Introducing Paillier Encryption Scheme

longcpp

longcpp9@gmail.com

March 18, 2020

Pascal Paillier 在其 1999 年的论文 "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes" 中引入了 Paillier 公钥加密机制, 其安全性由合成剩余类 (Composite Residuosity Class) 相关的困难问题保证. Paillier 的特殊之处是在概率加密的基础之上保留了加法同态的特性. 这一特性被广泛应用于构造各类密码协议, 例如近年来提出应用于数字货币领域的的两方 ECDSA 签名协议和多方阈值 ECDSA 签名协议.

1 Paillier 公钥加密机制

与其他公钥加密机制一样, Paillier 公钥加密机制也由密钥生成 (Key Generation) 加密 (Encryption) 和解密 (Decryption) 三个算法组成. 三个算法的详细计算过程在图 1 中展示, 图 1 截取自 Sigrun Goluch 的毕业论文 "The development of homomorphic cryptography". 为了理解图 1 中展示的过程, 需要先介绍 **Carmichael 函数**的概念.

1.1 Paillier 公钥加密算法

对于正整数 n, **Carmichael 函数** $\lambda(n)$ 表示满足下面要求的最小的正整数 t:

$$x^t = 1 \bmod n, \text{ 其中 } x \in \mathbb{Z} \text{ 并且 } \gcd(x, n) = 1.$$

对于素数 p, 根据费马小定理可知 $\lambda(p) = \varphi(p) = p - 1$, 其中 φ 表示欧拉函数. 根据代数基本定理可以对任意整数 n 做唯一分解, 记为 $n = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}$, 其中 $p_1, p_2, \cdots, p_k$ 为互不相同的素数. R. D. Carmichael 在 1912 年给出了求解 $\lambda(n)$ 的计算公式 (lcm 表示最小公倍数):

$$\lambda(n) = \mathsf{lcm}\left(\lambda\left(p_1^{a_1}\right), \lambda\left(p_2^{a_2}\right), \cdots, \lambda\left(p_k^{a_k}\right)\right), \lambda(p_i^{a_i}) = \begin{cases} 2^{a_i-2}, & \text{for } p_i = 2, a_i > 2 \\ (p_i - 1)p_i^{a_i-1}, & \text{otherwise} \end{cases}$$

Paillier 公钥加密机制中仅关心 $n = pq$ 且 p, q 为大素数的情形，根据上式有 $\lambda(n) = \mathsf{lcm}(p-1, q-1)$，并且 $\lambda(n^2) = \mathsf{lcm}(p(p-1), q(q-1)) = n \cdot \mathsf{lcm}(p-1, q-1) = n\lambda(n)$. 则对任意的 $x \in \mathbb{Z}_{n^2}^*$，都有

$$x^{n\lambda(n)} = 1 \bmod n^2.$$

由于对任意的 $x \in \mathbb{Z}_{n^2}^*$，$\mathsf{gcd}(x, n^2) = 1$ 都有 $\mathsf{gcd}(x, n) = 1$，则对于任意的 $x \in \mathbb{Z}_{n^2}^*$，都有

$$x^{\lambda(n)} = 1 \bmod n.$$

Key Generation: $\mathsf{KeyGen}(p, q)$	
Input: $p, q \in \mathbb{P}$	
Compute	$n = pq$
Choose $g \in \mathbb{Z}_{n^2}^*$ such that	$\gcd(L(g^\lambda \bmod n^2), n) = 1$ with $L(u) = \dfrac{u-1}{n}$
Output: (pk, sk) public key: $pk = (n, g)$ secret key: $sk = (p, q)$	

Encryption: $\mathsf{Enc}(m, pk)$	
Input: $m \in \mathbb{Z}_n$	
Choose	$r \in \mathbb{Z}_n^*$
Compute	$c = g^m \cdot r^n \bmod n^2$
Output: $c \in \mathbb{Z}_{n^2}$	

