> 本文主要记录隐私计算中的同态加密（Homomorphic Encryption，HE）技术，包括部分同态加密（RSA、GM、ElGamal、Paillier）、近似同态加密（BGN）、有限级数全同态加密和全同态加密（DGHV、BGV、BFV、CKKS、GSW、FHEW、TFHE）等技术，**仅供参考**。

由于**篇幅限制**，将同态加密技术的介绍分为四个部分，**第一部分**讲述部分同态加密和近似同态加密技术；**第二部分**讲述 Bootstrapping 和 DGHV、BGV 全同态加密方案；**第三部分**讲述 SIMD 打包技术与 BFV、CKKS 全同态加密方案；**第四部分**讲述 GSW、FHEW、TFHE 全同态加密方案。

在本文中为**简化表示**，将加密记为 $Enc(\cdot)$，解密记为 $Dec(\cdot)$，不标明使用的公钥和私钥代表只有一对公私钥。本文中介绍的加密方案基本都依赖于各种**计算难题**，在之前的文章中有过介绍。

一、同态加密介绍

1. 简单介绍

加密技术一般用来保护重要的隐私信息，只能实现安全的**存储**信息。而同态加密是一种特殊的加密技术，额外提供了对信息进行**计算**的功能。

“同态”可以理解为：明文运算后的密文可以由明文对应的密文直接运算得到，即**不需要解密**就可以实现明文对应的运算。一般可分为**加法同态**和**乘法同态**（也会存在其他的同态性质，这里以加法和乘法为例）。

- **加法同态**：对于两个消息 m_0, m_1，设其在某种加法同态加密算法下的密文分别为 c_0, c_1。存在一种函数运算，如 $f(x, y) = x + y$，使得：$f(c_0, c_1) = c_0 + c_1 = Enc(m_0 + m_1)$。
- **乘法同态**：对于两个消息 m_0, m_1，设其在某种加法同态加密算法下的密文分别为 c_0, c_1。存在一种函数运算，$f(x, y) = x \times y$，使得：$f(c_0, c_1) = c_0 \times c_1 = Enc(m_0 \times m_1)$。

传统加密方式要通过 c_0, c_1 得到 $m_0 + m_1$ 的结果，只能先解密再运算：$m_0 + m_1 = Dec(m_0) + Dec(m_1)$，这样解密会**泄露明文** m_0, m_1 的信息。

通过同态加密可以实现 $m_0 + m_1 = Dec(f(c_0, c_1))$，可以在不泄露明文 m_0, m_1 的信息的情况下得到最终结果，这是使用同态加密实现隐私计算的基本原理。

2. 同态加密分类

同态加密一般可以分为四类：**部分同态加密**（Partially Homomorphic Encryption）、**近似同态加密**（Somewhat Homomorphic Encryption）、**有限级数全同态加密**（Leveled Fully Homomorphic Encryption）、**全同态加密**（Fully Homomorphic Encryption）。

- **部分同态加密**：可以简单的理解为只支持加法同态或乘法同态**其中一种**（可计算的函数只能包含特定的运算操作）。
- **近似同态加密**：一般指支持**无限次**加法同态和**少量**乘法同态 （可计算的函数包含的运算操作不能随意组合）。
- **有限级数全同态加密**：支持加法同态和乘法同态，但同态**运算次数有限**（可计算的函数包含的运算可以随意组合，但总体复杂度有限制）。
- **全同态加密**：支持**无限次**加法同态和乘法同态运算（能实现任意功能的函数操作）。

二、部分同态加密

本文主要介绍四种部分同态加密技术：RSA加密、GM加密、ElGamal加密、Paillier加密。

1. RSA加密方案

RSA 加密方案由 Rivest、Shamir 和 Adleman 在 1978 发表[1]（似乎在1973年 Clifford Cocks 就给出了类似的加密方案，但因为工作保密没有发表）。

