DOI：10.20189/j.cnki.CN/61-1527/E.202501012

基于PTPv2协议的回波模拟器加密时钟同步系统设计

李　森
（中国兵器装备集团自动化研究所有限公司，四川 绵阳 621000）

摘　要：为满足雷达回波模拟器对时钟同步与同步时间安全性的需求，在现场可编程门阵列（Field Programmable Gate Array，FPGA）上采用精确时间协议（Precision Time Protocol，PTP）与SM4加密算法，实现回波模拟器之间的系统时钟同步，精确时间协议采用PTPv2协议版本实现。在FPGA上设计PTP主时钟与从时钟模块，使回波模拟器在组网时可以通过实际需求由软件配置为PTP主时钟或从时钟；PTP主时钟控制引擎采用超时控制机制，防止PTP主时钟在时钟同步的交互过程中由于延迟请求报文异常，导致状态机卡死；时钟偏移的计算采用归一到纳秒的方式，提高时钟偏移的计算精度；采用SystemVerilog语言设计SM4加密算法的加密模块、解密模块，实现对时间戳信息的加密、解密，解决时间戳信息传输的安全性需求；采用“PTP over UDP over IPv4”方式在传输层对PTP协议报文进行封装，实现PTP报文通过以太网进行接收与发送。将时钟同步系统在AMD XC7K325TFFG676 FPGA芯片上进行设计实现，测试结果表明：该时钟同步系统性能稳定，且在直连情况下同步误差均小于500 ns，满足设计指标要求。

关键词：PTPv2协议；时钟同步；FPGA；SM4加密算法；UDP协议栈
中图分类号：TN955　　**文献标志码**：A　　**开放科学标识码**（OSID）：
文章编号：2097-2741(2025)02-113-11

听语音　与作者互动　聊科研

PTPv2 Protocol-based Design of an Encrypted Clock Synchronization System for Echo Simulators

LI Sen
（Automation Research Institute Co., Ltd. of China South Industries Group Corporation, Mianyang 621000, China）

ABSTRACT: To satisfy the requirements of clock synchronization and synchronization time security for radar echo simulators, a PTP (Precision Time Protocol) and the SM4 encryption algorithm were implemented on a FPGA (Field Programmable Gate Array) to achieve system clock synchronization between echo simulators. The PTPv2 protocol version was adopted for implementation of the PTP. The PTP master clock and slave clock modules were designed on the FPGA, enabling echo simulators to be configured as PTP master clocks or slave clocks according to

收稿日期：2024-10-19

第一作者：李森（1993—），男，硕士研究生，主要从事数字集成电路与军用特种电子研究。E-mail：1694335941@qq.com

引用格式：李森. 基于PTPv2协议的回波模拟器加密时钟同步系统设计[J]. 火箭军工程大学学报，2025，39（2）：113-123. *LI Sen. PTPv2 protocol-based design of an encrypted clock synchronization system for echo simulators[J]. Journal of Rocket Force University of Engineering, 2025, 39(2): 113-123.*

the actual demands during networking. In the design of the PTP master clock control engine, overtime control was adopted to prevent the PTP master clock from receiving abnormal delay request messages during clock synchronization interaction, which resulted in the freezing of the state machine. Normalization to nanoseconds was used to improve the calculation accuracy of the clock offset. The encryption and decryption modules of the SM4 encryption algorithm were designed using the System Verilog language to encrypt and decrypt the timestamp information, and meet the security requirements of timestamp information transmission. The PTP protocol messages were encapsulated at the transport layer by using the "PTP over UDP over IPv4" method to enable the reception and transmission of PTP messages through Ethernet. Finally, a clock synchronization system was designed and implemented on the AMD XC7K325TFFG676 FPGA chip. The test results show that the clock synchronization system is stable in terms of performance, and the synchronization error was less than 500 ns in the direct connection, meet ing the design specifications.

KEYWORDS: PTPv2 protocol; clock synchronization; FPGA; SM4 encryption algorithm; UDP stack

回波模拟器能够模拟产生雷达实际工作过程中所接收到的回波信号，在雷达的研制、生产、测试过程中，发挥着非常重要的作用。在实际测试时，常常需要多台回波模拟器通过以太网组成一个分布式控制系统进行协作测试。在分布式控制系统中，对各节点的测量数据、事件记录、日志等数据进行融合，以及各节点之间的协同工作都需要各节点之间保持一致的时间序列，其关键在于各节点之间的时间必须同步，需要回波模拟器各节点必须具备时钟同步功能。然而，在系统工作过程中，由于回波模拟器物理时钟的分散性以及时钟漂移[1]，随着运行时间的推移，将导致各个回波模拟器的时钟不统一，且时钟偏差越来越大，严重影响系统正常运行。

