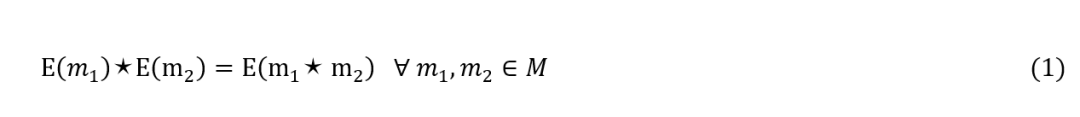
* **什么是同态加密**

同态加密（HE，homomorphic encryption）是密码学里一种特殊的加密模式，同态加密使我们可以将加密后的密文发给任意的第三方进行计算，并且在计算前不需要解密，即：在密文上进行计算。虽然同态加密的概念最早出现于 30 年前，但是第一个支持在密文上进行任意运算的 全同态加密 框架出现较晚，在 2009 年由 Craig Gentry 提出。

同态加密的数学定义为：



其中，E 为加密算法，M 是所有可能信息的集合。如果加密算法 E 满足公式 (1)，那么我们称 E 在★运算上符合同态加密的性质。目前的同态加密算法，主要支持两种运算上的同态：加法和乘法。

* **同态加密的组成**

同态加密算法一般包含以下四个部分：

* KeyGen：密钥生成算法，产生公钥和私钥
* Encryption：加密算法
* Decryption：解密算法
* Homomorphic Property：同态加密计算部分

其中，前三个部分在很多加密算法中都可以看到，第四部分则是同态加密算法的核心，指导密文下的运算。

* **同态加密的分类**

为了更好地理解与运用同态加密算法，我们按照将同态加密算法支持的运算类型和数量，将其分成 3 类：部分同态加密、层次同态加密、和全同态加密。

* + **部分同态加密**（Partial HE，简称 PHE）指同态加密算法只对加法或乘法（其中一种）有同态的性质。 例如：RSA 加密是最早应用的公钥加密算法框架，同时 RSA 算法也是一种 PHE 算法，其对乘法有同态的性质。PHE 的研究成果出现比较早，并且加法同态加密算法（Additive HE）比 乘法同态加密算法要多一些。PHE 的优点是原理简单、易实现，缺点是仅支持一种运算（加法或乘法）。
  + **层次同态加密算法**（LHE，Leveled HE 或 SWHE ，SomeWhat HE）一般支持有限次数的加法和乘法运算。 层次同态加密的研究主要分为两个阶段，第一个阶段是在 2009 年 Gentry 提出第一个 FHE 框架以前，比较著名的例子有：BGN 算法、姚氏混淆电路等；第二个阶段在 Gentry FHE 框架之后，主要针对 FHE 效率低的问题。LHE 的优点是同时支持加法和乘法，并且因为出现时间比 PHE 晚，所以技术更加成熟、一般效率比 FHE 要高很多、和 PHE 效率接近或高于 PHE，缺点是支持的计算次数有限。
  + **全同态加密算法**（Fully HE，简称 FHE）支持在密文上进行无限次数的、任意类型的计算。从使用的技术上分，FHE 有以下类别：基于理想格的 FHE 方案、基于 LWE/RLWE 的 FHE 方案等等。FHE 的优点是支持的算子多并且运算次数没有限制，缺点是效率很低，目前还无法支撑大规模的计算。

日程表

描述已自动生成

图 1 三种类型同态加密的研究时间线

图 1 展示了三类同态加密算法的研究时间线，同态加密的概念在 1976 年提出，随后 PHE 的研究成果逐渐丰富；在 Gentry 的 FHE 框架前，LHE 研究占主导；2009 年后，研究热点集中在 FHE。

我们在云计算应用场景下面进行介绍：

表格

低可信度描述已自动生成

Alice通过Cloud，以Homomorphic Encryption（以下简称HE）处理数据的整个处理过程大致是这样的：

1. Alice对数据进行加密。并把加密后的数据发送给Cloud；
2. Alice向Cloud提交数据的处理方法，这里用函数f来表示；
3. Cloud在函数f下对数据进行处理，并且将处理后的结果发送给Alice；
4. Alice对数据进行解密，得到结果。

