认证加密算法 SM4-GCM 的低成本硬件架构设计与实现

陈锐，李春强
（南京工业职业技术大学计算机与软件学院，江苏 南京 210021）

摘 要：物联网已被广泛应用于各行各业，该项技术的赋能使得行业应用更好地向数字化、智能化方向发展。在一些行业应用中，物联网设备采集的数据与用户隐私和财产安全关系密切。为了保护数据安全，基于国产认证加密算法 SM4-GCM（Galois/Counter Mode），提出一种低成本、多功能的硬件架构设计。设计兼顾性能、成本和硬件级的数据机密性和完整性保障，同时也支持 3 种工作模式：SM4-CTR、SM4-ECB 和 SM4-GCM。在现场可编程门阵列（FPGA, field programmable gate array）开发板上的实现结果显示，该设计仅需 1 761 个查找表和 1 825 个寄存器，占用的资源片仅为 604，而吞吐率达到 39.78 Mbit/s@100 MHz，能够满足物联网数据安全应用需求。

关键词：物联网；数据安全；认证加密；SM4-GCM；FPGA
中图分类号：TN918.4
文献标志码：A
doi: 10.11959/j.issn.2096−3750.2023.00362

Design and implementation of low-cost hardware architecture for authentication encryption algorithm SM4-GCM

CHEN Rui, LI Chunqiang
School of Computer and Software Engineering, Nanjing Vocational University of Industry Technology, Nanjing 210021, China

Abstract: The internet of things (IoT) has gained wide adoption across various industries, driving digitalization and intelligence in industry applications. However, the data collected by IoT devices in some industry applications may be closely linked to user privacy and property security. To ensure the security of such data, a cost-effective, multifunctional hardware architecture design based on the Chinese authenticated encryption algorithm SM4-GCM (Galois/Counter Mode) was proposed, which offered a balanced approach to performance, cost, and hardware-level data confidentiality and integrity assurance, and supported three operation modes: SM4-CTR, SM4-ECB, and SM4-GCM. The implementation results on the field programmable gate array (FPGA) development board demonstrate that the design requires only 1 761 look-up tables and 1 825 registers, occupies only 604 slices, and has a throughput rate of 39.78 Mbit/s@100 MHz. These results suggest that the proposed design can effectively meet the requirements of IoT data security applications.

Key words: IoT, data security, authenticated encryption, SM4-GCM, FPGA

0 引言

物联网之所以被广泛应用，主要是因为其采集的数据隐含的价值[1]。在智慧医疗[2-3]领域，通过采集心跳等生物信号，可以实现身体健康状态监控；在智能家居[4-6]领域，通过采集温度、光照等信息，

收稿日期：2023−04−07；**修回日期**：2023−07−12
通信作者：陈锐，chenrui@niit.edu.cn
基金项目：江苏省工业软件工程技术中心开放基金资助项目（No.ZK19-04-03）；南京工业职业技术大学创新基金资助项目（No.YK20-05-07）
Foundation Items: Open Fund of Jiangsu Industrial Software Engineering Technology Center (No.ZK19-04-03), Innovation Fund of Nanjing Vocational University of Industry Technology (No.YK20-05-07)

实现家电智能控制；在智能工厂[7]领域，通过采集设备状态信息，可以实现预测性维护。在这些典型应用中，物联网设备采集的数据与用户隐私、设备健康、财产安全关系密切，若被窃取，则会造成用户隐私泄露、设备损毁、财产损失等问题。近年来，网络连接设备日益增多，使得物联网引发的网络安全问题数量呈上升趋势[8]。随着越来越多的网络连接设备、系统和服务集成到国家关键基础设施，物联网引发的网络安全问题可能会引发其他方面更严重的安全问题[9]。

隐含在数据背后的价值是吸引攻击者非法攻击、窃取物联网采集数据的主要原因。若能在物联网设备源头为数据提供机密性、完整性保障，则能避免数据传输过程中可能发生的数据窃取、丢失和篡改等情况。认证加密算法能够同时提供数据机密性和完整性保障。将认证加密算法部署在物联网设备端，可从源头解决数据安全问题。

SM4 伽罗华/计数器模式（SM4-GCM, SM4 based Galois/Counter Mode）是蚂蚁集团在 IETF RFC8998[10]发布的认证加密算法。密码算法的实现方式主要有 3 种：软件实现、软硬协同实现、硬件实现。与软件实现和软硬协同实现方式相比，硬件实现具有以下 3 种特性使其更适用于物联网这类资源受限场景。首先，硬件实现性能高功耗低；其次，硬件实现不依赖 CPU 和指令集，不受二者漏洞影响；第三，硬件实现攻击难度大、安全性高。

上述需求和特性促使本文面向物联网数据安全应用场景，研究、设计和实现 SM4-GCM 算法的低成本硬件架构。

1　算法背景

SM4-GCM 是我国分组密码算法标准 SM4 与 GCM 的结合，其包含 SM4 计数器（SM4-CTR, SM4 based Counter）模式和伽罗华哈希（GHASH, Galois Hash）模式，SM4-GCM 算法示意图如图 1 所示，前者提供机密性保障，后者提供完整性保障。

SM4-CTR 是指工作在计数器模式下的 SM4 算法，表达式如式(1)所示，其中，PT_i 为明文，CT_i 为加密结果，Cnt_i 为计数值，Key 为密钥，⊕ 为异或运算，SM4 为加密运算，数据位宽均为 128 bit。在 SM4-CTR 模式下，算法运算结果为计数值加密之后再与明文异或的结果。

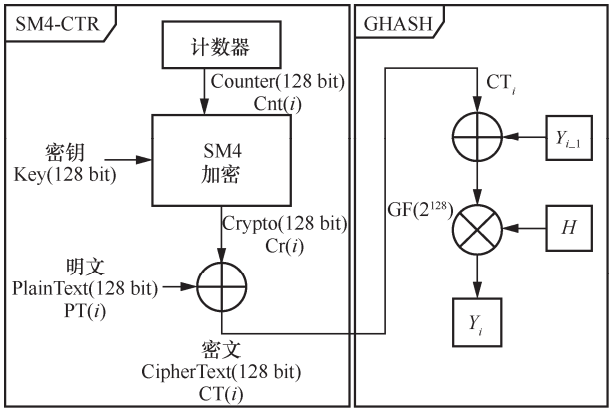


图 1　SM4-GCM 算法示意图

$$CT_i = PT_i \oplus SM4\left(Cnt_i, Key\right) \tag{1}$$

GHASH 算法用于计算辅助认证数据（AAD, additional authenticated data）、密文和认证辅助数据的长度以及密文长度的哈希值，其表达式如式(2)所示，其中，Y_{i-1} 为前一个数据块运算结果，Y_i 为当前数据块运算结果，H 为 128 bit 0 的 SM4 加密结果，⊗ 为伽罗华域乘法运算 GF(2^{128})。输入数据 Data 与前一个数据块的 GHASH 运算结果异或之后再进行 GF(2^{128})乘法运算，经过多次迭代，形成一个 128 bit 的哈希值。式(2)中的 Data 的数据类型依次为 AAD 数据块、密文数据块、长度数据块（64 bit AAD 的长度与密文长度拼接而成）。

$$Y_i = \left(Data \oplus Y_{i-1}\right) \otimes H \tag{2}$$

假设长度数据块的 GHASH 计算结果为 Y_n，那么最终输出的认证码为

$$MAC = Y_n \oplus SM4\left(IV \| 0^{31} \| 1, Key\right) \tag{3}$$

其中，“||”为拼接运算符，IV 为 12 byte 的初始化向量。

需要注意的是，RFC8998 发布的 SM4-GCM 算法中使用的 GCM 是原始 GCM 的子集，IV 长度固定为 12 byte，输出认证码长度固定为 16 byte，因而相对于原始算法复杂度有所降低。