目前常用的时钟同步技术有全球定位系统（Global Positioning System，GPS）、北斗卫星授时[2-3]、网络时间协议[4]（Network Time Protocol，NTP）以及精确时间协议（Precision Time Protocol，PTP）。GPS与北斗可以实现纳秒级的时间同步[5]，但是由于其需要加装通信模块，且需要能够接收卫星信号，受环境影响较大，因此使用范围受到一定的限制。PTP精确时间协议又名IEEE1588协议，目前广泛应用于工业无线传感网络、电力系统以及分布式系统中节点之间的时钟同步。基于网络进行数据报文交换实现时钟同步，不需要消耗额外的硬件资源，通过软件或FPGA编程即可实现时钟同步，相对于GPS、北斗卫星的授时方式，其同步方式应用更加广泛。单飞桥等[5]基于PTP协议设计了一种用于工业无线传感器网络的时间同步方法，利用反向自适应卡尔曼滤波算法，提高了时间同步的可靠性，解决了工业无线传感器网络对时间同步的需求。徐卓汀等[4]基于IEEE1588协议，采用时钟偏移补偿算法设计了一种应用于分布式系统的时钟同步系统，并在FPGA上进行设计实现，解决了分布式系统对时钟同步的需求。朱炎平等[6]基于IEEE1588精确时间同步协议对智能变电站中通信网络的路径延迟抖动导致的同步精度低的问题进行优化，提高了电力网络中时间同步的精度。

以上研究主要用于解决工业领域与电力系统等民用领域的时钟同步问题，且在同步过程中并未考虑时间戳数据的安全性问题。针对回波模拟器对时钟同步的需求，在考虑同步精度与硬件消耗的情况下，本文将PTP时钟同步的方式应用于装备领域，并与SM4加密算法进行结合，解决回波模拟器对时钟同步与安全的需求，采用精确时间协议PTPv2版本[7]作为回波模拟器之间的时钟同步协议。为了保证时钟同步处理的实时性，处理延时的稳定性，采用FPGA以硬件的方式实现时钟同步功能。

1 PTPv2时钟同步模型与协议

1.1 PTPv2时钟同步模型

PTPv2（IEEE 1588v2）协议通过在主从时钟节点之间发送PTPv2协议报文实现主从节点之间

的时钟同步与通信链路的延时测量。PTPv2协议的时钟同步原理如图1所示。首先，主节点发送同步报文（sync报文），并记录该报文发送的时间戳t_1，若采用one-step同步机制，则t_1时间戳直接放在sync同步报文中；若采用two-step同步机制，则将t_1时间戳放在跟随报文（follow_up报文）中发送给从节点。从节点收到sync报文后，记录sync报文的接收时间戳t_2。接着从节点向主节点发送延迟请求报文（delay_req报文），并记录报文的发送时间戳t_3。主节点收到delay_req报文后，记录接收到报文的时间戳t_4，并将t_4放在延迟请求响应报文（delay_resp报文)中发送给从节点，从节点收到delay_resp报文后提取出t_4[8-13]。

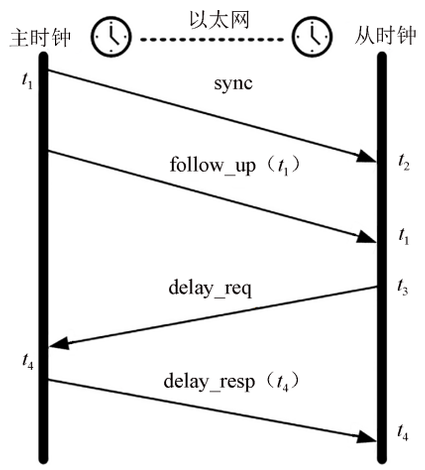


图1 PTPv2时钟同步原理

Fig. 1 PTPv2 clock synchronization principles

假设主从节点之间的通信延时对等，在对等延时机制下，设延时为D，从时钟相对于主时钟的时钟偏移为

$$S = t_m - t_s$$

式中：t_m和t_s分别为主时钟时间戳和从时钟时间戳，则：

$$S = [(t_4 - t_3) - (t_2 - t_1)]/2$$

$$D = [(t_4 - t_3) + (t_2 - t_1)]/2$$

通过以上表达式即可计算出从时钟与主时钟之间的偏差，以及链路传输延时，从时钟将自己的当前时钟t_s与时钟偏移S相加，即可得到同步后的时钟值，实现时钟同步。

1.2 PTPv2协议

PTPv2报文由报头、报文体与扩展字段3个部分组成，扩展字段长度可以为0。PTPv2协议的报文类型共有9种[14-15]。其中，事件报文类型4种，普通报文类型5种。要完成PTPv2时钟同步的过程只需要同步报文、跟随报文、延迟请求报文、延迟请求响应报文4种即可，相应报文的结构分别如表1、表2、表3所示。