1.1 密钥生成算法

1. 选取两个不同的大素数 p, q，计算 $n = pq$，$\varphi(n) = (p-q)(q-1)$，其中 $\varphi(\cdot)$ 为欧拉函数。
2. 选取两个正整数 e, d 使得 $ed \equiv 1 \ mod \ \varphi(n)$，即 $ed = k \cdot \varphi(n) + 1$。可通过随机取与 $\varphi(n)$ 互素的正整数 e，并计算逆元得到 $d \equiv e^{-1} \ mod \ \varphi(n)$。
3. 公钥为 e, n，私钥为 d。

1.2 加密算法

对于消息 m （m 为小于 n 的正整数），使用公钥加密得到密文 c 的过程如下：

$$c = Enc(m) = m^e \ mod \ n$$

1.3 解密算法

对于密文 c，使用私钥解密得到明文消息 m 的过程如下：

$$m = Dec(c) = c^d \ mod \ n$$

正确性

可以验证：$c^d = m^{ed} = m^{k \cdot \varphi(n) + 1} = (m^{\varphi(n)})^k \cdot m = m \ (mod \ n)$。

其中，根据欧拉定理，$m^{\varphi(n)} \equiv 1 \ (mod \ n)$。

安全性

RSA 加密方案的安全性依赖于**大数分解问题**。在密钥生成时，私钥 d 的安全性由 $\varphi(n)$ 决定，而 $\varphi(n)$ 的计算直接依赖于 p, q。因此只要无法根据 n 计算出 $p, q, \varphi(n)$，那么私钥 d 就无法被破解。

上述 RSA 加密方案**不满足语义安全性**，因为对于同一个消息 m，使用上述加密算法每次加密得到的**密文是确定的**，在实际应用中可以通过随机填充或引入哈希函数来弥补这个缺陷。

乘法同态性

RSA 加密方案具有**乘法同态性**，在模 n 意义下，有：

$$Enc(m_0) \cdot Enc(m_1) = m_0^e \cdot m_1^e = (m_0 m_1)^e = Enc(m_0 m_1)$$

---

2. GM 加密方案

GM 加密方案由 Goldwasser 和 Micali 在 1982 年发表[2]，该方案只能对一个比特进行加密，具有**异或同态性**，实用性有限。

在了解 GM 方案之前需要先了解一些前置知识：**Legendre 符号** 和 **Jacobi 符号**，简单介绍如下：

> **Legendre 符号**：对于奇素数 p，对于小于 p 的正整数 a，其 Legendre 符号记为 $\left(\frac{a}{p}\right)$，在本文中简记为 $L_p(a)$。当 a 为模 p 的二次剩余时，$L_p(a)$ 为 1，否则 $L_p(a)$ 为 0。（本文中只考虑 $a < p$ 的情况）。

> **Jacobi 符号**：对于奇正整数 n，对于小于 n 且与 n 互素的正整数 a，其 Jacobi 符号在本文中简记为 $J_n(a)$。若 n 的质因数分解为 $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$，则 $J_n(a) = \prod_{i=1}^{k} (L_{p_i}(a))^{e_i}$。当 n 为奇素数时，$J_n(a) = L_n(a)$。

Jacobi 符号在模数 n 为奇素数时和 Legendre 符号一样可以用来判断 a 是否为二次剩余：当 $J_n(a)=1$ 时，a 一定是二次剩余。

当奇正整数 n 不为素数时，Jacobi 符号无法直接用来判断a 是否为二次剩余：当 $J_n(a)=1$ 时，a 可能不是二次剩余，将满足这种性质的 a 称为**伪二次剩余**。

Jacobi 符号有如下几个性质：

1. 当 n 为奇素数时，$J_n(a)=a^{\frac{n-1}{2}}\ mod\ n$。

   证明：乘法群 $Z_n^*=\{1,2,3,\cdots,n-1\}$，设其一个生成元为 g，则 $Z_n^*=\{g^0,g^1,\cdots,g^{n-2}\}$（$g^{n-1}=1=g^0$），对其中每个元素平方后可以得到二次剩余集合 $\{g^0,g^2,\cdots,g^{n-3}\}$，即得到的每个二次剩余都是 g 的偶数次方。

