SM4 密码算法 S 盒的量子电路实现

罗庆斌[1,2]，李晓瑜[2*]，杨国武[3]

(1. 湖北民族大学信息工程学院 湖北 恩施 445000；2. 电子科技大学信息与软件工程学院 成都 610054；
3. 电子科技大学计算机科学与工程学院 成都 611731)

【摘要】SM4 密码算法是我国国家密码管理局 2006 年公开发布的用于 WAPI 的分组密码算法，2021 年 6 月成为国际标准。S 盒作为唯一的非线性组件，其安全性直接影响到 SM4 算法的安全性。该文首次给出 SM4 密码算法 S 盒的量子电路实现。根据 S 盒的代数表达式，首先利用高斯消元法给出表达式中仿射变换的量子电路，然后把求逆元运算转换为求该元素的 254 次方，再分别给出对应的平方计算和乘法计算的量子电路，最后通过改进的 Itoh-Tsujii 算法给出 S 盒的量子电路。量子电路的复杂度分析表明：所给出的 S 盒的量子电路共用 48 个量子比特，592 个量子门，电路深度为 289，具有较高的效率。该研究将会对量子环境下 SM4 密码算法的安全性分析奠定基础。

关 键 词 代数运算； 量子电路； S 盒； SM4
中图分类号 TP309 **文献标志码** A doi:10.12178/1001-0548.2021252

Quantum Circuit Implementation of S-box for SM4 Cryptographic Algorithm

LUO Qingbin[1,2], LI Xiaoyu[2*], and YANG Guowu[3]

(1. School of Information Engineering, Hubei Minzu University Enshi Hubei 445000;
2. School of Information and Software Engineering, University of Electronic Science and Technology of China Chengdu 610054;
3. School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 611731)

Abstract SM4 cryptographic algorithm is a block cipher algorithm for WAPI published by China's state cryptography administration in 2006. It was published as an international standard in June 2021. As the only nonlinear component, the security of S-box directly affects the security of SM4 algorithm. In this paper, the quantum circuit implementation of S-box for SM4 cryptographic algorithm is given for the first time. According to the algebraic expression of S-box, firstly, the affine transformation quantum circuit is given by Gaussian elimination method, then the inverse element operation is converted to the 254 power of the corresponding element, and then the corresponding quantum circuits for square calculation and multiplication calculation are given respectively. Finally, the quantum circuit of S-box is given by using improved Itoh-Tsujii algorithm. The complexity analysis shows that the given S-box quantum circuit uses 48 qubits, 592 quantum gates and the circuit depth is 289, which has high efficiency. The research of this paper will lay a foundation for the security analysis of SM4 cryptographic algorithm in quantum environment.

Key words algebraic operation; quantum circuit; S box; SM4

量子计算极大地改变了现代密码系统的安全性。对于非对称密码系统，Shor 算法[1] 能快速破解基于大整数分解的密码算法[2] 和基于离散对数的密码算法[3]。量子计算对于对称密码系统的影响虽然没有像非对称密码系统那样显著，但最近也得到大量关注。文献 [4] 首先分析了 Grover 算法[5] 对对称密码系统的量子威胁，建议把原来的密钥长度至少增加一倍。文献 [6] 利用 Simon 算法[7] 寻找碰撞周期以攻击对称密码系统，一些在经典环境中需要 $\Omega(2^{n/2})$次查询的密码算法在该量子环境下只需要 $O(n)$次查询。文献 [8] 结合 Grover 算法和 Simon 算法对具有 FX 结构的对称密码在量子环境下的安全性进行了分析。文献 [9] 结合 Grover 算法和 Simon 算法对具有 Feistel 结构的对称密码算法提出了量

收稿日期：2021－09－01；修回日期：2021－09－29
基金项目：国家重点研发计划 (2018YFA0306703)；国家自然科学基金 (61772006)；湖北省自然科学基金 (2020CFB326)；广西省自然科学基金 (2019GXNSFAA185033)；福建省自然科学基金 (2020J01812)
作者简介：罗庆斌 (1987－)，男，博士，主要从事量子计算和量子密码方面的研究.
*通信作者：李晓瑜，E-mail：xiaoyuuestc@uestc.edu.cn

子密钥恢复攻击。文献 [10] 通过在量子环境下利用 Demirci-Selçuk 中间人攻击和提高解 S 盒差分方程的效率分析了 AES 密码算法在量子环境下的安全性。文献 [11] 利用 Grover 算法和 Simon 算法为 SM4 密码算法构造了一个 8 轮的量子区分器。

这些对称密码在量子环境下的安全性分析基本是建立在它们已经可以用量子电路实现的假设上的。但是目前针对对称密码算法的量子电路实现研究较少，且基本是对 AES 密码算法的量子电路实现[12-13]，其他对称密码算法量子电路实现的研究几乎没有。