2　相关工作

与本文相关的研究工作可概括为两大类：修改标准 SM4 算法以进一步提升其安全性，如文献[11-12]提出的 SM4 Sbox 白盒实现方法，文献[13-14]提出的 SM4 Sbox 门限实现方法，文献[15]提出抗旁路攻击的实现方法，文献[16-17]提出抗功耗分析攻击的实现方法；标准 SM4 算法的实现或 SM4 算法结合

某一种分组密码算法工作模式的实现，这一类也是本文的关注点。

标准 SM4 算法的实现见表 1。文献[18-20]采用软件实现 SM4 算法，其中，文献[19]对比和优化了 SM4 算法在 4 款低端处理器上的性能表现，文献[20]采用 Bit-Slice 技术提升软件实现性能，文献[18]则是 SM4 算法的单指令流多数据流（SIMD, single-instruction multiple-data）软件加速实现方案。软件实现方案依赖 CPU 或指令集，若二者存在漏洞，则密码算法本身的安全性得不到保障。文献[21-29]采用硬件实现 SM4 算法或 SM4 算法与某种工作模式的结合，其中，文献[21-23]与本文研究内容最相似，均实现了 SM4-GCM 算法，但与本文不同的是，三者均面向高性能计算领域。与本文相似，文献[24]面向物联网领域实现了基于 SM4 算法的计数器模式和密码分组链接消息验证码（CCM, counter with cipher block chaining message authentication code（CBC-MAC））。考虑物联网设备可能部署在无人区域，网络环境不稳定，可能存在数据丢包的现象，在这种场景下，SM4-GCM 算法明显更灵活也更合适，因为 SM4-GCM 算法对每个数据块的处理是相互独立的。文献[27-30]仅实现了 SM4 算法，但实际应用中需要与具体的分组密码工作模式结合[31]才能为数据提供更高的安全性保障。

表 1 标准 SM4 算法的实现

文献	算法	实现	领域
[18]	SM4	软件	低成本
[19]	SM4	软件	低成本
[20]	SM4	软件	高性能
[21]	SM4-GCM	GPU	高性能
[22]	SM4-GCM	FPGA	高性能
[23]	SM4-GCM	FPGA	高性能
[24]	SM4-CCM	ASIC	低成本
[25]	SM4-XTS	FPGA	高性能
[26]	SM4-CBC	ASIC	高性能
[27]	SM4	FPGA	高性能
[28]	SM4	ASIC	低成本
[29]	SM4	ASIC	高性能
[30]	SM4	SW/HW	低成本

基于上述分析，笔者发现现有的研究缺少面向物联网资源受限、网络环境不稳定场景，成本低并且能够同时提供机密性和完整性保障的认证加密算法硬件实现方案。针对这一发现，面向物联网应用领域，本文提出一种低成本的认证加密算法 SM4-GCM 硬件架构设计。

3 低成本硬件架构设计

3.1 整体架构设计

低成本硬件架构设计方案如图 2 所示，可以看出，本文提出的架构包含总线接口、SM4 和 GHASH 共 3 个功能子模块，多个寄存器，以及多个数据通路选择器。整个架构对外提供 AXI-lite 总线读写接口。采用此接口封装 SM4-GCM 算法知识产权（IP, intellectual property）核，其目的主要是提高本文设计的可重用性。IP 核中的寄存器，如控制寄存器、数据寄存器、密钥寄存器、IV 寄存器、状态寄存器、H 寄存器均为存储映射寄存器，即上位机通过读写地址，写入配置即可控制 IP 的功能，读写寄存器数据。

采用 SM4-GCM 算法进行认证加密时，需要按顺序依次处理 5 种类型的数据，分别为 IV、密钥、AAD、明文以及 AAD/明文的长度。类型不同，数据的处理方式不同。为了更清晰地展示不同类型的数据在图 2 所示架构上的处理方式，不同类型数据的处理及数据通路如图 3 所示。

3.1.1 IV 数据处理

首先处理的数据是 IV，如图 3(a)所示，上位机将 12 byte 的 IV 末尾填充 32 byte 0 之后，经过地址译码之后，通过 AXI-lite 总线接口写入 IV 寄存器。

3.1.2 密钥数据处理

其次处理的数据是密钥，如图 3(b)所示，经过地址译码之后，通过 AXI-lite 总线接口将密钥写入密钥寄存器，然后启动 128 bit 0 的加密计算，待加密结束之后，将加密结果写入 H 寄存器。

3.1.3 AAD 数据处理

接下来处理的数据块是 AAD，如图 3(c)和图 3(d)所示。第一个 AAD 数据块的处理方式如图 3(c)所示，经过地址译码之后，通过 AXI-lite 总线接口将 AAD 数据块写入数据寄存器中，然后将数据寄存器与 128 bit 0 异或之后，再与 H 寄存器中的数据进行伽罗华域模乘，结果保存在乘法器内部寄存器之中。后续的 AAD 数据处理方式如图 3(d)所示，数据寄存器与乘法器内部寄存器异或之后，再与 H 寄存器中的数据进行伽罗华域模乘，结果继续保

