1 SM3 算法的数学推导与实现思路

1.1 数学推导与表示

SM3 是中国国家密码管理局于 2010 年公布的密码杂凑算法，其输出长度为 256 位。算法主要包括消息填充、消息分组、消息扩展以及压缩函数四个步骤。数学推导如下：

1.1.1 消息填充

设消息为比特串 M，长度为 l 位：

1. 在消息末尾填充一个比特 1。
2. 填充 k 个比特 0，其中 k 为满足 $(l+1+k) \equiv 448 \pmod{512}$ 的最小非负整数。
3. 在末尾附加 l 的 64 位大端表示，得到填充后的消息 M'。

$$M' = M \parallel 1 \parallel 0^k \parallel \mathrm{len}(M)_{64}$$

1.1.2 消息分组

将 M' 按 512 位划分为 n 个消息分组：

$$M' = B^{(0)} \parallel B^{(1)} \parallel \cdots \parallel B^{(n-1)}, \quad B^{(i)} \in \{0,1\}^{512}$$

1.1.3 消息扩展

对于每个分组 $B^{(i)}$：

$$W_0^{(i)}, W_1^{(i)}, \ldots, W_{15}^{(i)} = \mathrm{Cut}_{32}(B^{(i)}) \tag{1}$$

$$W_j^{(i)} = P_1\big(W_{j-16}^{(i)} \oplus W_{j-9}^{(i)} \oplus (W_{j-3}^{(i)} \lll 15)\big) \tag{2}$$

$$\oplus\, (W_{j-13}^{(i)} \lll 7) \oplus W_{j-6}^{(i)}, \quad j = 16, \ldots, 67 \tag{3}$$

$$W_j'^{(i)} = W_j^{(i)} \oplus W_{j+4}^{(i)}, \quad j = 0, \ldots, 63 \tag{4}$$

其中：

$$P_1(X) = X \oplus (X \lll 15) \oplus (X \lll 23)$$

符号 $\lll$ 表示循环左移。

1.1.4 压缩函数

设初始向量：

$$V^{(0)} = (A, B, C, D, E, F, G, H) = \text{IV}$$

对于每个消息分组 $B^{(i)}$：

$$A \leftarrow V_0^{(i)}, \quad B \leftarrow V_1^{(i)}, \quad \ldots, \quad H \leftarrow V_7^{(i)} \tag{5}$$

$$\text{for } j = 0 \text{ to } 63: \tag{6}$$

$$T_j = \begin{cases} 0x79CC4519, & 0 \le j \le 15 \\ 0x7A879D8A, & 16 \le j \le 63 \end{cases} \tag{7}$$

$$SS1 = ((A \lll 12) + E + (T_j \lll j)) \lll 7 \tag{8}$$

$$SS2 = SS1 \oplus (A \lll 12) \tag{9}$$

$$TT1 = FF_j(A, B, C) + D + SS2 + W_j' \tag{10}$$

$$TT2 = GG_j(E, F, G) + H + SS1 + W_j \tag{11}$$

$$D \leftarrow C, \quad C \leftarrow B \lll 9, \quad B \leftarrow A, \quad A \leftarrow TT1 \tag{12}$$

$$H \leftarrow G, \quad G \leftarrow F \lll 19, \quad F \leftarrow E, \quad E \leftarrow P_0(TT2) \tag{13}$$

其中：

$$FF_j(X, Y, Z) = \begin{cases} X \oplus Y \oplus Z, & 0 \le j \le 15 \\ (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z), & 16 \le j \le 63 \end{cases}$$

$$GG_j(X, Y, Z) = \begin{cases} X \oplus Y \oplus Z, & 0 \le j \le 15 \\ (X \wedge Y) \vee (\neg X \wedge Z), & 16 \le j \le 63 \end{cases}$$

$$P_0(X) = X \oplus (X \lll 9) \oplus (X \lll 17)$$

最后：

$$V^{(i+1)} = V^{(i)} \oplus (A, B, C, D, E, F, G, H)$$

最终的哈希值为：

$$\text{SM3}(M) = V^{(n)}$$

1.2 实现与优化思路

- **基本实现**：按照上述数学推导，使用 `uint32_t` 类型保存 32 位字，逐步实现填充、分组、扩展与压缩。
- **循环展开**：将消息扩展与压缩循环部分进行循环展开（loop unrolling），减少分支判断开销。

- **内联函数**：将 P_0、P_1、FF_j、GG_j 等函数声明为 `static inline`，减少函数调用开销。
- **指令级并行**：在支持的编译器下利用 SIMD（如 SSE/AVX/NEON）优化消息扩展和压缩函数的运算。
- **内存对齐**：在消息数组分配时进行 32 位或 64 位对齐，减少 CPU 访存延迟。
- **编译优化**：开启编译器优化选项（如 `-O2` 或 `-O3`），结合 `-march=native` 利用硬件特性。