Decryption: $\mathsf{Dec}(c, sk)$	
Input: $c \in \mathbb{Z}_{n^2}$	
Compute	$m = \dfrac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n$
Output: $m \in \mathbb{Z}_n$	

Figure 1: Paillier 公钥加密机制

密钥生成算法 KeyGen 的输入为两个大素数 p, q，而 g 则是从 $\mathbb{Z}_{n^2}^*$ 中选取的满足 $\mathsf{gcd}\left(L\left(g^\lambda \mod n^2\right), n\right) = 1$ 的元素，其中的映射 $L : \mathbb{Z}_{n^2}^* \to \mathbb{Z}_n$ 定义为 $L(u) = \frac{u-1}{n}$. 值得注意的是，与通常的密码学构造书写规范不同的是，映射 L 的定义中 $\frac{u-1}{n}$ 的含义是，计算 $(u-1)$ 除以 n 的商，而非在 $\mathbb{Z}_{n^2}^*$ 中计算 $(u-1)$ 乘以 n 的逆. Paillier 机制的构造能够保证 $\frac{u-1}{n}$ 总是能够整除，后续会介绍. 映射 L 输入中的 λ 即为之前介绍的 Carmichael

函数 $\lambda(n)$. 关于 g 的约束条件 $\mathsf{gcd}\left(L\left(g^{\lambda} \mod n^2\right), n\right) = 1$ 可以确保 Paillier 加解密算法 Enc 和 Dec 能够正确执行, 后续讨论原理. 则 Paillier 加密机制的公钥 $pk = (n, g)$, 私钥 $sk = (p, q)$. 加密算法 $\mathsf{Enc}(m, pk)$(或者记为 $\mathsf{Enc}_{pk}(m)$) 将明文 $m \in \mathbb{Z}_n$ 映射成密文 $c \in \mathbb{Z}_{n^2}$:

$$c = g^m \cdot r^n \bmod n^2, r \in_R \mathbb{Z}_n^*.$$

其中 $r \in_R \mathbb{Z}_n^*$ 表示 r 是从 Z_n^* 中随机选择的元素, 确保了 Paillier 机制的概率加密特性. 解密算法 $\mathsf{Dec}(c, sk)$(或者记为 $\mathsf{Dec}_{sk}(c)$) 将密文 $c \in \mathbb{Z}_{n^2}$ 映射成明文 $m \in \mathbb{Z}_n$:

$$m = \left(L(c^{\lambda} \bmod n^2)/L(g^{\lambda} \bmod n^2)\right) \bmod n.$$

加密算法 $\mathsf{Enc}_{pk}(m)$ 和解密算法 $\mathsf{Dec}_{sk}(c)$ 的正确性将在下一节讨论, 在此之前, 先介绍 Paillier 机制的加法同态特性.

1.2 Paillier 加密机制的加法同态

Paillier 机制的特殊之处是在概率加密的基础之上提供了加法同态的特性. 假设

$$c_1 = \mathsf{Enc}_{pk}(m_1) = g^{m_1} r_1^n \bmod n^2,\ c_2 = \mathsf{Enc}_{pk}(m_2) = g^{m_2} r_2^n \bmod n^2,$$

则有

$$c_1 \cdot c_2 = g^{m_1} r_1^n \cdot g^{m_2} r_2^n \bmod n^2 = g^{m_1+m_2}(r_1 r_2)^n \bmod n^2 = \mathsf{Enc}_{pk}(m_1 + m_2).$$

在公钥相同时, 两个密文的乘积等于两个明文之和对应的密文. 值得提及的是, 同样可以利用 Paillier 机制进行同态减法 ($\mathbb{Z}_n$ 内的加法和减法实际为同一种运算):

$$c_1 \cdot c_2^{-1} = g^{m_1} r_1^n \cdot g^{-m_2} r_2^{-n} \bmod n^2 = g^{m_1-m_2}(r_1/r_2)^n \bmod n^2 = \mathsf{Enc}_{pk}(m_1 - m_2).$$

2 Paillier 加密机制原理

文章最开始提及 Paillier 公钥加密机制的安全性是由合成剩余类相关的困难问题保证的. 具体说来, 其安全性建立在区分模 n^2 的 n 次剩余和 n 次非剩余问题的困难性. 本节介绍模 n^2 的 n 次剩余并介绍 Paillier 加密机制的正确性和安全性.