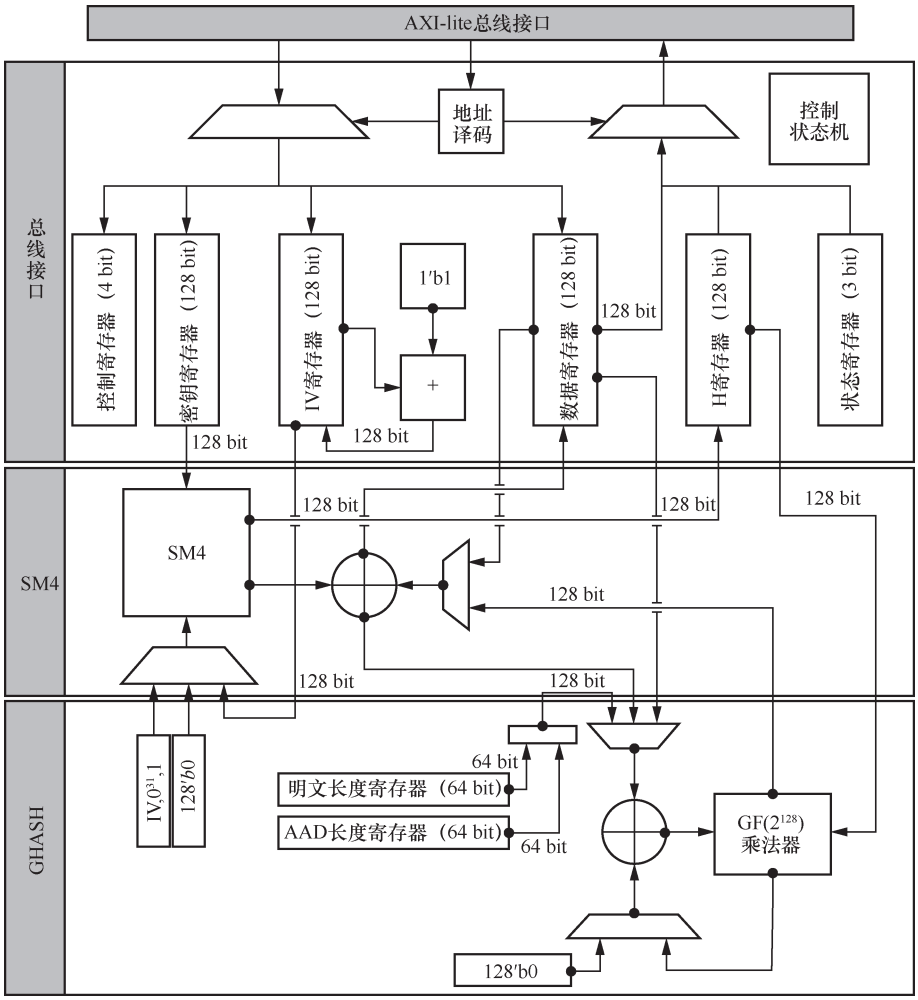
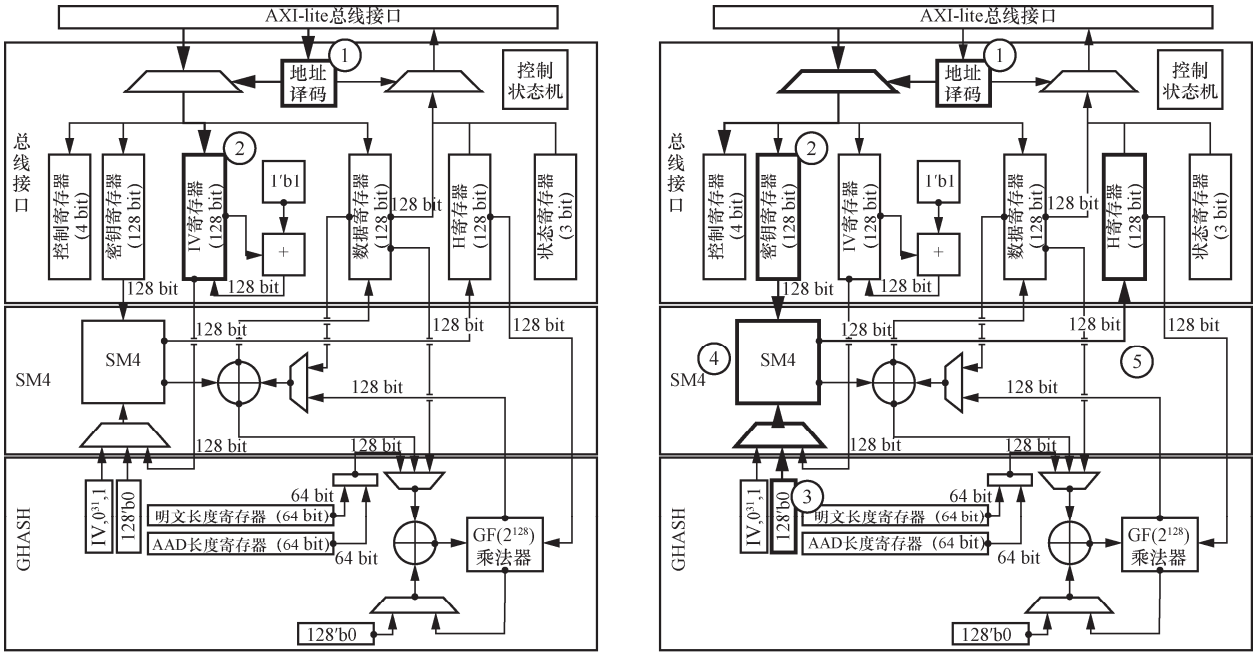
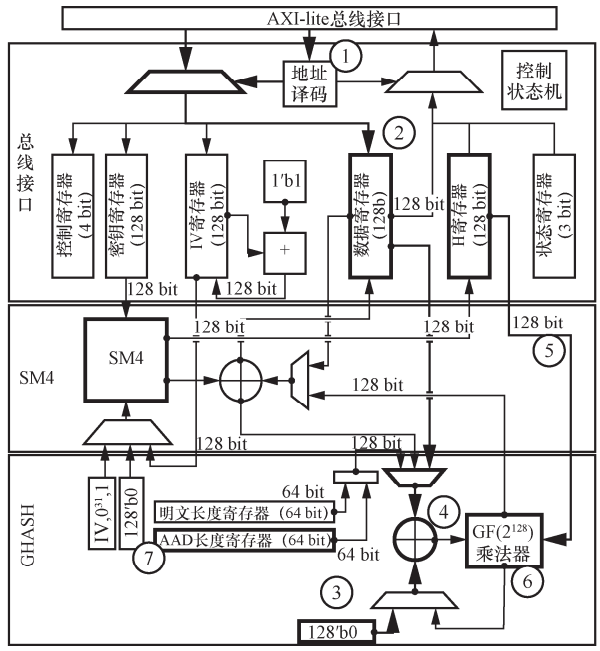