SM4 算法是用于 WAPI 的分组密码算法，2006 年由我国国家密码管理局公开发布[14]，2012 年 3 月发布成为国家密码行业标准[15](标准号为 GM/T 0002-2012)，2016 年 8 月发布成为国家标准[16](标准号为 GB/T 32907-2016)，2021 年 6 月发布成为国际标准[17](标准号为 ISO/IEC 18033-3：2010/AMD1：2021)。SM4 算法的安全性自发布以来，得到了广泛的关注。在这些安全性分析中，SM4 算法中唯一的非线性变换 S 盒起着关键性的作用。文献 [18] 分析了 SM4 算法中最小活跃 S 盒个数与抵抗差分攻击轮数之间的关系。文献 [19-20] 分别提出了基于掩码的 S 盒实现方案和基于门限理论的 S 盒的实现方案，以提高 SM4 密码算法的安全性。

本文主要通过 SM4 密码算法 S 盒的代数结构，使用 NOT 门、CNOT 门和 Toffoli 门构建实现 S 盒的量子电路。

1　SM4 密码算法

SM4 算法是分组密码算法，其数据分组长度和密钥分组长度都为 128 bit。加密算法与解密算法都以字节 (8 位) 和字 (32 位) 为单位进行数据处理。

1.1　加密算法

加密算法采用 32 轮迭代结构，每轮使用一个轮密钥。

设输入明文为$(\boldsymbol{X}_0,\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{X}_3)\in(\mathrm{GF}(2^{32}))^4$，输出的密文为$(\boldsymbol{Y}_0,\boldsymbol{Y}_1,\boldsymbol{Y}_2,\boldsymbol{Y}_3)\in(\mathrm{GF}(2^{32}))^4$，$\mathbf{rk}_i\in\mathrm{GF}(2^{32})$ $(i=0,1,2,\cdots,31)$为轮密钥。加密算法可描述如下：

$$\begin{aligned}&\boldsymbol{X}_{i+4}=F(\boldsymbol{X}_i,\boldsymbol{X}_{i+1},\boldsymbol{X}_{i+2},\boldsymbol{X}_{i+3},\mathbf{rk}_i)=\\&\boldsymbol{X}_i\oplus T(\boldsymbol{X}_{i+1}\oplus\boldsymbol{X}_{i+2}\oplus\boldsymbol{X}_{i+3}\oplus\mathbf{rk}_i)\end{aligned}\tag{1}$$

式中，$i=0,1,2,\cdots,31$，则有：

$$\begin{aligned}&(\boldsymbol{Y}_0,\boldsymbol{Y}_1,\boldsymbol{Y}_2,\boldsymbol{Y}_3)=(\boldsymbol{X}_{35},\boldsymbol{X}_{34},\boldsymbol{X}_{33},\boldsymbol{X}_{32})=\\&R(\boldsymbol{X}_{32},\boldsymbol{X}_{33},\boldsymbol{X}_{34},\boldsymbol{X}_{35})\end{aligned}\tag{2}$$

式中，F 是轮函数，变换$T:\mathrm{GF}(2^{32})\to\mathrm{GF}(2^{32})$由非线性变换$\tau$和线性变换 L 构成，即：

$$T(\cdot)=L(\tau(\cdot))\tag{3}$$

变换τ由 4 个 8 bit 的非线性变换 S 盒并行构成。设$\boldsymbol{A}=(\alpha_0,\alpha_1,\alpha_2,\alpha_3)\in(\mathrm{GF}(2^8))^4$是非线性变换$\tau$的输入，$\boldsymbol{B}=(\beta_0,\beta_1,\beta_2,\beta_3)\in(\mathrm{GF}(2^8))^4$是输出，则变换$\tau$定义如下：

$$\begin{aligned}&\boldsymbol{B}=(\beta_0,\beta_1,\beta_2,\beta_3)=\tau(\boldsymbol{A})=\\&(\mathrm{Sbox}(\alpha_0),\mathrm{Sbox}(\alpha_1),\mathrm{Sbox}(\alpha_2),\mathrm{Sbox}(\alpha_3))\end{aligned}\tag{4}$$

式中，$\mathrm{Sbox}(\cdot)$为以字节为单位的非线性替换。S 盒的替换表可以参考文献 [18]，将在第 2 章介绍它的代数结构。

变换τ的输出$\boldsymbol{B}$将作为线性变换 L 的输入。设$\boldsymbol{C}\in\mathrm{GF}(2^{32})$是$L$的输出。则线性变换 L 定义为：

$$\begin{aligned}&\boldsymbol{C}=L(\boldsymbol{B})=\boldsymbol{B}\oplus(\boldsymbol{B}<<<2)\oplus(\boldsymbol{B}<<<10)\oplus\\&(\boldsymbol{B}<<<18)\oplus(\boldsymbol{B}<<<24)\end{aligned}\tag{5}$$