   对于任意一个二次剩余 g^{2k}，在模 n 意义下可以验证 $J_n(g^{2k})=(g^{2k})^{\frac{n-1}{2}}=g^{(n-1)k}=1$。

   对于任意一个二次非剩余 g^{2k+1}，在模 n 意义下可以验证 $J_n(g^{2k+1})=(g^{2k+1})^{\frac{n-1}{2}}=g^{(n-1)k}\cdot g^{\frac{n-1}{2}}=g^{\frac{n-1}{2}}$。由于 $g^{n-1}=1$，则 $g^{\frac{n-1}{2}}=\pm1$。考虑到生成元 g 的阶为 $n-1$，这说明 $g^{\frac{n-1}{2}}\neq1$，因此 $J_n(g^{2k+1})=g^{\frac{n-1}{2}}=-1$。

   综上所述，可以证明 $J_n(a)=a^{\frac{n-1}{2}}\ mod\ n$ 成立。

2. 当 n 为奇素数时，$J_n(ab)=J_n(a)J_n(b)$。

   证明：$J_n(ab)=(ab)^{\frac{n-1}{2}}=a^{\frac{n-1}{2}}\cdot b^{\frac{n-1}{2}}=J_n(a)J_n(b)$。

3. 当 $n=pq$，其中 p,q 为大素数时，$J_n(ab)=J_n(a)J_n(b)$。

   证明：$J_n(ab)=J_p(ab)\cdot J_q(ab)=J_p(a)J_p(b)\cdot J_q(a)J_q(b)=J_n(a)J_n(b)$。

2.1 密钥生成算法

1. 选取两个不同的大素数 p,q，计算 $n=pq$，可以得到模 n 意义下的乘法群 $\mathbb{Z}_n^*$（小于 n 且与 n 互素的正整数组成的集合）。
2. 选取正整数 $x\in\mathbb{Z}_n^*$ 使得 x 是模 n 意义下的一个伪二次剩余。
3. 公钥为 n,x，私钥为 p,q。

2.2 加密算法

对于消息 $m\in\{0,1\}$ ，随机选取 $r\in\mathbb{Z}_n^*$，使用公钥加密得到密文 c 的过程如下：

$$c=Enc(m)=x^m\cdot r^2\ mod\ n$$

2.3 解密算法

对于密文 c ，记 $c_p=c\ mod\ p$ 且 $c_q=c\ mod\ q$，使用私钥解密得到明文消息 m 的过程如下：

- 若 $J_p(c_p)=1$ 且 $J_q(c_q)=1$ 时，$m=0$；
- 其余情况，$m=1$。

正确性

记 $r_p=r\ mod\ p$ 且 $r_q=r\ mod\ q$，则

- 当 $m=0$ 时，$c=r^2$，此时 $c=r^2\equiv r_p^2\ (mod\ p)$，即 $c_p\equiv r_p^2\ (mod\ p)$，即 c_p 为模 p 二次剩余，则 $J_p(c_p)=1$。同理：$J_q(c_q)=1$。
- 当 $m=1$ 时，$c=x\cdot r^2\ mod\ n$，由于 x 不是模 n 的二次剩余，则 c 也**不是**模 n 的**二次剩余**。此时一定可以得到 c_p 不是模 p 的二次剩余，且 c_q 不是模 q 的二次剩余，即 $J_p(c_p)=J_q(c_q)=-1$。

证明：可以计算出 $J_n(c) = J_n(x) \cdot J_n(r^2) = 1$，从而得到 $J_p(c_p)J_q(c_q) = J_n(c) = 1$，因此可以解出：$J_p(c_p) = J_q(c_q) = 1$ 或 $J_p(c_p) = J_q(c_q) = -1$。