2.1 模 n^2 的 n 次剩余

对于 $z \in \mathbb{Z}_{n^2}^*$, 如果存在 $y \in \mathbb{Z}_{n^2}^*$ 满足 $z = y^n \bmod n^2$, 则称 z 为**模 n^2 的 n 次剩余**, 而 y 则被称为 z 的 **n 次根**. 用 $\mathcal{R}_n$ 表示模 n^2 的 n 次剩余的集合. 容易看出 $\mathcal{R}_n$ 为 $\mathbb{Z}_{n^2}^*$ 的乘法子

群, 下面考虑其阶 $|\mathcal{R}_n|$. 为了计算 $|\mathcal{R}_n|$, 考虑群同态映射 $f(x)=x^n \bmod n^2:\mathbb{Z}_{n^2}^* \to \mathbb{Z}_{n^2}^*$. 映射 f 的核

$$\mathsf{Kernel}(f)=\{x\in Z_{n^2}:x^n=1 \bmod n^2\}.$$

考虑 $\mathbb{Z}_{n^2}^*$ 的代数结构:

$$\mathbb{Z}_{n^2}^* \cong \mathbb{Z}_n \times \mathbb{Z}_n^* \cong \mathbb{Z}_n \times \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1},$$

而 $\mathbb{Z}_n \times \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$ 中的单位元为 $(0,0,0)$, 则

$$\mathsf{Kernel}(f)=\{(a,b,c)\in \mathbb{Z}_n \times \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}:(na,nb,nc)=(0,0,0)\}.$$

对于任意的 $a\in\mathbb{Z}_n$ 都有 $na=0$. 由于 $\mathsf{gcd}(n,p-1)=1$ 和 $\mathsf{gcd}(n,q-1)=1$, 仅有 $b=0\in\mathbb{Z}_{p-1}, c=0\in\mathbb{Z}_{q-1}$ 满足 $nb=0$ 和 $nc=0$, 则有 $|\mathsf{Kernel}(f)|=n$. 根据

$$|\mathsf{Kernel}(f)|\times|\mathsf{Image}(f)|=|\mathbb{Z}_{n^2}^*|=n\varphi(n),$$

可知 $|\mathsf{Image}(f)|=n\varphi(n)/n=\varphi(n) \implies |\mathcal{R}_n|=\varphi(n)$.

下一个很自然的问题是对于 $\mathbb{Z}_{n^2}^*$ 中的 n 次剩余 z 在 $\mathbb{Z}_{n^2}^*$ 有多少个不同的 n 次根? 很自然的猜测是 n 个不同的 n 次根, 这也是正确答案. $|\mathbb{Z}_{p^2}^*|=p(p-1)$, $|\mathbb{Z}_{q^2}^*|=q(q-1)$ 以及 $\mathbb{Z}_{p^2}^*$ 和 $\mathbb{Z}_{q^2}^*$ 为循环群. 有限循环群 G 中方程 $y^n=z$ 有 $\mathsf{gcd}(n,|G|)$ 个不同的解. 由此可知 $z=y^n \bmod p^2$ 有 $\mathsf{gcd}(n,p(p-1))=p$ 个不同的解, 同理 $z=y^n \bmod q^2$ 有 $\mathsf{gcd}(n,q(q-1))=q$ 个不同的解. 根据中国剩余定理即可知道 $z=y^n \bmod n^2$ 有 $pq=n$ 个不同的解. 好消息是, n 个不同的解有特殊的形式.