图 2　低成本硬件架构设计方案

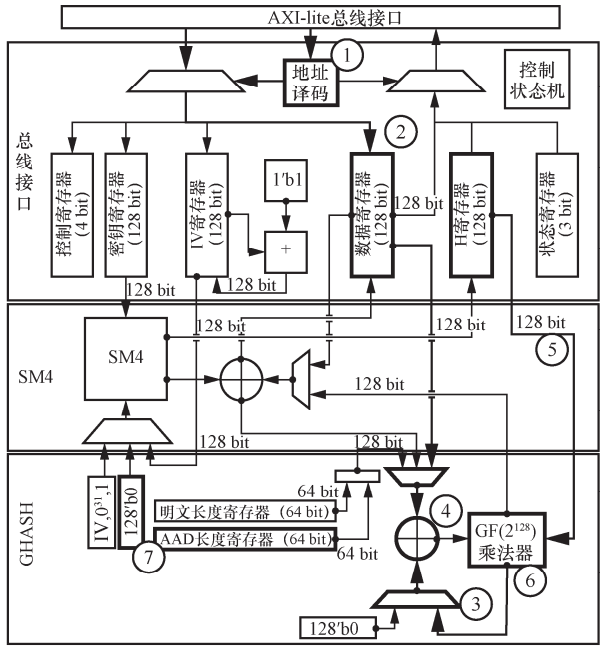


（a）IV数据的处理　　（b）密钥数据的处理

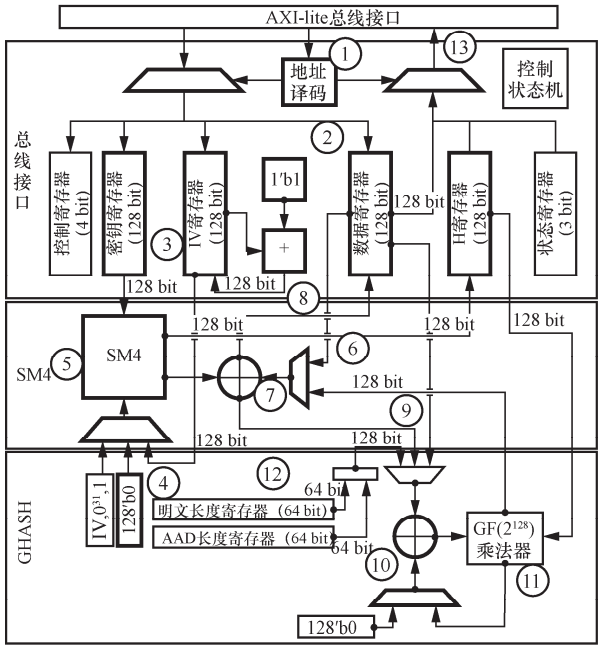
图 3　不同类型数据的处理及数据通路



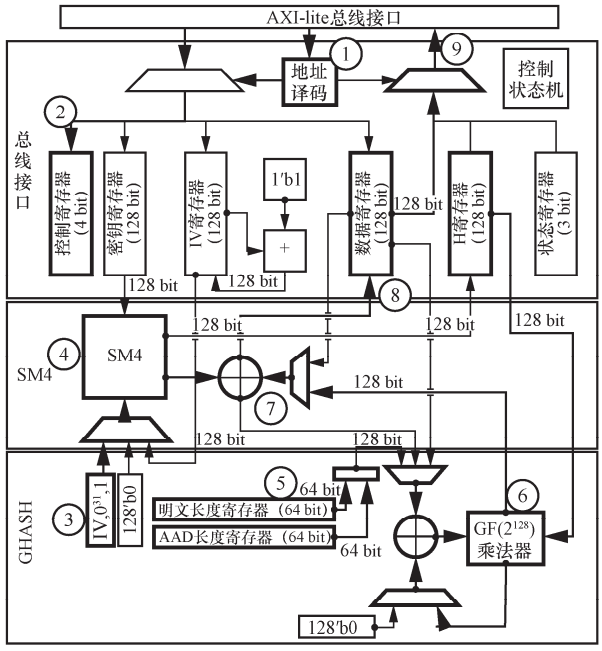
(c) 第一个AAD数据块的处理



(d) 第N个AAD数据块的处理



(e) 明文数据的处理



(f) 长度数据处理及认证码的生成

图 3　不同类型数据的处理及数据通路（续）

存在乘法器内部寄存器之中。同时更新 AAD 长度寄存器，记录输入的 AAD 数据的长度。

3.1.4　明文数据处理

之后处理明文数据块，如图 3(e)所示。经过地址译码后，通过 AXI-lite 总线接口将明文数据块写入数据寄存器，然后将数据寄存器与 IV 寄存器值的加密结果（序号③～⑤）异或，将其结果作为密文写回数据寄存器（序号⑦～⑧）。密文与伽罗华域乘法器内部寄存器异或之后，再与 H 寄存器中的数据进行伽罗华域模乘，结果继续保存在乘法器内部寄存器之中（序号⑨～⑪）。之后更新明文长度寄存器，记录已经处理的明文数据长度，同时也更新状态寄存器，通知上位机加密计算完毕，可以通过总线访问数据寄存器读取加密结果。

3.1.5　长度数据处理及认证码生成

最后处理的是长度数据，如图 3(f)所示，先将

明文/AAD 长度数据拼接为 128 bit 数据，然后与伽罗华域乘法器内部寄存器异或，之后与 H 寄存器中的数据进行模乘（序号⑤～⑥），其结果与 IV 初始值加密结果异或之后作为最终的认证码写回数据寄存器（序号③～④、⑦～⑧）。最后，更新状态寄存器，通知上位机认证码计算完毕，可以通过总线访问数据寄存器读取 128 bit 的认证码（序号⑨）。

为了节约设计成本，平衡性能和面积，本文设计采用了共享和分时复用两种技术手段。

3.2　SM4 算法架构设计

SM4 算法由加解密算法和密钥扩展算法组成。由于 SM4-GCM 算法不需要 SM4 算法解密，因此无须考虑 SM4 解密算法。密钥扩展可采用离线计算或在线计算的方式实现，一般而言，在线计算方法无须保存扩展密钥，因此所需的资源较低。本文所设计的 SM4 算法硬件架构采用在线密钥扩展方法。

SM4 算法硬件架构如图 4 所示，其中，D0~D3 为数据寄存器，K0~K3 为密钥寄存器，CK_i 为标准 SM4 算法规定的常数。为了节省资源开销，SM4 加密算法和密钥扩展算法共享数据通路，以交替时分复用的方式共享轮函数。在轮函数内部，Sbox 用于实现非线性变换（字节替换）。由于 SM4 算法的 Sbox 是由 256 个 8 byte 的常数组成，占用资源较多，因此仅使用了单个 Sbox。32 bit 数据的非线性变换共享单个 Sbox，4 byte 依次送入 Sbox，待字节替换完成再拼接回 32 bit。

SM4 算法内核的状态转换示意图如图 5 所示，图 5 中序号为状态跳转的顺序。当收到新数据计算请求时，启动状态机，由空闲状态跳转至密钥扩展状态（序号①）。由于每一轮的 SM4 加密算法依赖一个 32 bit 的扩展密钥，因此在每一轮的迭代中，密钥扩展算法先于加密算法调用轮函数（序号②）。每一轮加密算法计算完毕之后返回至密钥扩展进行下一个扩展密钥的计算（序号③）。待 32 轮迭代全部完成，执行反序变换输出加密结果（序号④），状态机恢复空闲状态（序号⑤），等待下一个数据块的输入。