式中，$<<<i$表示把 32 位字循环左移i位。

SM4 密码算法的加密过程如图 1 所示。由于解密算法和加密算法采用相同的变换结构和过程，只是反序使用轮密钥，这里不再赘述。

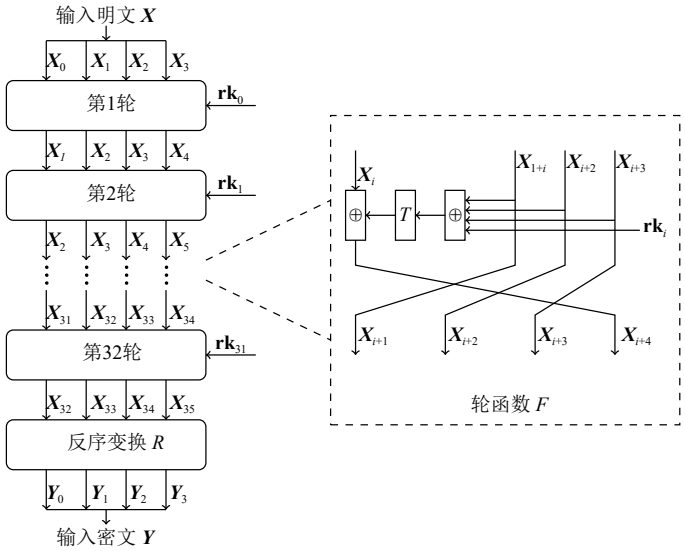

图 1　SM4 分组密码算法加密过程

1.2 密钥扩展算法

轮密钥由 128 bit 密钥通过密钥扩展算法生成。设 $\mathbf{MK}=(\mathbf{MK}_0,\mathbf{MK}_1,\mathbf{MK}_2,\mathbf{MK}_3)\in(\mathrm{GF}(2^{32}))^4$为输入密钥，$\mathbf{rk}_i\in\mathrm{GF}(2^{32})(i=0,1,2,\cdots,31)$为输出的轮密钥，$\boldsymbol{K}_i\in\mathrm{GF}(2^{32})(i=0,1,2,\cdots,35)$为中间数据，密钥扩展算法描述如下：

$$(\boldsymbol{K}_0,\boldsymbol{K}_1,\boldsymbol{K}_2,\boldsymbol{K}_3)=(\mathbf{MK}_0\oplus\mathbf{FK}_0,\mathbf{MK}_1\oplus\mathbf{FK}_1,\mathbf{MK}_2\oplus\mathbf{FK}_2,\mathbf{MK}_3\oplus\mathbf{FK}_3) \tag{6}$$

以及

$$\mathbf{rk}_i=\boldsymbol{K}_{i+4}=\boldsymbol{K}_i\oplus T'(\boldsymbol{K}_{i+1}\oplus\boldsymbol{K}_{i+2}\oplus\boldsymbol{K}_{i+3}\oplus\mathbf{CK}_i) \tag{7}$$

式中，$\mathbf{FK}_i(i=0,1,2,3)$是系统参数；$\mathbf{CK}_i(i=0,1,2,\cdots,31)$是固定参数；$T'(\cdot)$是和加密算法中变换$T(\cdot)$非常类似的变换。

变换T'和变换 T 唯一不同的地方是把 T 中的线性变换 L 替换成如下的线性变换 L'：

$$L'(\boldsymbol{B})=\boldsymbol{B}\oplus(\boldsymbol{B}<<<13)\oplus(\boldsymbol{B}<<<23) \tag{8}$$

系统参数$\mathbf{FK}_i$的十六进制表示如下所示：

$$\begin{aligned}\mathbf{FK}_0&=0\text{xA3B1BAC6}\\ \mathbf{FK}_1&=0\text{x56AA3350}\\ \mathbf{FK}_2&=0\text{x677D9197}\\ \mathbf{FK}_3&=0\text{xB27022DC}\end{aligned} \tag{9}$$

固定参数$\mathbf{CK}_i=(\mathrm{ck}_{i,0},\mathrm{ck}_{i,1},\mathrm{ck}_{i,2},\mathrm{ck}_{i,3})\in(\mathrm{GF}(2^8))^4$采用如下的方式计算：

$$\mathrm{ck}_{i,j}=(4i+j)\times7(\mathrm{mod}\,256) \tag{10}$$

式中，$i=0,1,2,\cdots,31$；$j=0,1,2,3$。

2 S 盒的代数结构

在 SM4 密码算法中，S 盒是唯一的非线性变换，是保证安全的关键组件。S 盒通常由 256 个元素构成的查询表描述。文献 [21] 研究了 S 盒的代数结构，并给出了如下的具体表达式：

$$\mathrm{Sbox}(\boldsymbol{a})=A_2\cdot I\cdot A_1(\boldsymbol{a})\ \ \forall\boldsymbol{a}\in\mathrm{GF}(2^8) \tag{11}$$