对于任意的 $x\in\mathbb{Z}_n$ 都有 $(1+n)^x=1+xn \bmod n^2$, 通过归纳法容易证明. 对于 $x=0,1$ 的情况, $(1+n)^x=1+xn \bmod n^2$ 显然成立. 考虑 $x\to x+1$ 的情况,

$$(1+n)^{x+1}=(1+n)^x(1+1) \bmod n^2=(1+xn)(1+n)=(1+(x+1)n) \bmod n^2.$$

另外通过下面等式:

$$(1+n)^{xn}=(1+xn)^n \bmod n^2=1+(xnn) \bmod n^2=1 \bmod n^2.$$

可知, $1+xn, \forall x\in\mathbb{Z}_n$ 是 $\mathbb{Z}_{n^2}^*$ 中单位元的 n 个不同的 n 次根. 可以注意到, 单位元的 n 个不同的 n 次根中, 仅有一个小于 n, 也即 1.

2.2 Paillier 加密机制的正确性

介绍完模 n^2 的 n 次剩余, 接下来考察 Paillier 加密机制的 KeyGen 算法中关于 g 的约束条件 $\mathsf{gcd}\left(L\left(g^\lambda \mod n^2\right),n\right)=1$ 保证 Paillier 加解密算法 Enc 和 Dec 能够正确执行的原理.

Paillier 加密算法 Enc_{pk} 可以看成是从 $\mathbb{Z}_n \times \mathbb{Z}_n^*$ 到 $\mathbb{Z}_{n^2}^*$ 的映射, 其中明文 $m \in \mathbb{Z}_n$, 随机数 $r \in \mathbb{Z}_n^*$, 而密文 $c = g^m r^n \bmod n^2 \in \mathbb{Z}_{n^2}^*$. 由于 $|\mathbb{Z}_n \times \mathbb{Z}_n^*| = n\varphi(n) = \varphi(n^2) = |\mathbb{Z}_{n^2}^*|$, 为了确保加解密算法的正确性只需保证 Enc_{pk} 是 $\mathbb{Z}_n \times \mathbb{Z}_n^*$ 到和 $\mathbb{Z}_{n^2}^*$ 之间的单射. 而当 $g \in \mathbb{Z}_{n^2}^*$ 满足 $n|\ \mathsf{order}(g)$, $\mathsf{order}(g) \neq 0$ 时可以保证 Enc_{pk} 为单射. 而 $\mathsf{gcd}\left(L\left(g^\lambda \mod n^2\right), n\right) = 1$ 等价于 $n|\ \mathsf{order}(g)$. 接下来介绍最后两个结论是如何推演出来的.

首先证明当 g 满足 $n|\ \mathsf{order}(g)$, $\mathsf{order}(g) \neq 0$ 时, Enc_{pk} 是 $\mathbb{Z}_n \times \mathbb{Z}_n^*$ 到和 $\mathbb{Z}_{n^2}^*$ 之间的单射. 用反正法来证明. 假设存在 $m_1, m_2 \in \mathbb{Z}_n$ 并且 $r_1, r_2 \in \mathbb{Z}_n^*$ 满足

$$g^{m_1} r_1^n = g^{m_2} r_2^n \bmod n^2.$$

等式两边同时乘以 g^{-m_2} 和 r_1^{-n} 可以得到:

$$g^{m_1 - m_2} = (r_2/r_1)^n \bmod n^2 \implies \left(g^{m_1 - m_2}\right)^{\lambda(n)} = (r_2/r_1)^{n\lambda(n)} \bmod n^2,$$

根据前述的关于 Carmichael 函数的结论 $(r_2/r_1)^{n\lambda(n)} = 1 \bmod n^2$, 则有

$$g^{(m_1 - m_2)\lambda(n)} = 1 \bmod n^2 \implies \mathsf{order}(g)|(m_1 - m_2)\lambda(n),$$