当 $J_p(c_p) = J_q(c_q) = 1$ 时，即 c 为模 p 和模 q 的二次剩余，此时有 $c \equiv t_p^2 \ (mod\ p)$ 和 $c \equiv t_q^2 \ (mod\ q)$ 成立，根据中国剩余定理可以构造出模 n 意义下唯一的 t 满足 $t \equiv t_p \ (mod\ p)$ 且 $t \equiv t_q \ (mod\ q)$。因此可以得到 $c \equiv t^2 \ (mod\ p)$ 和 $c \equiv t^2 \ (mod\ q)$ 成立，从而得到 $c \equiv t^2 \ (mod\ n)$ 成立，说明 c 是模 n 意义下的二次剩余，出现矛盾。因此 $J_p(c_p) = J_q(c_q) = -1$ 得证。

安全性

GM 加密算法的安全性依赖于在不知道 p, q 具体取值的情况下，只根据 n 判断正整数 c 是否为模 p 和 q 的二次剩余**是困难的**。

在密钥生成算法中，注意到 x **必须**是模 n 意义下的一个**伪二次剩余**，即要求 $J_n(x) = 1$ 且 x 不为模 n 的二次剩余。这是出于正确性和安全性的考虑，原因如下：

- 若 x 为模 n 的二次剩余，显然密文 $c = x^m \cdot r^2$ 无论 m 取何值，c 都为模 n 意义下的二次剩余，则也为模 p 和 q 意义下的二次剩余，此时 $J_p(c_p) = 1$ 且 $J_q(c_q) = 1$ 恒成立，就**失去了解密手段**。
- 若 x 不为模 n 的二次剩余，但不要求 $J_n(x) = 1$，由于 Jacobi 符号的计算存在多项式时间内的算法，此时根据 $c = x^m \cdot r^2$ 可以在多项式时间内计算出 $J_n(c) = J_n(x^m) \cdot J_n(r^2) = J_n(x^m)$ 的取值，显然当 $m = 0$ 时，$J_n(x) = 1$，否则 $J_n(x) \neq 1$。攻击方可以在多项式时间内根据 Jacobi 符号暴露出的信息获取明文 m 的取值，就**失去了安全性**。

该方案满足语义安全性，因为每次加密都会**随机选取** r，可以保证同一消息的多次重复加密结果不确定。

异或（模2加法）同态性

GM 加密方案具有**异或同态性**，在模 n 意义下，有：

$$Enc(m_0) \cdot Enc(m_1) = x_0^{m_0} \cdot r_0^2 \cdot x_1^{m_1} \cdot r_1^2 = x_0^{m_0} \cdot x_1^{m_1} \cdot (r_0 r_1)^2 = Enc(m_0 \oplus m_1)$$

根据 GM 加密方案的正确性，易证只有当 $m_0 = m_1$ 时，密文 $c = Enc(m_0 \oplus m_1)$ 才满足 $J_p(c_p) = 1$ 且 $J_q(c_q) = 1$。

---

3. ElGamal 加密方案

ElGamal 加密方案由 T. ElGamal 于 1985 年发表[3]，该方案基于 Diffie-Hellman 密钥交换的思想，其安全性本质上依赖于离散对数问题难解。ElGamal 加密方案一般可以在循环群或椭圆曲线上实现，两种实现方式思想一样，本文只介绍在循环群上的加密方案。

3.1 密钥生成算法

对于大素数 p，有模 p 乘法循环群 $\mathbb{Z}_p^*$，设 g 为该群的一个生成元，随机取群中元素 g^a。

公钥为 p, g, g^a，私钥为 a。

3.2 加密算法

对于消息 m，随机取小于 p 的正整数 b，则使用公钥加密得到密文 $c = (c_1, c_2)$ 的过程如下：

$$(c_1, c_2) = Enc(m) = (g^b, g^{ab} \cdot m) \ mod\ p$$

3.3 解密算法

对于密文 $c = (c_1, c_2)$ ，使用私钥解密得到明文消息 m 的过程如下：

$$m = Dec(c) = c_2 \cdot (c_1^a)^{-1} \ mod\ p$$

正确性

可以验证：$c_2 \cdot (c_1^a)^{-1} = m \cdot g^{ab} \cdot (g^{ab})^{-1} = m \ (mod \ p)$。