表1 同步/延迟请求报文结构

Table 1 Structure of sync/delay_req message

Bits								Octets	offset
7	6	5	4	3	2	1	0		
报文头部								34	0
源时间戳								10	34

注：one-step设备，源时间戳为sync报文的发送时间戳t_1；two-step设备，为精度±1秒的时间戳；delay_req报文，源时间戳数值为0或者精度为±1秒的时间戳。

表2 跟随报文结构

Table 2 Structure of follow_up message

Bits								Octets	offset
7	6	5	4	3	2	1	0		
报文头部								34	0
精确源时间戳								10	34

注：精确源时间戳为two-step设备sync报文发送时间戳t_1。

表3 延迟请求响应报文结构

Table 3 Structure of delay_resp messages

Bits								Octets	offset
7	6	5	4	3	2	1	0		
报文头部								34	0
源时间戳								10	34
请求端口标识								10	44

注：接收时间戳，响应delay_req报文的接收时间戳t_4；请求端口标识，响应delay_req报文的发送设备端口ID。

2 系统整体设计

2.1 设计指标

时钟同步系统用于实现回波模拟器之间的时钟同步功能，该系统的功能要求如下。1）通过以太网接口实现系统时钟同步；2）采用用户数据报协议（User Datagram Protocol，UDP）作为PTPv2协议的载体；3）PTPv2报文的目的节点IP地址可配置；4）直连情况下，同步误差不超过500 ns；5）节点兼容主时钟与从时钟功能，在系统中根据需要可以将其配置为主时钟或从时钟。

2.2 整体框架设计

时钟同步系统中的以太网通信、UDP/IP协议栈以及PTPv2协议主时钟、从时钟均在FPGA上进行设计实现。整个回波模拟器时钟同步系统的

结构如图2所示。系统包括：1）三速以太网IP核（Tri-MAC IP核），用于实现数据链路层MAC（Media Access Control，MAC）控制器的功能；2）UDP_IP协议栈，用于实现UDP协议、IP协议通信功能，作为PTPv2报文的载体，实现PTPv2报文的封装与传输；3）发送仲裁控制模块，根据输入的主从时钟配置端口，实现主时钟与从时钟发送数据的仲裁选择；4）PTPv2从时钟模块，该模块由从时钟接收模块、从时钟发送模块、SM4解密模块、SM4加密模块、从时钟控制引擎以及时间戳同步计算模块6个子模块组成，用于实现PTPv2时钟同步系统中从时钟的功能；5）PTPv2主时钟模块，该模块由主时钟接收模块、主时钟发送模块、加密模块、主时钟控制引擎以及主时钟本地走时5个子模块组成，实现PTPv2时钟同步系统中主时钟的功能；6）PLL锁相环模块，用于将晶振输入的100 MHz时钟倍频为125 MHz，为内部逻辑模块提供系统时钟。