由于 $n|\ \mathsf{order}(g)$, $\mathsf{order}(g) \neq 0$, 则有 $n|(m_1 - m_2)\lambda(n)$, 考虑到 $m_1, m_2 \in \mathbb{Z}_n$, 则有 $m_1 = m_2 \bmod n$. 在这一前提下有

$$g^{m_1} r_1^n = g^{m_2} r_2^n \bmod n^2 \implies r_1^n = r_2^n \bmod n^2 \implies (r_1/r_2)^n = 1 \bmod n^2.$$

由于 $r_1, r_2 \in \mathbb{Z}_n^*$ 根据前一小节关于 $\mathbb{Z}_{n^2}^*$ 中单位元的 n 次根的结论可知:

$$r_1/r_2 = 1 \in Z_n^* \implies r_1 = r_2 \bmod n,$$

证明完成. 也即当 g 满足 $n|\ \mathsf{order}(g)$, $\mathsf{order}(g) \neq 0$ 时, 可以保证 Paillier 加密算法 Enc_{pk} 的单射特性.

但是如何选择满足上述条件的 g, 前述的约束条件并没有给出如何验证选择的 g 是否满足条件. 与前述条件等价的约束条件 $\mathsf{gcd}\left(L\left(g^\lambda \mod n^2\right), n\right) = 1$ 给出了切实可行的验证过程. 我们来证明这两个约束条之间确实等价. 为了方便叙述, 构造集合 $\mathcal{B}_\alpha$ 为所有阶为 $n\alpha$ 的元素集合:

$$\mathcal{B}_\alpha = \{g \in \mathbb{Z}_{n^2}^* |\ \mathsf{order}(g) = n \cdot \alpha, \alpha \in \{1, \ldots, \lambda(n)\}\} \subset \mathbb{Z}_{n^2}^*,$$

记 $\mathcal{B}$ 为 $\mathcal{B}_\alpha, \alpha \in \{1, \ldots, \lambda(n)\}$ 的并集. 则约束条件 g 满足 $n|\ \mathsf{order}(g)$, $\mathsf{order}(g) \neq 0$ 等价于 $g \in \mathcal{B}$.

先证明 $g \in \mathcal{B} \implies \mathsf{gcd}\left(L\left(g^{\lambda} \mod n^2\right), n\right) = 1$. 对于任意的 $x \in \mathbb{Z}_{n^2}^*$ 根据之前的结论, 都有 $x^{n\lambda(n)} = 1 \bmod n^2$. 由于 $g \in \mathcal{B}$ 则有

$$g^{\lambda(n)} = 1 \bmod n^2 \implies g^{\lambda(n)} = 1 \bmod n.$$

则存在 $k \in \mathbb{Z}_n$ 满足

$$g^{\lambda(n)} = (1 + kn) \bmod n^2,$$

值得注意的是, 根据 L 的定义有

$$k = \frac{g^{\lambda(n)} - 1}{n} = L\left(g^{\lambda(n)} \bmod n^2\right).$$

接下来考察 $\mathsf{gcd}\left(L\left(g^{\lambda} \mod n^2\right), n\right)$ 也即 $\mathsf{gcd}(k, n)$. 如果 $\mathsf{gcd}(k, n) = b > 1$, 则存在 $a < n$ 满足 $n|(ak)$, 则有:

$$g^{a\lambda(n)} = (1 + kn)^a \bmod n^2 = 1 + (ak)n \bmod n^2 = 1 \bmod n^2 \implies g \notin \mathcal{B},$$

这与 $g \in \mathcal{B}$ 的前提条件矛盾, 因此有 $\mathsf{gcd}(k, n) = 1$. 证毕.