式中，I 是$\mathrm{GF}(2^8)$上的乘法逆元，这里的不可约多项式为：

$$f(x)=x^8+x^7+x^6+x^5+x^4+x^2+1 \tag{12}$$

A_1 和 A_2 是如下的仿射变换：

$$A_1(\boldsymbol{a})=A_2(\boldsymbol{a})=\boldsymbol{F}^{\mathrm{T}}\cdot\boldsymbol{a}\oplus\boldsymbol{v} \tag{13}$$

式中，

$$\boldsymbol{F}=\begin{pmatrix}1&1&1&0&0&1&0&1\\1&1&1&1&0&0&1&0\\0&1&1&1&1&0&0&1\\1&0&1&1&1&1&0&0\\0&1&0&1&1&1&1&0\\0&0&1&0&1&1&1&1\\1&0&0&1&0&1&0&1\\1&1&0&0&1&0&1&1\end{pmatrix} \tag{14}$$

是GF(2)上8×8的矩阵；$\boldsymbol{v}=(1,1,0,0,1,0,1,1)^{\mathrm{T}}$是常数列向量。

3 S 盒的量子电路

3.1 实现思路分析

用量子电路实现 S 盒，实际上就是实现式 (11)。在式 (11) 中仿射变换的表达式已经由式 (13) 给出，可以用类似高斯消元的方式实现矩阵 $\boldsymbol{F}^{\mathrm{T}}$，即通过行变换把矩阵 $\boldsymbol{F}^{\mathrm{T}}$ 化成单位阵，每做一次行变换便在对应量子电路中添加一个 CNOT 门，反序排列这些 CNOT 门，便构造出了矩阵 $\boldsymbol{F}^{\mathrm{T}}$ 的量子电路。对于列向量$\boldsymbol{v}$，在出现“1”的对应位置添加 NOT 门便可实现。

乘法逆元 I 会相对复杂。文献 [22] 证明了$\mathrm{GF}(2^n)$上的任一非零元素$\boldsymbol{a}$的逆元可以表示成$(\boldsymbol{a})^{-1}=(\boldsymbol{a})^{2^n-2}$。因此需要计算：

$$I(\boldsymbol{a})=\boldsymbol{a}^{254} \tag{15}$$

为了快速计算式 (15)，需要分别实现$\mathrm{GF}(2)[x]/(f(x))$上的平方计算和乘法计算。$\mathrm{GF}(2)[x]$上的平方计算具有线性的性质，即对元素$\boldsymbol{a}=\sum_{i=0}^{n-1}a_ix^i$有：

$$\boldsymbol{a}^2=\left(\sum_{i=0}^{n-1}a_ix^i\right)^2=\sum_{i=0}^{n-1}a_ix^{2i} \tag{16}$$

因此，可以得出：$\forall\boldsymbol{a}\in\mathrm{GF}(2)[x]/(f(x))$，有$\boldsymbol{a}^2=\mathbf{Sq}\cdot\boldsymbol{a}$，其中：

$$\mathbf{Sq}=\begin{pmatrix}1&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&1\\0&1&0&0&1&1&0&0\\0&0&0&0&0&1&1&1\\0&0&1&0&1&1&1&0\\0&0&0&0&1&1&1&0\\0&0&0&1&1&0&1&0\\0&0&0&0&1&0&1&0\end{pmatrix} \tag{17}$$

对于乘法计算，文献 [23] 得出如下结论：$\forall \boldsymbol{a},\boldsymbol{b}\in \mathrm{GF}(2)[x]/(f(x))$，记$\boldsymbol{c}=\boldsymbol{a}\cdot\boldsymbol{b}$，有：

$$\boldsymbol{c}=\boldsymbol{d}+\boldsymbol{Q}^{\mathrm{T}}\cdot\boldsymbol{e} \tag{18}$$

其中，

$$\boldsymbol{d}=\begin{pmatrix}d_0\\d_1\\\vdots\\d_{n-1}\end{pmatrix}=\boldsymbol{L}\cdot\boldsymbol{b}=\begin{pmatrix}a_0&0&\cdots&0\\a_1&a_0&\cdots&0\\\vdots&\vdots&\ddots&\vdots\\a_{n-1}&a_{n-2}&\cdots&a_0\end{pmatrix}\cdot\begin{pmatrix}b_0\\b_1\\\vdots\\b_{n-1}\end{pmatrix} \tag{19}$$

$$\boldsymbol{e}=\begin{pmatrix}e_0\\e_1\\\vdots\\e_{n-1}\end{pmatrix}=\boldsymbol{U}\cdot\boldsymbol{b}=\begin{pmatrix}0&a_{n-1}&a_{n-2}&\cdots&a_1\\0&0&a_{n-1}&\cdots&a_2\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&a_{n-1}\end{pmatrix}\cdot\begin{pmatrix}b_0\\b_1\\\vdots\\b_{n-1}\end{pmatrix} \tag{20}$$