据此，我们可以很直观的得到一个HE方案应该拥有的函数：

**KeyGen函数**：密钥生成函数。这个函数应该由Alice运行，用于产生加密数据Data所用的密钥Key。当然了，应该还有一些公开常数PP（Public Parameter）；

**Encrypt函数**：加密函数。这个函数也应该由Alice运行，用Key对用户数据Data进行加密，得到密文CT（Ciphertext）；

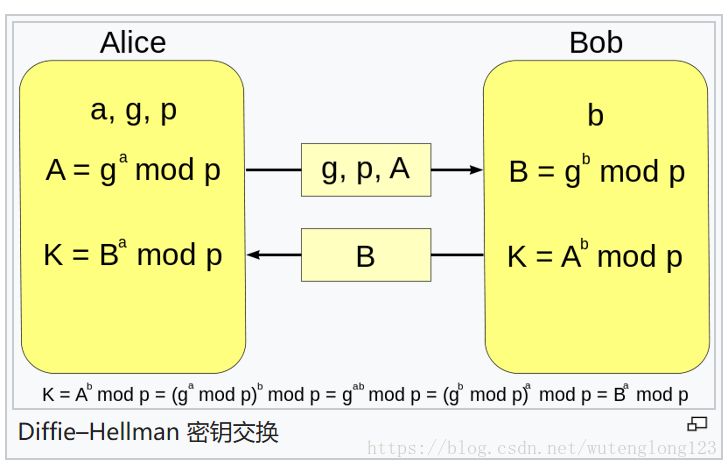
**Evaluate函数**：评估函数。这个函数由Cloud运行，在用户给定的数据处理方法f下，对密文进行操作，使得结果相当于用户用密钥Key对f(Data)进行加密。

**Decrypt函数**：解密函数。这个函数由Alice运行，用于得到Cloud处理的结果f(Data)。

* **Diffie-Hellman算法**

Diffie-Hellman算法是第一个公开[密钥](https://so.csdn.net/so/search?q=%E5%AF%86%E9%92%A5&spm=1001.2101.3001.7020" \t "_blank)算法，早在 1976 年就发现了。其安全性源于在有限域上计算离散对数，比计算指数更为困难。该算法可以使两个用户之间安全地交换一个密钥，但不能用于加密或解密信息。

Diffie-Hellman密钥交换方案（DHKE）提供了实际中密钥分配问题的解决方案，即它允许双方通过不安全的信道进行交流，得到一个共同密钥。许多公开和商业的密码协议中都实现了这种基本的密钥协议技术，例如**SSH、TLS、IPSec**。



假设Alice和Bob希望在不安全的网络环境中协商一个对称密钥进行通信，那么可按如下步骤进行：

文本

描述已自动生成

DH算法用于在不安全的公共通道中协商密钥，安全性体现在：在有限域上计算离散代数非常困难。DH不仅支持两点密钥交互，也可以支持多点密钥交互。

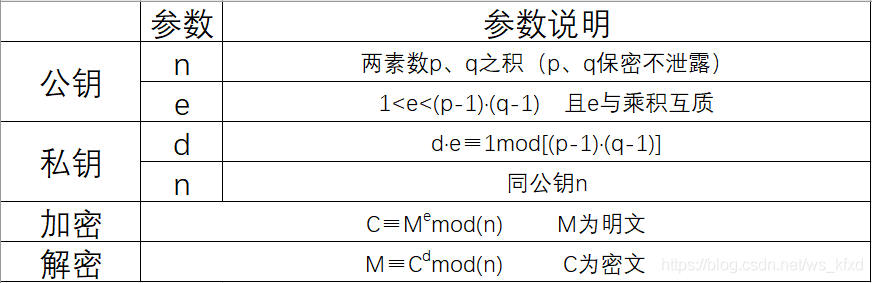
但是DH算法本身存在安全性缺陷，即没有对交互双方进行身份验证，容易遭受中间人攻击。