接下来考虑 $\mathsf{gcd}\left(L\left(g^{\lambda(n)} \mod n^2\right), n\right) = 1 \implies g \in \mathcal{B}$ 的证明. 仍然记 $k = \left(g^{\lambda(n)} - 1\right)/n$, 则有 $g^{\lambda(n)} = (1 + kn) \bmod n^2$. 为了计算 $\mathsf{order}(g)$, 考虑使 $\left(g^{\lambda(n)}\right)^a = 1 \bmod n^2, a \neq 0$ 的条件:

$$g^{a\lambda(n)} = (1 + kn)^a \bmod n^2 = 1 + (ak)n \bmod n^2$$

由于 $\mathsf{gcd}(k, n) = 1$, 为了是 $n|(ak)$, a 必须为 n 的非零整数倍, 也即 $g \in \mathcal{B}$.

至此, 可以总结 Paillier 加密机制的 KeyGen 算法中的约束 $\mathsf{gcd}\left(L\left(g^{\lambda(n)} \mod n^2\right), n\right) = 1$ 保证了所选取的 $g \in \mathbb{Z}_{n^2}^*$ 的阶是 n 的非零整数倍, 进而保证了 Enc_{pk} 是 $\mathbb{Z}_n \times \mathbb{Z}_n^*$ 到和 $\mathbb{Z}_{n^2}^*$ 之间的单射. 这就使得对密文的唯一解密成为可能, 但仍然有一个问题需要澄清: Dec_{sk} 是否真的能够解密得到明文 m? 也即

$$\mathsf{Dec}_{sk}\left(\mathsf{Enc}_{pk}(m)\right) = m$$

是否成立.

接下来考察 Pallier 解密算法的正确性. 首先引入 n 次剩余类 (n^{th} Residuosity Class) 的概念, 假设 $g \in \mathcal{B}$, $c \in \mathbb{Z}_{n^2}^*$, 如果存在 $r \in \mathbb{Z}_n^*$ 使得 $m \in \mathbb{Z}_n$ 满足

$$c = g^m r^n \bmod n^2,$$

注意 m 值是唯一的, 称 m 为 c 的关于 g 的 n 次剩余类, 记为 $[c]_g = m$. 容易验证, c 为 $\mathbb{Z}_{n^2}^*$ 中的 n 次剩余类等价于 $[c]_g = 0$ 以及 $[g]_g = 1$. 设对于任意的 $c \in \mathbb{Z}_{n^2}^*$ 和 $g_1, g_2 \in \mathcal{B}$, 存在 r_1, $r_2 \in \mathbb{Z}_n^*$ 以及 $m_1 = [c]_{g_1}, m_2 = [c]_{g_2}$ 满足

$$c = g_1^{m_1} r_1^n \bmod n^2,\ c = g_2^{m_2} r_2^n \bmod n^2 \tag{1}$$

令 $m_3 = [g_2]_{g_1}$, 则存在 $r \in \mathbb{Z}_n^*$ 满足

$$g_2 = g_1^{m_3} r^n \bmod n^2 \tag{2}$$

组合公式 (1) 和公式 (2), 得到

$$c = g_1^{m_1} r_1^n \bmod n^2 = (g_1^{m_3} r^n)^{m_2} r_2^n \bmod n^2 = g_1^{m_2 m_3} (r^{m_2} r_2)^n \bmod n^2 \implies m_1 = m_2 m_3.$$

由此有

$$[c]_{g_1} = [c]_{g_2} [g_2]_{g_1} \mod n, \; [c]_{g_2} = [c]_{g_1} [g_2]_{g_1}^{-1} \mod n. \tag{3}$$

而当取 $c = g_1$ 时, 根据公式 (3) 有

$$[g_1]_{g_1} = [g_1]_{g_2} [g_2]_{g_1} \bmod n \implies 1 = [g_1]_{g_2} [g_2]_{g_1} \bmod n \implies [g_1]_{g_2} = [g_2]_{g_1}^{-1} \bmod n \tag{4}$$

