**Scheme** Paillier

An additively homomorphic, probabilistic public-key encryption scheme based on the difficulty of computing discrete logarithms. The scheme is parameterized by the bit-length $k$ of the underlying primes $p$ and $q$.

$\textsf{KeyGen}_k() \dashrightarrow (sk, pk)$

1: Choose[1] two $k$-bit prime numbers $p$ and $q$.
2: $n \leftarrow pq$, the public key.
3: $\lambda \leftarrow \mathsf{lcm}(p-1, q-1)$.
4: $\mu \leftarrow \mathsf{quot}((n+1)^{\lambda} \mod n^2)^{-1} \mod n$. ABORT if no inverse.
**return** $sk \equiv (\lambda, \mu)$ and $pk \equiv n$ as secret and public key respectively.

$\textsf{Encrypt}(pk, m \in \mathbb{Z}_n) \dashrightarrow [\![m]\!]$

1: Sample $r \xleftarrow{\$} \mathbb{Z}_n^*$ s.t. $\gcd(r, n) = 1$.
2: $[\![m]\!] \leftarrow r^n (n+1)^m \mod n^2$, the ciphertext of $m$.
**return** $c \equiv [\![m]\!]$

$\textsf{Decrypt}(pk, sk, c \in \mathbb{Z}_{n^2}^*) \dashrightarrow m$

1: $m \leftarrow \mathsf{quot}(c^{\lambda} \mod n^2)\mu \mod n$, the decryption of $[\![m]\!]$.
**return** $m$.

$\textsf{Add}(pk, [\![m_1]\!] \equiv c_1 \in \mathbb{Z}_{n^2}^*, [\![m_2]\!] \equiv c_2 \in \mathbb{Z}_{n^2}^*) \dashrightarrow [\![m_1 + m_2]\!]$

1: $c_{sum} \leftarrow c_1 c_2 \mod n^2$.
**return** $c_{sum} \equiv [\![m_1 + m_2]\!]n$.

$\textsf{ScalarMultiply}(pk, s \in \mathbb{Z}_n, [\![m]\!] \equiv c \in \mathbb{Z}_{n^2}^*) \dashrightarrow [\![s \cdot m]\!]$

1: $c_{mult} \leftarrow (c(n+1)^s) \mod n^2$.
**return** $c_{mult} \equiv [\![s \cdot m]\!]n$.

# References

[1] In practice we resort to a secure prime number generator from standard libraries. Note that $\gcd(pq, (p-1)(q-1)) = 1$ holds because both primes are of equal length.