* **RSA: RSA的安全性依赖于大数的因子分解**

RSA是第一个比较完善的公开密钥算法，它既能用于加密，也能用于数字签名。RSA以它的三个发明者Ron Rivest, Adi Shamir, Leonard Adleman的名字首字母命名，这个算法经受住了多年深入的密码分析，虽然密码分析者既不能证明也不能否定RSA的安全性，但这恰恰说明该算法有一定的可信性，目前它已经成为最流行的公开密钥算法，具有乘同态性质。

　　RSA的安全基于大数分解的难度。其公钥和私钥是一对大素数（100到200位十进制数或更大）的函数。从一个公钥和密文恢复出明文的难度，等价于分解两个大素数之积（这是公认的数学难题）。

　　RSA的公钥、私钥的组成，以及加密、解密的公式可见下表：

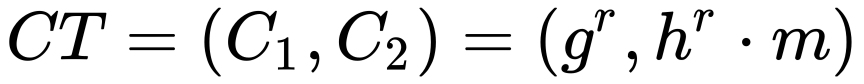


算法描述：

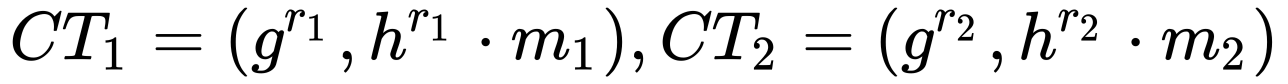
1. 选择一对不同的、足够大的素数p，q。
2. 计算n=pq。
3. 计算f(n)=(p-1)(q-1)，同时对p, q严加保密，不让任何人知道。
4. 找一个与f(n)互质的数e，且1<e<f(n)。
5. 计算d，使得d·e≡1 mod f(n)。
6. 公钥KU=(n，e)，私钥KR=(n，d)。
7. 加密时，先将明文变换成0至n-1的一个整数M。若明文较长，可先分割成适当的组，然后再进行交换。设密文为C，则加密过程为：C≡M^e mod(n)。
8. 解密过程为：M≡C^d mod (n) 。

这里要解释一下，≡是数论中表示同余的符号。公式中，≡符号的左边必须和符号右边同余，也就是两边模运算结果相同。显而易见，不管f(n)取什么值，符号右边1 mod f(n)的结果都等于1；符号的左边d与e的乘积做模运算后的结果也必须等于1。这就需要计算出d的值，让这个同余等式能够成立。

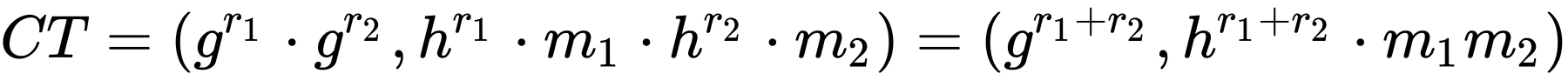
* **Elgamal加密方案的密文形式为：**



其中，r是加密过程中选的一个随机数，g是一个生成元，h是公钥。如果我们有两个密文：



我们把这两个密文的第一部分相乘，第二部分相乘，会得到：



也就是说，相乘以后的密文正好是m1m2所对应的密文。这样，用户解密后得到的就是m1m2的结果了。而且注意，整个运算过程只涉及到密文和公钥，运算过程不需要知道m1m2的确切值。所以我们说Elgamal具有乘同态性质。但是很遗憾，其没有加同态性质。

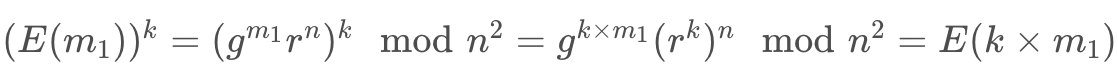
* **Paillier同态加密算法**

我们先来简短认识一下Paillier同态加密算法，具有加同态性质：

手机屏幕截图

描述已自动生成

扩展至批量乘法（常数与明文相乘等同于密文的常数次幂）：



手机屏幕截图

描述已自动生成