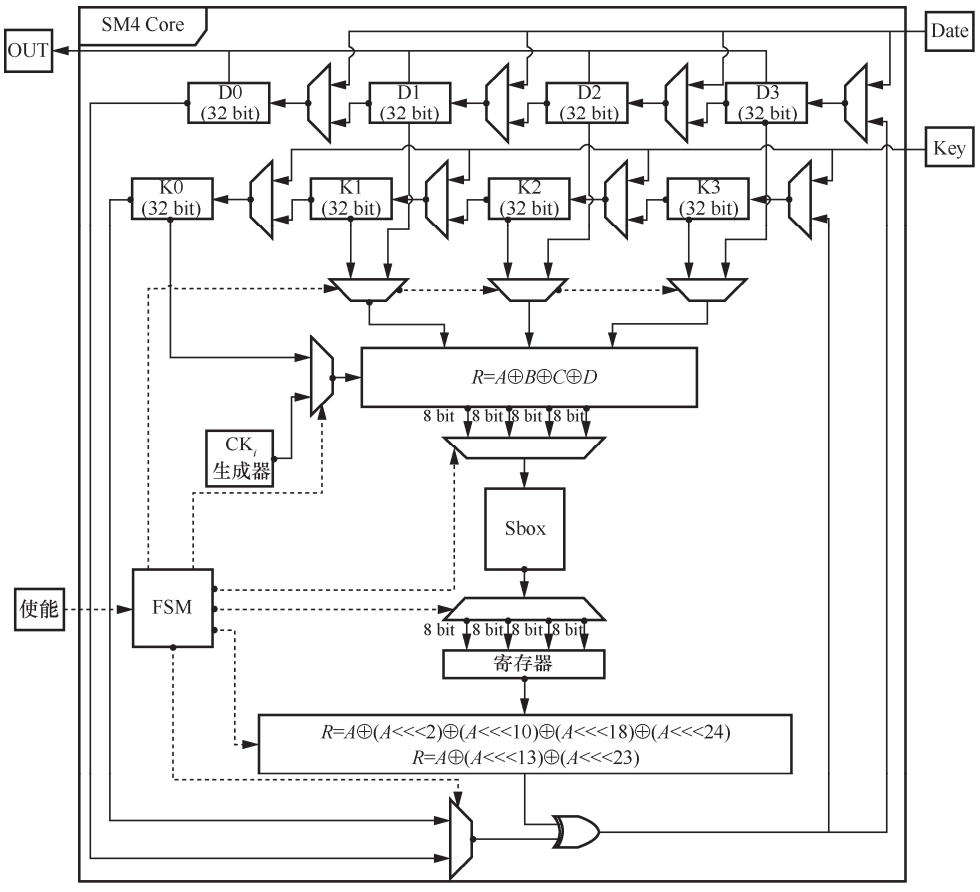


图 4　SM4 算法硬件架构

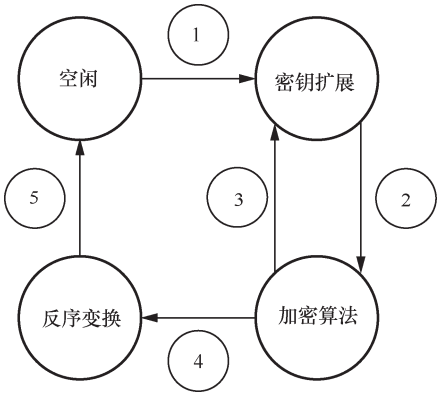


图 5　SM4 算法内核的状态转换示意图

3.3　GF(2^{128})迭代乘法算法硬件架构设计

GF(2^{128})迭代乘法算法是 GHASH 算法的核心组成部分，可采用循环迭代以异或和移位运算实现，也可结合 Karatsuba 大数乘法算法和快速模约算法实现。前者性能低，资源开销也低，后者性能高，资源开销也较高。对于本文而言，前者更适合。以文献[32]提供的 GF(2^{128})迭代乘法算法为基础，低成本的 GF(2^{128})迭代乘法算法硬件架构如图 6 所示。可以看出，该架构较为简单，仅由两个 128 bit 的寄存器、两个 128 bit 的异或运算、一个移位运算、5 个数据选择器以及状态控制器组成。当需要运算，首先使能状态机，然后将寄存器 V 初始化为寄存器 H 中的数值，寄存器 Z 初始化为 0。之后迭代更新寄存器 V 和 Z（如图中的更新 Z 和更新 V 所示）。

3.4　多功能设计

为了提升资源利用率，拓展 IP 用途，本文引入了 IP 的多功能设计。除了支持 SM4-GCM 算法，本文架构额外支持的两种分组密码算法工作模式如图 7 所示。由于 SM4-GCM 本身包含 SM4-CTR，单独执行 SM4-CTR 模式时，只需要复用 SM4-GCM 中的明文数据处理流程即可，区别仅在于无须 GHASH 算法计算。对于 SM4 电码本（SM4-ECB, SM4 based electronic codebook）模式，需要将 IV 寄存器作为输入，将 H 寄存器作为输出，具体如图 7(b)所示。

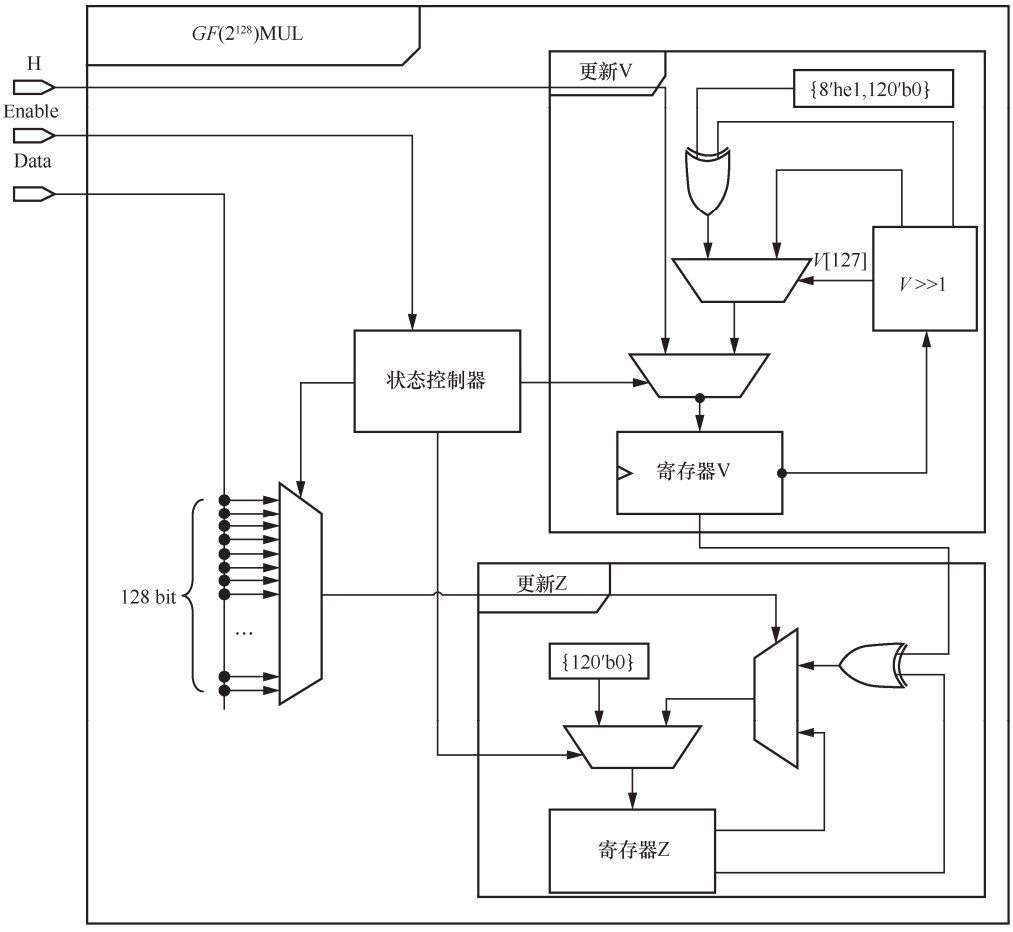
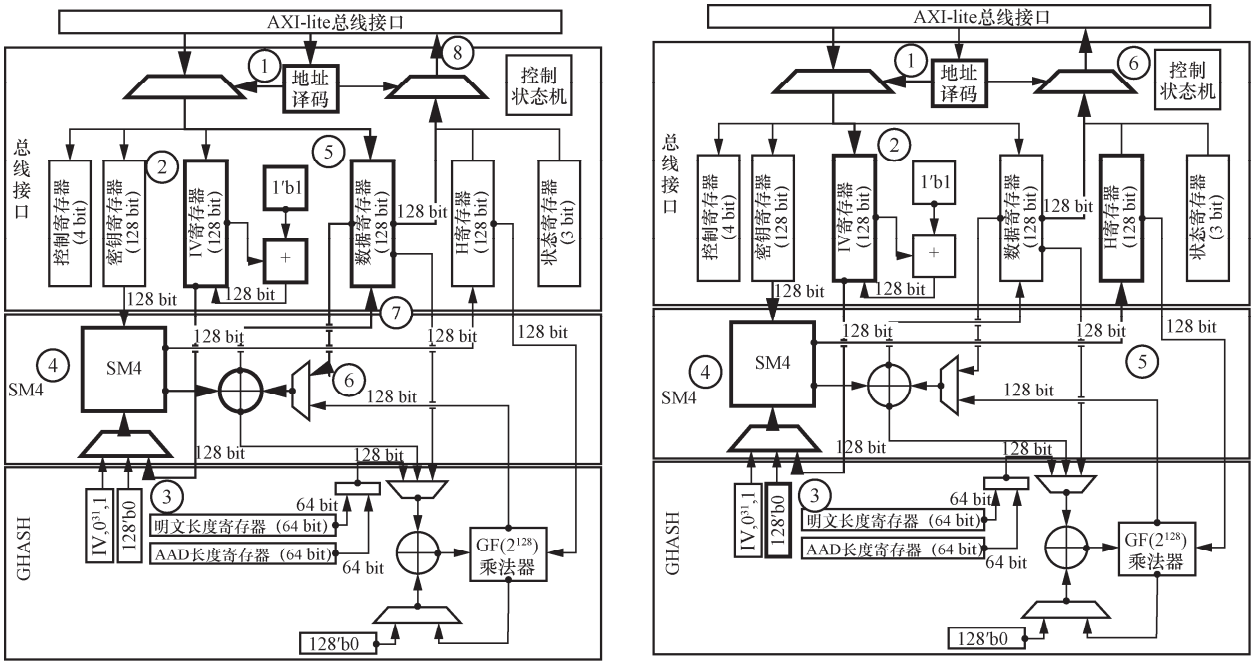