安全性

ElGamal 加密方案的安全性在于在已知 g 的情况下，公开 g^a 和 g^{ab} 难以求得 g^b，只要离散对数问题的求解是困难的，该加密方案就是安全的。

该方案同时满足语义安全性，因为每次加密都会**随机选取** b，可以保证同一消息的多次重复加密结果不确定。

乘法同态性

ElGamal 加密方案具有**乘法同态性**，在模 p 意义下，有：

$$Enc(m_0) \cdot Enc(m_1) = (g^{b_0+b_1}, g^{a(b_0+b_1)}) \cdot m_0 \cdot m_1 = Enc(m_0 m_1)$$

4. Paillier 加密方案

Paillier 加密方案由 Pascal Paillier 在 1999 年发表[4]，后经过多次优化，是目前效率较高的加法同态加密方案。

4.1 密钥生成算法

1. 选取两个不同的大素数 p, q，计算 $n = pq$，以及 $\lambda = lcm(p-1, q-1)$。
2. 随机选择 $g \in \mathbb{Z}_{n^2}^*$，满足 $gcd(\frac{g^\lambda \ mod \ n^2 - 1}{n}, n) = 1$。
3. 公钥为 n, g，私钥为 λ。

几点个人理解：

- λ 的选取目的是为了要满足方程 $g^\lambda \equiv 1 \ (mod \ n)$ ，并不一定非要取 $\lambda = lcm(p-1, q-1)$。

  由欧拉定理可以得到 $g^{\varphi(n)} = g^{(p-1)(q-1)} \equiv 1 \ (mod \ n)$，取 $\lambda = \varphi(n)$ 也是没问题的。

  这里取 $\lambda = lcm(p-1, q-1)$ 的主要目的是为了**减少计算量**，因为 λ 是 $\mathbb{Z}_n^*$ 里元素的**最高阶**，即 λ 为满足 $g^x \equiv 1 \ (mod \ n)$ 中的 x 可取的**最小值**（可以用中国剩余定理证明），显然指数越小计算复杂度越小。

  事实上，RSA 加密方案中也可以使用 $\lambda = lcm(p-1, q-1)$ 来替代 $\varphi(n)$ 来优化计算。

- g 的选取看上去很复杂，但实际上只是为了满足一些条件。考虑到 $g^\lambda \equiv 1 \ (mod \ n)$，即可以写成 $g^\lambda = 1 + kn$ 的形式，那么 $\frac{g^\lambda \ mod \ n^2 - 1}{n} = k$。这里选取 g 的条件是满足 k 和 n 互素，因为在后面解密算法中要求 k 在模 n 意义下的逆元，**不互素逆元不存在**。

  考虑到 g 作为公钥是公开的，因此在实际应用中可以直接取 $g = 1 + n$，就符合选取条件了。

4.2 加密算法

对于明文 $m \in \mathbb{Z}_n$，随机选择 $r \in \mathbb{Z}_n^*$，使用公钥进行加密得到密文 c 的过程如下：

$$c = Enc(m) = g^m r^n \ mod \ n^2$$

当选择 $g = 1 + n$ 时，对 $(1+n)^m$ 在模 n^2 的意义下进行二项式展开，加密过程可以简化为：

$$c = (1+n)^m r^n \ mod \ n^2 = (1 + mn) r^n \ mod \ n^2$$

4.3 解密算法

对于密文 c ，使用私钥解密得到明文消息 m 的过程如下：

$$m = Dec(c) = \frac{c^\lambda \ mod \ n^2 - 1}{n} \cdot (\frac{g^\lambda \ mod \ n^2 - 1}{n})^{-1} \ mod \ n$$

正确性

不妨设 $g^\lambda \ mod \ n^2 = 1 + kn$，可以计算出：

$$c^\lambda \ mod \ n^2 = g^{m\lambda} \cdot r^{n\lambda} = (1+kn)^m \cdot 1 = 1 + kmn$$