式中，a_i是$\boldsymbol{a}$对应位置上的元素；$\boldsymbol{Q}^{\mathrm{T}}$表示矩阵$\boldsymbol{Q}$的转置，其中$\boldsymbol{Q}$是使得

$$\begin{pmatrix}x^n\\x^{n+1}\\\vdots\\x^{2n-2}\end{pmatrix}\equiv\boldsymbol{Q}\cdot\begin{pmatrix}1\\x\\\vdots\\x^{n-1}\end{pmatrix}(\mathrm{mod}\ f(x)) \tag{21}$$

成立的二元矩阵，于是可以计算出：

$$\boldsymbol{Q}^{\mathrm{T}}=\begin{pmatrix}1&1&0&0&0&1&0\\0&1&1&0&0&0&1\\1&1&1&1&0&1&0\\0&1&1&1&1&0&1\\1&1&1&1&1&0&0\\1&0&1&1&1&0&0\\1&0&0&1&1&0&0\\1&0&0&0&1&0&0\end{pmatrix} \tag{22}$$

3.2　量子电路实现

本文使用 Python 语言并利用 qiskit 软件包实现所求的量子电路。对于式 (13) 中仿射变换的表达式，使用高斯消元法实现矩阵 $\boldsymbol{F}^{\mathrm{T}}$，并通过在对应位置添加 NOT 的方法实现向量$\boldsymbol{v}$，最终式 (13) 中仿射变换的量子电路如图 2 所示。GF(2^8)上的平方计算可以通过式 (17) 的线性变换表示，利用高斯消元法实现平方计算的量子电路如图 3 所示。式 (19) 和式 (20) 中的$\boldsymbol{d}$和$\boldsymbol{e}$可以通过 Toffoli 门实现，式 (22) 中的矩阵$\boldsymbol{Q}^{\mathrm{T}}$可以通过高斯消元法实现，因此式 (18) 便可以实现，量子电路如图 4 所示。

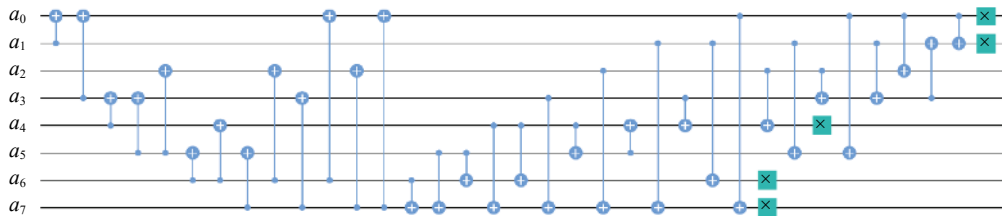

图 2　实现式 (13) 仿射变换的量子电路图

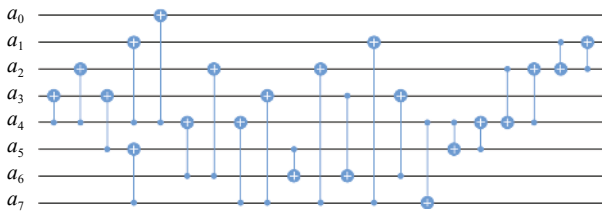

图 3　实现式 (17) 平方计算的量子电路图

为了更快速地计算$I(\boldsymbol{a})=\boldsymbol{a}^{254}$，本文采用 Itoh-Tsujii 算法[24]，但为了尽可能少地使用辅助量子比特，把计算顺序调整如下：

1) $(\boldsymbol{a})^2=\boldsymbol{a}^2$

2) $\boldsymbol{a}\cdot\boldsymbol{a}^2=\boldsymbol{a}^3$

3) $(\boldsymbol{a}^3)^{2^2}=\boldsymbol{a}^{12}$

4) $\boldsymbol{a}^2\cdot\boldsymbol{a}^{12}=\boldsymbol{a}^{14}$

5) $\boldsymbol{a}\cdot\boldsymbol{a}^{14}=\boldsymbol{a}^{15}$

6) $(\boldsymbol{a}^{15})^{2^4}=\boldsymbol{a}^{240}$

7) $\boldsymbol{a}^{14}\cdot\boldsymbol{a}^{240}=\boldsymbol{a}^{254}$

由于仿射变换和平方计算的量子电路都为 8 量子比特，乘法计算量子电路为 24 量子比特，为了更加方便地表示 S 盒的量子电路，本文以 8 量子比特为单位描述 S 盒的量子电路。记 A_1 和 A_2 为图 2 中实现仿射变换的量子电路；Sq 为图 3 中实现平