图 6　低成本的 GF(2^{128})迭代乘法算法硬件架构



(a) SM4-CTR模式　　(b) SM4-ECB模式

图 7　本文架构额外支持的两种分组密码算法工作模式

4　实验结果比较与分析

本文采取如图 8 所示的实验方法对 SM4-GCM 算法架构设计进行评估。实验流程包括设计与验证、IP 封装与 SoC 集成、软件 API 测试。首先，采用 Verilog HDL 算法对电路架构设计进行描述；然后，通过 Xilinx Vivado 仿真器进行功能仿真并评估其性能，SM4-GCM 算法仿真波形如图 9 所示；之后，将其封装到 AXI-lite 总线接口的 IP 地址，并将其挂载到 Xilinx ZYNQ SoC 系统上，原型验证搭建的 SoC 系统如图 10 所示，原型验证环境如图 11 所示；接着，对整个 SoC 系统进行 FPGA 实现，并导出码流文件和 XSA 文件；最后，将码流烧录至 FPGA 芯片，并编写硬件 IP 的驱动，调用 API 向 IP 核写入测试数据，通过串口终端查看和检查计算结果是否正确，原型验证终端输出结果如图 12 所示。

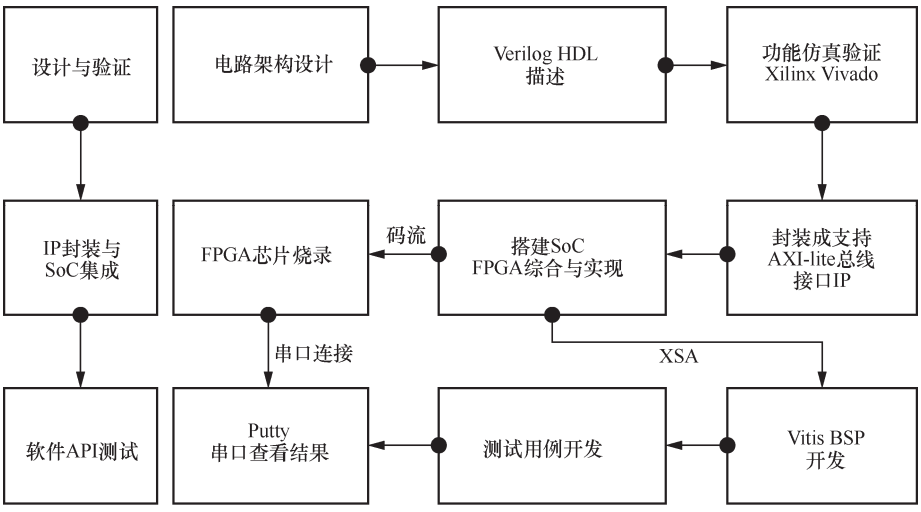


图 8　实验方法

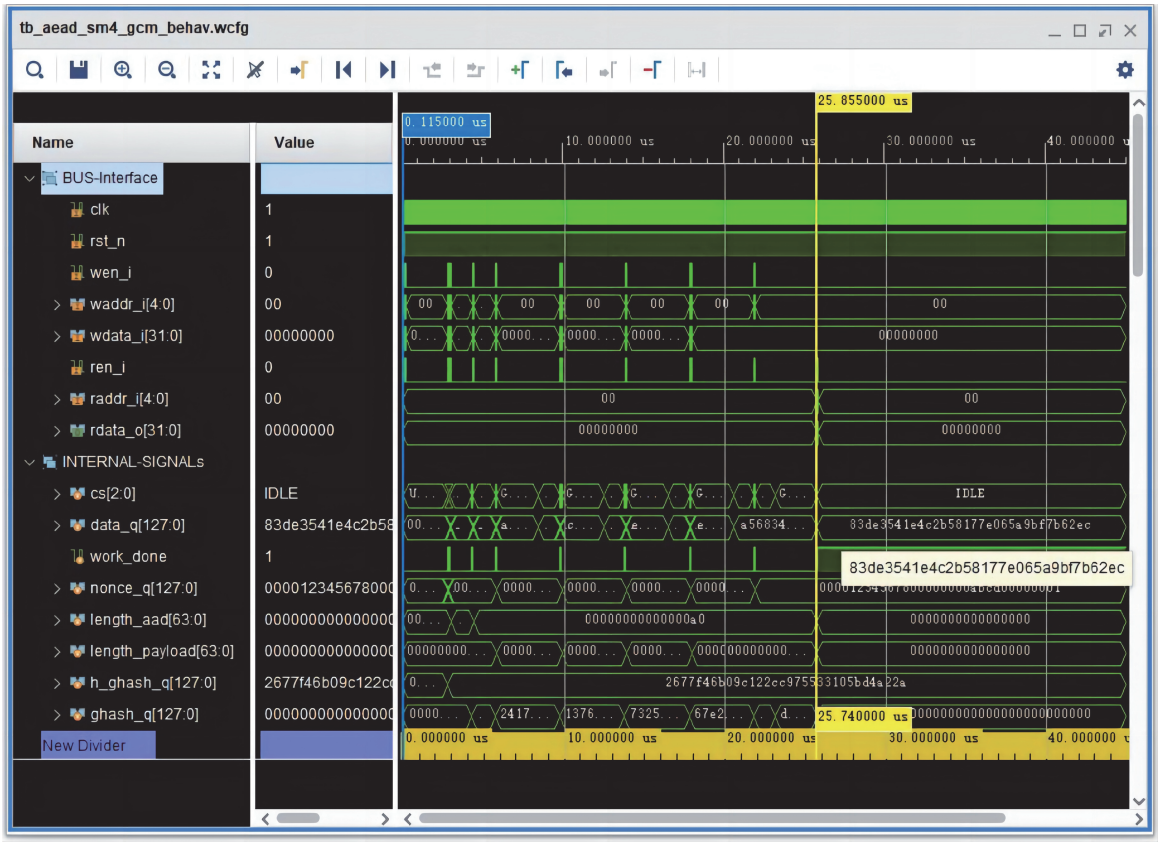


图 9 SM4-GCM 算法仿真波形

4.1　吞吐率

本文以 RFC8998 提供的 SM4-GCM 算法示例数据为测试数据（测试数据包含 128 bit 密钥、IV、AAD 数据和明文，组成 8 个 128 bit 的数据块，共 1 024 bit），从仿真结果来看，生成的认证码与参考数据一致，证明本文设计的功能正确性。吞吐率评估结果见表 2。从加载密钥至认证码的生成共消耗了 2 574 个时钟周期，因此吞吐率的评估结果为 39.78 Mbit/s @ 100 MHz。

表 2　　吞吐率评估结果

指标	评估结果
时钟周期	2 574 Cycle
吞吐率	0.397 8 bit/Cycle 39.78 Mbit/s@100 MHz

4.2　FPGA 资源开销

本文设计在 FPGA 芯片 Xilinx ZYNQ 7020（xc7z020clg400-1）上实现，FPGA 资源开销见表 3。本文设计未使用 Block RAM 和数字信号处理器（DSP, digital signal processor），占用的查找表（LUT, look-up table）和寄存器也较少，占用的资源片（slice）仅为 4.54%。

表 3　　FPGA 资源开销

指标	占用量/个	占比
Slice LUT	1 761	3.53%
Slice Registers	1 825	1.72%
slice	604	4.54%
Block RAM	0	0
DSP	0	0

4.3　FPGA 板级验证