其中 $r^{n\lambda} = 1$ 是因为 $r^\lambda = 1 \ (mod \ n)$。此时在模 n 意义下可以验证：

$$Dec(c) = \frac{c^\lambda \ mod \ n^2 - 1}{n} \cdot (\frac{g^\lambda \ mod \ n^2 - 1}{n})^{-1} = km \cdot k^{-1} = m$$

安全性

Paillier 加密方案的安全性依赖于复杂剩余判断问题，此处即判断一个数是模 n^2 的 n 次剩余是困难的。

如果模 n^2 的 n 次剩余判断是可以快速解决的，那么 Paillier 加密方案就**不满足语义安全**，即不满足在**选择明文攻击**（Chosen-Plaintext Attack，CPA）下的安全性。

简单理解就是当攻击方拥有两个明文 m_0, m_1 时，给出其中一个明文的密文 c_b，攻击方可以判断出该密文是由哪个明文加密而来的。

一个**不严谨的证明**：不妨设密文为 $c_0 = g^{m_0} r_0^n \ mod \ n^2$，攻击方可以计算 $t_0 = c_0 \cdot (g^{m_0})^{-1} = r_0^n$ 以及 $t_1 = c_0 \cdot (g^{m_1})^{-1} = g^{m_0 - m_1} \cdot r_0^n$。显然 t_0 是一个模 n^2 的 n 次剩余，而 t_1 不是（或者大概率不是？没有证明过），攻击方如果可以快速判断 n 次剩余，就可以区分出密文由哪个明文加密而来的了。

加法同态性

Paillier 加密方案具有**加法同态性**，在模 n^2 意义下，有：

$$Enc(m_0) \cdot Enc(m_1) = g^{m_0} r_0^n \cdot g^{m_1} r_1^n = g^{m_0 + m_1} \cdot (r_0 r_1)^n = Enc(m_0 + m_1)$$

三、近似同态加密

本文主要介绍一个近似同态加密技术：BGN加密。

1. BGN 加密方案

BGN 加密方案由 Boneh、Goh 和 Nissim 在 2005 年发表[5]，该方案构造一个双线性映射，具有无限次加法同态和一次乘法同态操作。

1.1 密钥生成算法

1. 选取两个不同的大素数 p, q，计算 $n = pq$。
2. 选择两个 n 阶循环群 G, G_1，取 g 为 G 的一个生成元，构造双线性映射 $e : G \times G \to G_1$，满足：
   - **双线性**：$\forall a, b \in \mathbb{Z}_n$，有：$e(g^a, g^b) = e(g, g)^{ab}$。也即 $\forall u, v \in G$，有：$e(u^a, v^b) = e(u, v)^{ab}$。
   - **非退化性**：$e(g, g) \neq 1$。在 BGN 加密中还要求 $e(g, g)$ 为 G_1 的一个生成元。
   - **可计算性**：$\forall u, v \in G$，存在高效算法计算 $e(u, v)$。
3. 取 u 为 G 的另一个生成元，计算 $h = u^q$，则 h 为群 G 中的 p 阶元素，即 $h^p = u^{pq} = u^n = 1$。（模 n 意义下）
4. 公钥为 n, G, G_1, e, g, h，私钥为 p。

1.2 加密算法

对于明文 $m \in \mathbb{Z}_k$，随机选择 $r \in \mathbb{Z}_n$，使用公钥进行加密得到密文 c 的过程如下：

$$c = Enc(m) = g^m h^r \ mod \ n$$

注意： k 应当取一个较小的值才能进行解密，例如 $m \in \mathbb{Z}_1 = \{0, 1\}$。

1.3 解密算法

对于密文 c ，使用私钥解密得到明文消息 m 的过程如下：

$$m = Dec(c) = log_{g^p} c^p$$

注意：m 的取值空间较小时才能解密。

正确性

可以验证：

$$Dec(c) = log_{g^p} c^p = log_{g^p}(g^{mp} h^{rp}) = log_{g^p} g^{mp} = m$$

由于离散对数的计算困难，只有m 的取值空间较小时可以通过枚举解出 m 的值。

安全性

BGN 加密方案只有在子**群判定问题**困难的情况下才是语义安全的，可以参考 Paillier 加密算法的安全性分析。BGN 加密采用 h^r 来混淆，如果能快速判断 h^r 是否属于 G 的 p 阶子群，那么攻击方就可以区分出密文由哪个明文加密而来。