方计算的量子电路；图 4 中实现乘法计算量子电路的乘数输入分别记为两个实心圆点，乘积结果记为 M。主要根据计算$\boldsymbol{a}^{254}$的步骤，可以得出 S 盒的量子电路如图 5所示。

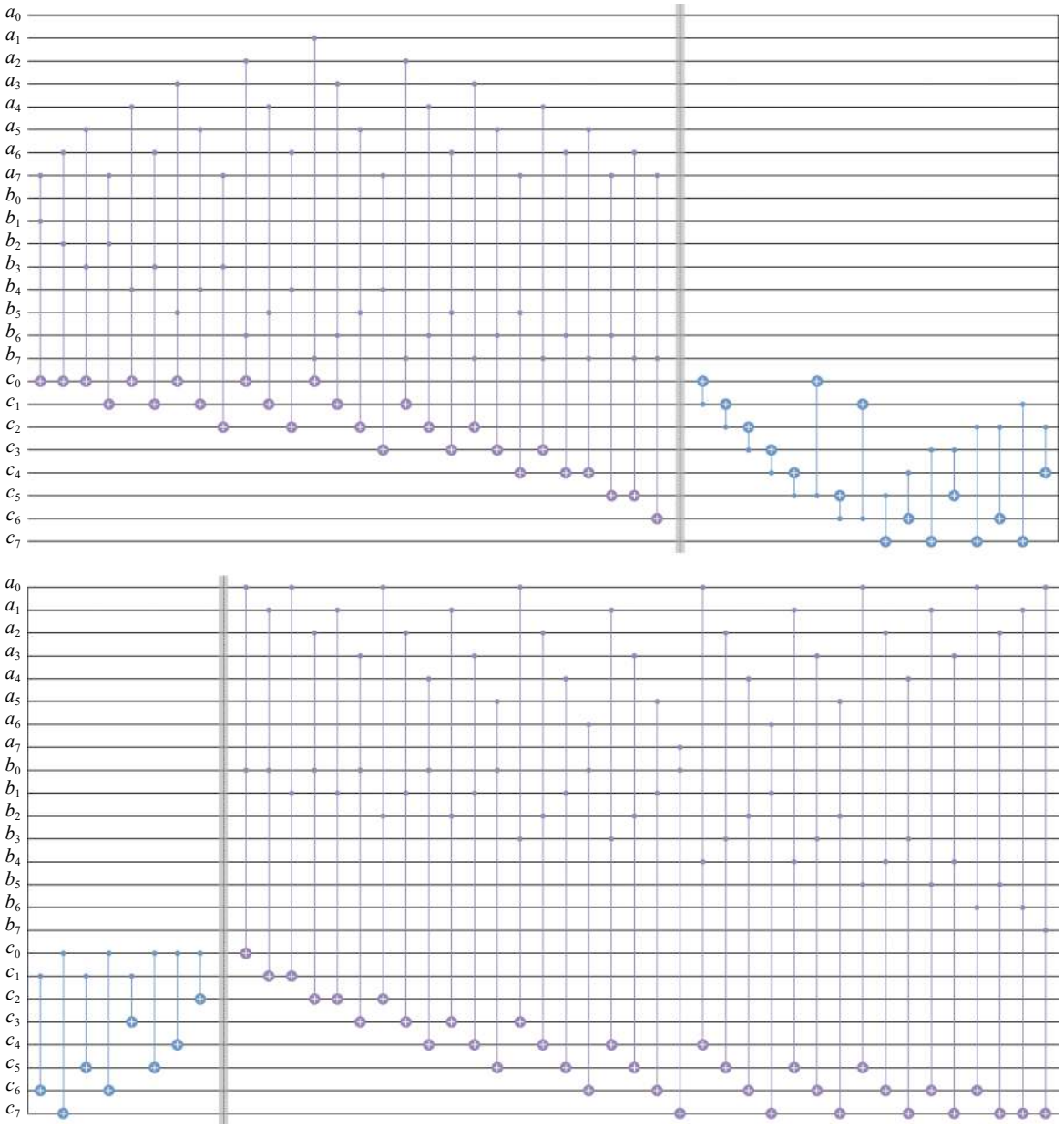

图 4 实现式 (18) 乘法计算的量子电路图

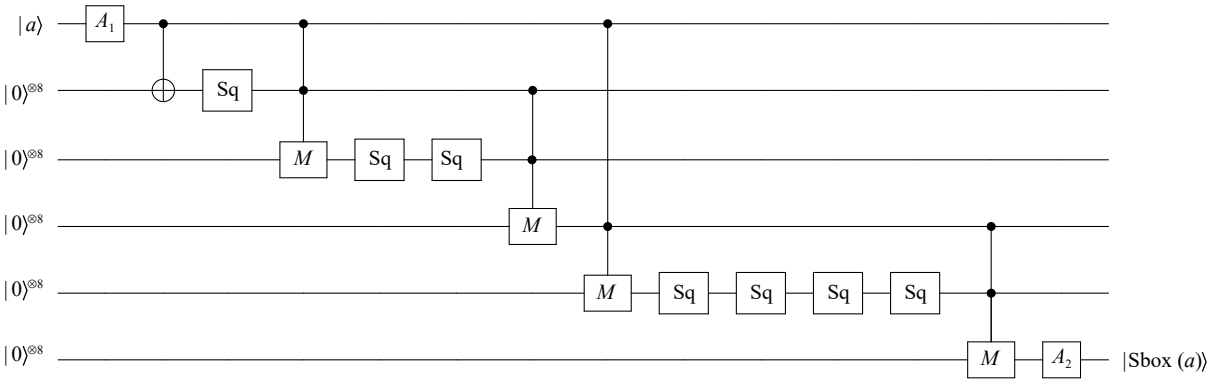

图 5 S 盒的量子电路示意图

3.3　量子电路复杂度分析

这里的量子电路是利用 NCT 门库实现，即采用 NOT 门、CNOT 门和 Toffoli 门实现。通过这些量子电路所用量子比特的数量、量子门的数量和量子电路的深度描述构建的量子电路的复杂度如表 1 所示。

表 1　量子电路所用量子资源情况表

变换方式	量子比特数	量子门数	电路深度
仿射变换	8	39	23
平方计算	8	22	16
乘法计算	24	88	39
S盒	48	592	289

由图 5 可知，最终 S 盒的实现电路中需要 2 次调用仿射变换的量子电路，每个仿射变换的量子电路由 34 个 CNOT 门和 5 个 NOT 门构成，电路深度为 23；需要 7 次调用平方计算的量子电路，每个平方计算的量子电路由 22 个 CNOT 门构成，电路深度为 16；需要 4 次调用乘法计算的量子电路，每个乘法计算的量子电路由 64 个 Toffoli 门和 24 个 CNOT 门构成，电路深度为 39；此外还需要用 1 组 (8 个)CNOT 门对量子比特进行复制。整个 S 盒的量子电路中一共使用了 48 个量子比特，592 个量子门，电路深度为 289。由于本文是首次实现 SM4 密码算法 S 盒的量子电路，不能通过对比的方式来分析该电路的复杂度。但文献 [12] 实现了 AES 密码算法 S 盒的量子电路，在该量子电路中，共使用了 3 组 (24 个)CNOT 门对量子比特进行复制，使用的量子比特数为 56。由此可以看出，本文实现的 SM4 算法 S 盒的量子电路还是比较高效的。

4　结 束 语

本文首次给出了 SM4 密码算法 S 盒的量子电路。根据 S 盒的代数结构，首先给出 S 盒代数表达式中仿射变换的量子电路，然后把代数表达式中求 $GF(2^8)$下元素的逆元转换为求该元素的 254 次方，为此，分别构建了$GF(2^8)$中平方计算的量子电路和乘法计算的量子电路，再通过改进的 Itoh-Tsujii 算法给出 S 盒的量子电路。所构建的 S 盒的量子电路共使用了 48 个量子比特，592 个量子门，电路深度为 289。本文的研究将对量子环境下 SM4 密码算法的研究产生积极影响。

参　考　文　献

[1] PETER W. Shor polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer[J]. SIAM J Comput, 1997, 26(5): 1484-1509.

[2] RIVEST R L, SHAMIR A, ADLEMAN L. A method for obtaining digital signatures and public-key cryptosystems[J]. Commun ACM, 1978, 21(2): 120-126.

[3] ELGAMAL T. A public-key cryptosystem and a signature scheme based on discrete logarithms[J]. IEEE Trans Inf Theory, 1985, 31(4): 469-472.

[4] YAMAMURA A, ISHIZUKA H. Quantum cryptanalysis of block ciphers (Algebraic systems, formal languages and computations)[J]. RIMS Kokyuroku, 2000(1166): 235-243.

[5] GROVER L K. A fast quantum mechanical algorithm for database search[C]//Proc of the 28th Annual ACM Symposium on Theory of Computing (STOC). [S.l.]: ACM, 1996: 212-219.

[6] KAPLAN M, LEURENT G, LEVERRIER A, et al. Breaking symmetric cryptosystems using quantum period finding[C]//Annual International Cryptology Conference. Berlin, Heidelberg: Springer, 2016: 207-237.