图 10 是基于 Xilinx ZYNQ 搭建的片上（SoC, system-on-chip）系统。测试数据通过总线以存储映射的方式写入 IP（图中名为 ipSM4_GCM_0 的模块）。图 11 为原型验证环境，包含一台安装有 Xilinx Vivado 2022.1 和 Vitis 2022.1 FPGA 开发套件的笔记本计算机和一块 PYNQ Z2 开发板，二者通过 USB 接口线连接。图 12 为串口终端输出结果，与仿真结果保持一致。

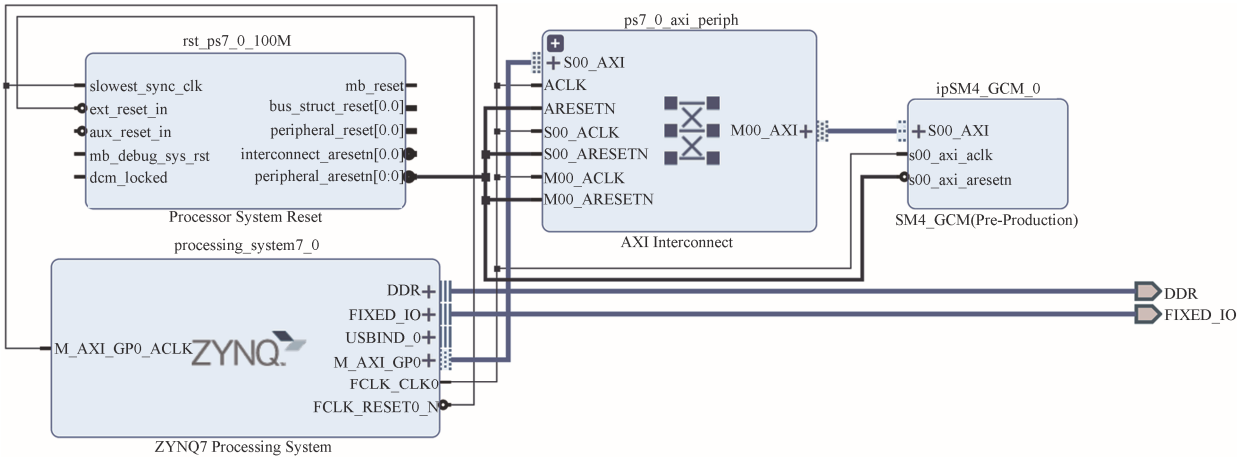


图 10　原型验证搭建的 SoC 系统

图 11　原型验证环境

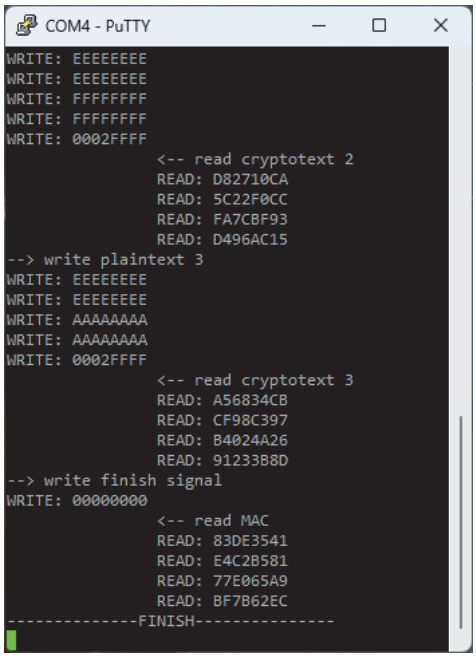


图 12　原型验证终端输出结果

4.4　文献比较

SM4-GCM 算法实现对比见表 4，可以看出，仅有本文设计面向物联网低成本应用领域，而其他文献均面向高性能领域。领域不同，设计指标不同，因此，本文的架构设计面积开销和性能均较低，但足以满足物联网应用需求，如主流物联网同通信协议 LoRa 的数据速率为 0.3～50 kbit/s，NB-IoT 上行和下行速率分别为 234.7 kbit/s 和 204.8 kbit/s[33]，而本文的架构设计吞吐率达到 39.78 Mbit/s@100 MHz，即使工作在 10 MHz 的低频时钟下，吞吐率也达到 3.978 Mbit/s。

表 4　　SM4-GCM 算法实现对比

文献	领域	实现	面积	吞吐率
[21]	高性能	GPU	—	1.62 Gbit/s
[22]	高性能	ZYNQ 7035 FPGA	12 579 FF + 22 856 LUT	28.16 Gbit/s
[23]	高性能	Virtex-5 FPGA	10 609 FF + 10 609 LUT	32.1 Gbit/s
本文	低成本	ZYNQ 7020 FPGA	604 slices (1 825 FF + 1 761 LUT)	39.78 Mbit/s

5　结束语

面向物联网领域，本文提供了一个低成本、多功能的 SM4-GCM 算法硬件架构设计方案，为物联网设备采集的数据提供机密性和完整性保障。FPGA 芯片上的实现结果显示，本文设计仅需 1 761 个 LUT 和 1 825 个寄存器，占用的 slice 仅有 604 个，吞吐率达到 39.78 Mbit/s@100 MHz，能够满足物联网应用需求。未来将以此 IP 为基础，开发更多的数据保护应用，如图像加密。

参考文献

[1] YANG P, XIONG N X, REN J L. Data security and privacy protection for cloud storage: a survey[J]. IEEE Access, 2020(8): 131723-131740.

