-0:0->1

Rank 1 → 0 发送消息，key 为：root:P2P-0:1->0

Rank 0 → 2 发送消息，key 为：root:P2P-0:0->2

Rank 0 → 1 发送消息，key 为：root:P2P-1:0->1

# 参 考 文 献

[1] NIST SP 800-57 Part1 Rev. 5 Recommendation for Key Management: Part 1 – General

[2] GB/T 25069-2022 信息安全技术 术语

[3] GM/T 32905 - 2016 信息安全技术 SM3 密码杂凑算法

[4] GB/T 32918-2017 信息安全技术 SM2 椭圆曲线公钥密码算法