MPC及同态加密

多方安全计算(MPC)是在无可信第三方情况下，多个参与方协同完成计算目标，并保证每个参与者除计算结果外均不能得到其他参与实体的任何输入信息。多方安全计算技术可以获取数据使用价值，却不泄露原始数据内容，保护隐私，可适用于多方联合数据分析、数据安全查询（PIR，Private Information Retrieval）、隐私求交（PSI，rivate Set Intersection）、数据可信交换等应用场景。

以计算平均工资为例，一种简单的的方式是有一个可信第三方F负责收集A，B，C每个人的工资收入数据，并计算平均数。然而，显示生活中的可信第三方不总是存在，也并不一定安全，其有可能会泄漏其收集的隐私数据。另一种方式就是采用多方安全计算，在无可信第三方的情况下，来保障各方的工资数据不泄漏，且能计算出平均工资数。

|  |  |
| --- | --- |
| Alice总工资 | 200$ |
| Bob总工资 | 350$ |
| Camille总工资 | 500$ |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 留给Alice的资金 | 留给Bob的资金 | 留给Camille的资金 |
| Alice分出来的工资 | 70 | 30 | 100 |
| Bob分出来的工资 | 150 | 100 | 100 |
| Camille分出来的工资 | 120 | 150 | 230 |
| sum | 340 | 280 | 430 |

然后我们可以统计三个人将工资重新分配后的总值分别是（340，280，430，）然后这三组数相加就是三人工资总和，即可得出三人平均工资是350.因为在这期间每个人如何分配工资只有自己知道，工资的总值又是确定的，这样就能在不暴露每个人的实际工资的情况下计算平均薪资。

同态加密

同态加密（Homomorphic Encryption）是指将原始数据经过同态加密后，对得到的密文进行特定的运算，然后将计算结果再进行同态解密后得到的明文等价于原始明文数据直接进行相同计算所得到的数据结果。

同态加密与一般加密方案的关注点不同，一般的加密方案关注的是数据存储安全，即我要给其他人发送信息或者存储信息，我需要对数据进行加密之后再发送和存储，这里我们只需要保证在数据传送和存储的过程中不被其他人窃听到即可，在这个过程中用户时不能对加密的结果做任何操作的，否则可能会导致解密失败

加密过程

（本地）生成一对公私钥，公钥pub和密钥priv，公钥用于加密，密钥用于解密；

（本地）使用公钥pub分别加密m1和m2，得到Epub(m1)和Epub(m2)；

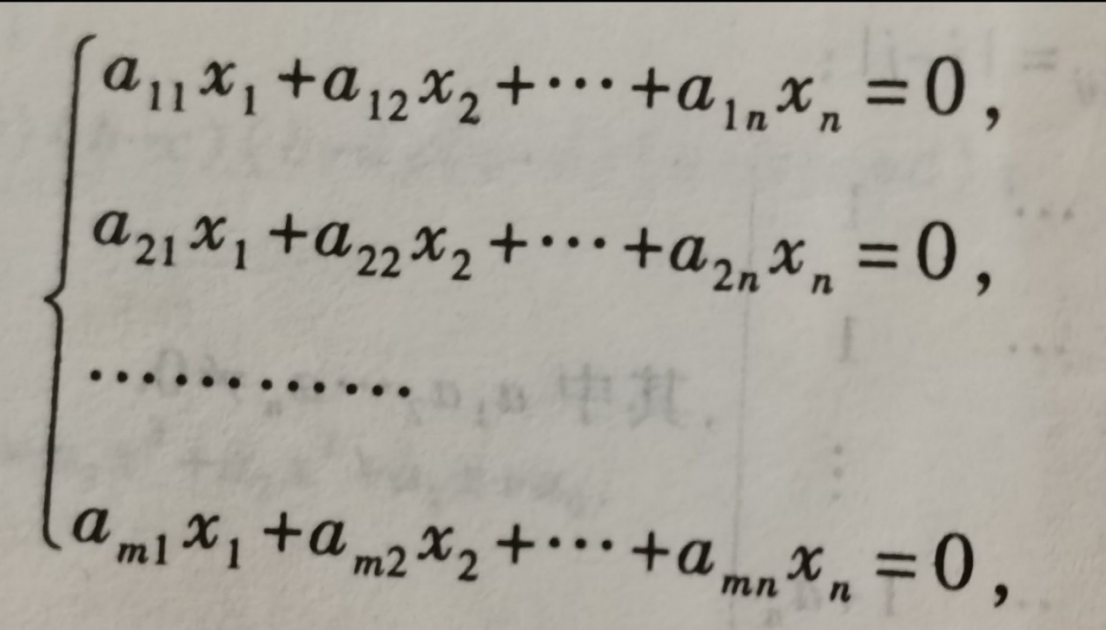
（第三方）使用Addpub函数处理Epub(m1)和Epub(m2)，即Addpub(Epub(m1),Epub(m2))；

（本地）使用密钥priv解密Addpub(Epub(m1),Epub(m2))，即Dpriv(Addpub(Epub(m1),Epub(m2)))；

Dpriv(Addpub(Epub(m1),Epub(m2)))就等于m1+m2。第三方通过上述步骤3实现了m1和m2在加密状态下做加法的操作。

通态加密的关注点则是数据处理安全，同态加密提供了一种对加密数据进行处理的功能。也就是说其他人可以对加密后的数据进行处理，在这个过程中不会泄露任何原始的内容，在数据处理完成之后再进行解密，得到的正是对原始数据进行相同处理后的结果。

对此我觉得同态加密有点像解决线性方程组



其中的X就是我们要加密的信息，aij ，就是加密算法，其他人对加密后的数据进行处理就相当于对方程组采用高斯消元法来对方程组进行变化。经过任意次加密，最后的解密密文是不变的。