[2] UKIL A, BANDYOAPDHYAY S, PURI C, et al. IoT healthcare analytics: the importance of anomaly detection[C]//Proceedings of 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA). Piscataway: IEEE Press, 2016: 994-997.

[3] SELVARAJ S, SUNDARAVARADHAN S. Challenges and opportunities in IoT healthcare systems: a systematic review[J]. SN Applied Sciences, 2020, 2(1): 139.

[4] DORRI A, KANHERE S S, JURDAK R, et al. Blockchain for IoT security and privacy: the case study of a smart home[C]//Proceedings of 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). Piscataway: IEEE Press, 2017: 618-623.

[5] ZHENG S, APTHORPE N, CHETTY M, et al. User perceptions of smart home IoT privacy[J]. Proceedings of the ACM on Human-Computer Interaction, 2(CSCW): 1-20.

[6] LIN H C, BERGMANN N. IoT privacy and security challenges for smart home environments[J]. Information, 2016, 7(3): 44.

[7] YU W J, LIU Y H, DILLON T, et al. An integrated framework for health state monitoring in a smart factory employing IoT and big data techniques[J]. IEEE Internet of Things Journal, 2022, 9(3): 2443-2454.

[8] 绿盟科技. 守望高质量：网络安全 2022[EB]. 2023.
NSFOCUS. Watching for high quality: network security 2022[EB]. 2023.

[9] United States Department of Homeland Security. Strategic principles for securing the internet of things[EB]. 2016.

[10] YANG P. ShangMi (SM) Cipher Suites for TLS 1.3[EB]. 2021.

[11] BAI K P, WU C K. A secure white-box SM4 implementation[J]. Security and Communication Networks, 2016, 9(10): 996-1006.

[12] 潘文伦, 秦体红, 贾音, 等. 对两个 SM4 白盒方案的分析[J]. 密码学报, 2018, 5(6): 651-670.
PAN W L, QIN T H, JIA Y, et al. Cryptanalysis of two white-box SM4 implementations[J]. Journal of Cryptologic Reseatch, 2018, 5(6): 651-670.

[13] 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 Reseatch, 2018, 5(6): 641-650.

[14] WEI M, SUN S W, WEI Z H, et al. Unbalanced sharing: a threshold implementation of SM4[J]. Science China Information Sciences, 2021, 64(5): 1-3.

[15] ZHOU F, ZHANG B J, WU N, et al. The design of compact SM4 encryption and decryption circuits that are resistant to bypass attack[J]. Electronics, 2020, 9(7): 1102.

[16] YU S Y, LI K L, LI K Q, et al. A VLSI implementation of an SM4 algorithm resistant to power analysis[J]. Journal of Intelligent & Fuzzy Systems, 2016, 31(2): 795-803.

[17] NIU Y B, JIANG A P. The low power design of SM4 cipher with resistance to differential power analysis[C]//Proceedings of Sixteenth International Symposium on Quality Electronic Design. Piscataway: IEEE Press, 2015: 470-474.

[18] 王磊, 龚征, 刘哲, 等. 基于塔域的 SM4 算法快速软件实现[J]. 密码学报, 2022, 9(6): 1081-1098.
WANG L, GONG Z, LIU Z, et al. Fast software implementation of SM4 based on tower field[J]. Journal of Cryptologic Reseatch, 2022, 9(6): 1081-1098.

[19] KWON H, KIM H, EUM S, et al. Optimized implementation of SM4 on AVR microcontrollers, RISC-V processors, and ARM processors[J]. IEEE Access, 2022, 10: 80225-80233.

[20] ZHANG J B, MA M, WANG P. Fast implementation for SM4 cipher algorithm based on bit-slice technology[M]. Lecture Notesin Computer Science. Cham: Springer International Publishing, 2018: 104-113.

[21] 张才贤. 基于CUDA的并行SM4-GCM设计与实现[D]. 西安: 西安电子科技大学, 2019.
ZHANG C X. Design and implementation of parallel SM4-GCM based on CUDA[D]. Xi'an: Xi’dian University, 2019.

[22] 翟嘉琪, 李斌, 周清雷, 等. 基于FPGA的高性能可扩展SM4-GCM算法实现[J]. 计算机科学, 2022, 49(10): 74-82.
ZHAI J Q, LI B, ZHOU Q L, et al. Implementation of FPGA-based high-performance and scalable SM4-GCM algorithm[J]. Computer Science, 2022, 49(10): 74-82.

[23] LI L, YANG F, PAN Y M, et al. An implementation method for SM4-GCM on FPGA[C]//Proceedings of 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC).Piscataway: IEEE Press, 2017: 1977-1981.

[24] CHEN R, LI B. Exploration of the high-efficiency hardware architecture of SM4-CCM for IoT applications[J]. Electronics, 2022, 11(6): 935.

[25] ZHENG L, LI C T, LIU Z B, et al. Implementation of high throughput XTS-SM4 module for data storage devices[M]. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Cham: Springer International Publishing, 2018: 271-290.

[26] 樊凌雁, 周盟, 骆建军, 等. 多引擎并行CBC模式的SM4算法的芯片级实现[J]. 计算机研究与发展, 2018, 55(6): 1247-1253.
FAN L Y, ZHOU M, LUO J J, et al. IC design with multiple engines running CBC mode SM4 algorithm[J]. Journal of Computer Research and Development, 2018, 55(6): 1247-1253.

[27] GUAN Z Y, LI Y H, SHANG T, et al. Implementation of SM4 on

FPGA: trade-off analysis between area and speed[C]//Proceedings of 2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR). Piscataway: IEEE Press, 2018: 192-197.

[28] ZHU K S, ZHANG L C, DAI Z B, et al. Design and implementation of low-cost SM4 for consumer electronic product[C]//Proceedings of 2016 IEEE International Conference on Consumer Electronics-China (ICCE-China). Piscataway: IEEE Press, 2016: 1-5.

[29] LI Y Q, WU X J, BAI G Q. Implementation of SM4 algorithm based on asynchronous dual-rail low-power design[C]//Proceedings of 2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT). Piscataway: IEEE Press, 2018: 1-3.

[30] ZHENG X, XU C Y, HU X H, et al. The software/hardware co-design and implementation of SM2/3/4 encryption/decryption and digital signature system[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020, 39(10): 2055-2066.

[31] KAVUN E B, MENTENS N, VLIEGEN J, et al. Efficient utilization of DSPs and BRAMs revisited: new AES-GCM recipes on FPGAs[C]//Proceedings of 2019 International Conference on ReConFigurable Computing and FPGAs (ReConFig). Piscataway: IEEE Press, 2019: 1-2.

[32] DWORKIN M. NIST special publication 800-38D: recommendation for block cipher modes of operation: galois/counter mode (GCM) and GMAC[EB]. 2023.

[33] SINHA R S, WEI Y Q, HWANG S H. A survey on LPWA technology: LoRa and NB-IoT[J]. ICT Express, 2017, 3(1): 14-21.

[作者简介]

陈锐（1986− ），男，博士，南京工业职业技术大学讲师，主要研究方向为物联网安全、FPGA 应用等。

李春强（1975− ），男，博士，南京工业职业技术大学讲师，主要研究方向为网络安全、物联网安全等。