加法同态性

BGN 加密方案具有**加法同态性**，类似于 Paillier 加密方案，在模 n 意义下，有：

$$Enc(m_0) \cdot Enc(m_1) = g^{m_0} h^{r_0} \cdot g^{m_1} h^{r_1} = g^{m_0 + m_1} \cdot h^{r_0 + r_1} = Enc(m_0 + m_1)$$

乘法同态性

BGN 加密方案具有一次**乘法同态性**，不妨设 $h = u^q = g^{aq}$，$g_1 = e(g, g)$，$h_1 = e(g, h) = e(g, g^{aq}) = g_1^{aq}$。随机取 $r \in \mathbb{Z}_n$，则乘法同态操作为：

$$\begin{aligned}
e(c_0, c_1) \cdot h_1^r &= e(g^{m_0} h^{r_0}, g^{m_1} h^{r_1}) \cdot h_1^r \\
&= e(g^{m_0 + aqr_0}, g^{m_1 + aqr_1}) \cdot h_1^r \\
&= g_1^{m_0 m_1 + m_0 aqr_1 + m_1 aqr_0 + aaqqr_0 r_1} \cdot h_1^r \\
&= g_1^{m_0 m_1} \cdot g_1^{aq(m_0 r_1 + m_1 r_0 + aqr_0 r_1)} \cdot h_1^r \\
&= g_1^{m_0 m_1} \cdot h_1^{m_0 r_1 + m_1 r_0 + aqr_0 r_1 + r} \\
&= Enc_{G_1}(m_0 m_1)
\end{aligned}$$

最终可以获得群 G_1 中 $m_0 m_1$ 的加密密文，由于乘法同态使用双线性映射将密文空间从 G 转移至 G_1，且该转移只能进行一次，因此**乘法同态操作只有一次**。值得注意的是在转移后仍然支持无限次加法同态操作。

下一部分将介绍 Bootstrapping 和 DGHV、BGV 全同态加密方案。

参考文献

- [1] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” Commun. ACM, vol. 21, no. 2, pp. 120–126, Feb. 1978.
- [2] S. Goldwasser and S. Micali, “Probabilistic encryption & how to play mental poker keeping secret all partial information,” in Proceedings of the fourteenth annual ACM symposium on Theory of computing – STOC ’82, San Francisco, California, United States: ACM Press, 1982, pp. 365–377.
- [3] T. Elgamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” IEEE Transactions on Information Theory, vol. 31, no. 4, pp. 469–472, Jul. 1985.
- [4] P. Paillier, “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes,” in Advances in Cryptology — EUROCRYPT ’99, J. Stern, Ed., Berlin, Heidelberg: Springer, 1999, pp. 223–238.
- [5] D. Boneh, E.-J. Goh, and K. Nissim, “Evaluating 2-DNF Formulas on Ciphertexts,” in Theory of Cryptography, J. Kilian, Ed., Berlin, Heidelberg: Springer, 2005, pp. 325–341.

本文为作者在学习相关知识时的一种记录，便于以后的回顾。作者并没有系统地学习过密码学，因此在表述上可能会存在不严谨甚至出错的地方，文章仅供参考，欢迎大家与我交流，一起进步！

其他平台：

- 知乎（Totoro）：https://www.zhihu.com/people/totoro-14-60
- CSDN（_Totoro_）：https://blog.csdn.net/orz_Totoro
- B站（Totoro_134）：https://space.bilibili.com/279377771
- Github（Totoro134）：https://github.com/Totoro134
- 公众号（知识长生所）