借助 n 次剩余类的概念, 可以看出 Paillier 解密算法的正确性. 对于任意的 $c \in \mathbb{Z}_{n^2}^*$, 有如下结论:

$$L(c^{\lambda(n)} \bmod n^2) = \lambda(n) [c]_{1+n} \bmod n. \tag{5}$$

这是因为 $(1+n)^n = 1 \bmod n^2$, 也即 $(1+n) \in \mathcal{B}$, 根据之前的结论, 存在唯一的 $(a, b) \in \mathbb{Z}_n \times \mathbb{Z}_n^*$ 满足

$$c = (1+n)^a b^n \bmod n^2,$$

也即 $a = [c]_{1+n}$, 从而有 (注意到 $b^{n\lambda(n)} = 1 \bmod n^2$):

$$c^{\lambda(n)} = (1+n)^{a\lambda(n)} b^{n\lambda(n)} = (1+n)^{a\lambda(n)} = 1 + a\lambda(n) n \bmod n^2,$$

从而有

$$L\left(c^{\lambda(n)} \bmod n^2\right) = L\left(1 + a\lambda(n) n \bmod n^2\right) = \lambda(n) a \bmod n = \lambda(n) [c]_{1+n} \bmod n.$$

则根据公式 (3), 公式 (4) 和公式 (5) 有:

$$\mathsf{Dec}_{sk}(c) = \frac{L\left(c^{\lambda(n)} \bmod n^2\right)}{L\left(g^{\lambda(n)} \bmod n^2\right)} = \frac{\lambda(n)[c]_{1+n}}{\lambda(n)[g]_{1+n}} = \frac{[c]_{1+n}}{[g]_{1+n}} = [c]_{1+n} [1+n]_g = [c]_g \bmod n. \tag{6}$$

由此验证了 Paillier 解密算法 Dec_{sk} 的正确性:

$$\mathsf{Dec}_{sk}\left(\mathsf{Enc}_{pk}(m)\right) = m.$$

借助 n 次剩余类的概念, 容易看出 Paillier 加密机制的所支持的加法同态特性: 对于任意的 $g \in \mathcal{B}$, 映射 $c \to [c]_g$ 是从 $(\mathbb{Z}_{n^2}^*, \cdot)$ 到 $(\mathbb{Z}_n, +)$ 的同态映射. 这意味着对于任意的 $c_1, c_2 \in \mathbb{Z}_{n^2}^*$, 都有

$$[c_1 \cdot c_2]_g = [c_1]_g + [c_2]_g \bmod n.$$

证明过程比较简单. 设对于 $m_1, m_2 \in \mathbb{Z}_n$ 存在 $r_1, r_2 \in \mathbb{Z}_n^*$ 使得:

$$c_1 = g^{m_1} r_1^n \bmod n^2, c_2 = g^{m_2} r_2^n \bmod n^2,$$

则对于 $c = c_1 \cdot c_2$, 就有 $r = r_1 \cdot r_2$ 满足:

$$c_1 \cdot c_2 = g^{m_1+m_2}(r_1 \cdot r_2)^n \bmod n.$$

2.3 Paillier 加密机制的安全性

Paillier 公钥加密机制的安全性, 依赖于两个计算问题: 合成剩余类问题 (Composite Residuosity Class Problem) 以及合成剩余问题 (Composite Residuosity Problem), 用 $\mathsf{CLASS}[n, g]$ 表示合成剩余类问题, 即给定 $c \in \mathbb{Z}_{n^2}^*$ 以及 $g \in \mathcal{B}$, 计算 $[c]_g$ 的问题. 用 $\mathsf{CR}[n]$ 表示合成剩余问题, 即给定 $c \in \mathbb{Z}_{n^2}^*$, 判断 c 是否是 $\mathbb{Z}_{n^2}^*$ 中的 n 次剩余. Pascal Paillier 在 EUROCRYPT'99 上推断 $\mathsf{CR}[n]$ 是困难的, 也即 **DCRA 假设 (Decisional Composite Residuosity Assumption): 不存在多项式时间算法可以解决 $\mathsf{CR}[n]$, e.g. $\mathsf{CR}[n]$ 是计算困难的.**