[7] SIMON D. On the power of quantum computation[C]// Proceedings of the 35th IEEE Symposium on the Foundations of Computer Science (FOCS). [S.l.]: IEEE, 1994: 116-123.

[8] LEANDER G, MAY A. Grover meets Simon-quantumly attacking the FX-construction[C]//International Conference on the Theory and Application of Cryptology and Information Security. Cham: Springer, 2017: 161-178.

[9] DONG X Y, WANG X Y. Quantum key-recovery attack on feistel structures[J]. Science China Information Sciences, 2018, 61(10): 1-7.

[10] BONNETAIN X, NAYA-PLASENCIA M, SCHROTTENLOHER A. Quantum security analysis of AES[J]. IACR Transactions on Symmetric Cryptology, 2019(2): 55-93.

[11] SAMIR H, LARS R K. A quantum distinguisher for 7/8-round SMS4 block cipher[J]. Quantum Information Processing, 2020, 19(11): 1-22.

[12] ALMAZROOIE M, SAMSUDIN A, ABDULLAH R, et al. Quantum reversible circuit of AES-128[J]. Quantum Information Processing, 2018, 17(5): 1-30.

[13] ZOU J, WEI Z, SUN S, et al. Quantum circuit implementations of AES with fewer qubits[C]// International Conference on the Theory and Application of Cryptology and Information Security. Cham: Springer, 2020: 697-726.

[14] 国家密码管理局. 国家密码管理局关于发布无线局域网产品密码事宜公告[EB/OL]. (2006-01-06). https//sca.gov.cn/sca/xwdt/2006-01/06/content_1002355.shtml.
State Cryptography Administration. Announcement of the State Cryptography Administration on issuing ciphers for WLAN products[EB/OL]. (2006-01-06). https//sca.gov.cn/sca/xwdt/2006-01/06/content_1002355.shtml.

[15] 国家密码管理局. GM/T 0002-2012 SM4 分组密码算法[S]. 北京: 中国标准出版社, 2012.
State Cryptography Administration. GM/T 0002-2012

SM4 block cipher algorithm[S]. Beijing: Standards Press of China, 2012.

[16] 中国标准化委员会. GB/T 32907-2016 信息安全技术 SM4 分组密码算法[S]. 北京: 中国质检出版社, 2016.
China Standardization Commission. GB/T 32907-2016 information security technology--SM4 block cipher algorithm[S]. Beijing: China Quality Inspection Press, 2016.

[17] 国家密码管理局. 我国 SM4 分组密码算法正式成为 ISO/IEC 国际标准[EB/OL]. [2021-07-08]. https//www.oscca.gov.cn/sca/xwdt/2021-07/08/content_1060866.shtml.
State Cryptography Administration. China's SM4 block cipher algorithm has officially become an ISO / IEC international standard[EB/OL]. [2021-07-08]. https//www.oscca.gov.cn/sca/xwdt/2021-07/08/content_1060866.shtml.

[18] 吕述望, 苏波展, 王鹏, 等. SM4 分组密码算法综述[J]. 信息安全研究, 2016, 2(11): 995-1007.
LYU S W, SU B Z, WANG P, et al. Overview on SM4 algorithm[J]. Journal of Information Security Research, 2016, 2(11): 995-1007.

[19] LIANG H, WU L, ZHANG X, et al. Design of a masked S-Box for SM4 based on composite field[C]//2014 Tenth International Conference on Computational Intelligence and Security. Kunming: IEEE, 2014: 387-391.

[20] 李新超, 钟卫东, 张帅伟, 等. 一种 SM4 算法 S 盒的门限实现方案[J]. 密码学报, 2018, 5(6): 641-650.
LI X C, ZHONG W D, ZHANG S W, et al. A new threshold implementation of the S-box in SM4[J]. Journal of Cryptologic Research, 2018, 5(6): 641-650

[21] LIU F, JI W, HU L, et al. Analysis of the SMS4 block cipher[C]//Information Security and Privacy. Berlin, Heidelberg: Springer, 2007: 158-170.

[22] WANG C C, TRUONG T K, SHAO H M, et al. VLSI architectures for computing multiplications and inverses in GF(2^m)[J]. IEEE Transactions on Computers, 1985, 100(8): 709-717.

[23] REYHANI-MASOLEH A, HASAN M A. Low complexity bit parallel architectures for polynomial basis multiplication over GF(2^m)[J]. IEEE Transactions on Computers, 2004, 53(8): 945-959.

[24] ITOH T, TSUJII S. A fast algorithm for computing multiplicative inverses in GF (2^m) using normal bases[J]. Information and Computation, 1988, 78(3): 171-177.

编 辑 蒋 晓