注意 $\mathsf{CLASS}[n, g]$ 与 Pallier 解密算法 Dec_{sk} 之间的关系. 值得注意的是 $\mathsf{CLASS}[n, g]$ 的难度与 $g \in \mathcal{B}$ 的具体选择无关, 并且对于所有的 $c \in \mathbb{Z}_{n^2}^*$, $\mathsf{CLASS}[n, g]$ 都一样困难. 也即 $\mathsf{CLASS}[n, g]$ 的计算难度仅与 n 有关系, 简记为 $\mathsf{CLASS}[n]$. 另外 $\mathsf{CLASS}[n]$ 问题与 n 的分解问题 $\mathsf{Factor}[n]$ 一样困难. 这是因为如果知道了 n 的分解 $n = pq$, 则很容易计算 $\lambda(n) = \mathsf{lcm}(p-1, q-1)$, 则根据公式 (6) 容易计算出 $[c]_g \bmod n$, 也即

$$\mathsf{CLASS}[n] \Leftarrow \mathsf{Factor}[n].$$

困难问题 $\mathsf{CR}[n]$ 与 $\mathsf{CLASS}[n]$ 之间关系可以通过 $\mathsf{CLASS}[n]$ 的判定问题 $\mathsf{D\text{-}CLASS}[n]$ 联系起来. $\mathsf{D\text{-}CLASS}[n]$ 表示给定 $c \in \mathbb{Z}_{n^2}^*$, $g \in \mathcal{B}$, $m \in \mathbb{Z}_n$, 判定 m 是否等于 $[c]_g$, 则 $\mathsf{CR}[n] \equiv \mathsf{D\text{-}CLASS}[n]$. 两个问题的等价性可以通过如下方法证明. 如果有预言机 (Oracle) 能够解决 $\mathsf{CR}[n]$ 问题, 可以向预言机查询 $c \cdot g^{-m} \bmod n^2$ 是否是 n 次剩余. 如果 $c \cdot g^{-m} \bmod n^2$ 是 n 次剩余则有

$$c \cdot g^{-m} = g^{[c]_g - m} r^n \bmod n^2 \implies [c]_g = m,$$

也即解决了 $\mathsf{D\text{-}CLASS}[n]$ 问题. 反之, 如果有能够解决 $\mathsf{D\text{-}CLASS}[n]$ 问题的预言机, 则随机选择 $g \in \mathcal{B}$ 并向预言机提交查询 $(c, g, 0)$, 如果返回 '是' 的答案, 则 c 是 $\mathbb{Z}_{n^2}^*$ 中的 n 次剩余, 也即解决了 $\mathsf{CR}[n]$ 问题另外显然有 $\mathsf{D\text{-}CLASS}[n] \Leftarrow \mathsf{CLASS}[n]$, 因为验证答案总是比计算答案更容易, 综上就有:

$$\mathsf{CR}[n] \equiv \mathsf{D\text{-}CLASS}[n] \Leftarrow \mathsf{CLASS}[n] \Leftarrow \mathsf{Factor}[n].$$

基于该计算层级, Pascal Paillier 推断计算合成剩余问题 (Computational Composite Residuosity Problem) 是困难的, 也即 **CCRA 假设 (Computational Composite Residuosity Assumption): 不存在概率多项式时间算法能够解决计算合成剩余问题, e.g. $\mathsf{CLASS}[n]$ 是计算困难的.**

CCRA 假设确保了则 Paillier 加密机制的单向性, 因为解密就是在有限门信息 ($n = pq$) 的情况下解决 $\mathsf{CR}[n]$ 的过程. 而 DCRA 假设则确保了 Paillier 加密机制的语义安全 (Semantically Secure), 这是因为假设 c 是两个已知明文 m_0 和 m_1 中的一个对应的密文, 则 c 是明文 m_0 对应的密文等价于 $cg^{-m_0} \bmod n^2$ 是 n 次剩余, 反之亦然.