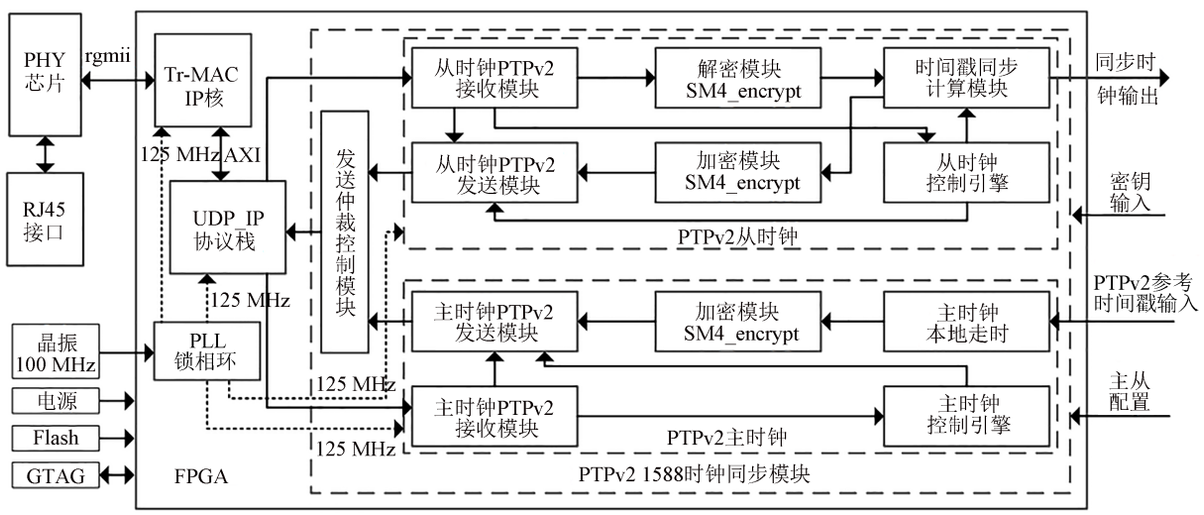


图 2　回波模拟器时钟同步系统结构

Fig. 2　Block diagram of the clock synchronization system of the echo simulator

3　PTPv2协议设计与实现

PTPv2报文的封装方式有“PTP over Ethernet”“PTP over UDP over IPv4”“PTP over UDP over IPv6”3种方式[16]，即将PTPv2报文封装在以太网帧中，或者封装在IPv4、IPv6协议之上的UDP报文中。本设计采用“PTP over UDP over IPv4”的方式实现PTPv2报文的封装，其报文结构如图3所示。

图 3　PTP over UDP over IPv4报文结构

Fig. 3　Structure of PTP over UDP over IPv4 message

3.1　PTPv2从时钟设计与实现

3.1.1　接收模块实现

从时钟接收模块（ptp_receive_slave模块）实现从时钟对PTPv2报文的接收。从时钟在时钟同步的过程中只需对sync报文、follow_up报文和delay_resp报文进行接收解析。每一种报文解析出报文头部以及报文体，提取出sync报文或follow_up报文中的时间戳t_1、delay_resp报文中的时间戳t_4，并输出报文信息有效控制信号。整个PTPv2报文的接收过程采用并行状态机实现，其状态转移如图4所示。

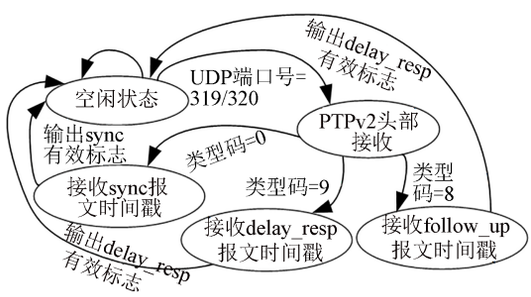


图 4　ptp_receive_slave 模块状态机

Fig. 4　State machine of ptp_receive_slave module

首先，判断UDP报文的端口号，当端口号为319或320时，对报文头部进行解析，并对其中的类型字段进行判断。当类型码=0时，说明为sync报文，提取出t_1时间戳，输出sync报文有效标志；

同理，当类型码=8、类型码=9时，提取出相应的时间戳，并输出follow_up报文、delay_resp报文有效标志，回到空闲状态，等待下一帧PTPv2报文到达。

3.1.2 发送模块实现

从时钟发送模块（ptp_send_slave模块）用于发送delay_req报文。在PTPv2时钟同步的过程中，当从时钟接收到主时钟的sync报文后，需要立刻向主时钟发送delay_req报文。ptp_send_slave模块采用状态机实现对发送过程的控制，其状态转移过程如图5所示。当检测到delay_req发送使能有效时，锁存待发送delay_req报文的头部信息，接着进入到delay_req报文头部发送状态；当delay_req报文的头部发送完毕后，进入到delay_req报文时间戳发送状态，发送完毕后回到空闲状态。

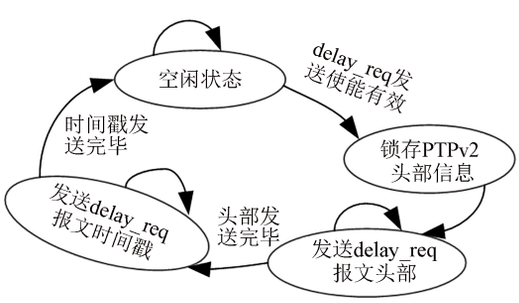


图5 ptp_send_slave模块状态机

Fig. 5 State machine of ptp_send_slave module

3.1.3 控制引擎设计

从时钟控制引擎（ptp_ctrl_engine_slave模块）用于实现从时钟整个同步过程的控制工作，对时钟同步过程的3次交互以及时间戳同步计算模块的计算过程进行控制。该模块采用状态机进行设计，为了防止状态机卡死，在设计中采用了超时控制机制，对等待交互的状态进行超时控制，其状态转移过程如图6所示。开始工作时，状态机处于空闲状态，当检测到sync报文有效时，如果为one-step类型的主时钟，则跳转到“发送delay_req报文状态”；如果为two-step类型的主时钟，则跳转到“follow_up报文等待状态”。当状态机处于“follow_up报文等待状态”时，检测到follow_up报文有效，则直接进入到“delay_req报文发送状态”；否则一直在该状态等待，直到超时时间到达。当状态机处于“delay_resp报文等待状态”时，会一直检测delay_resp报文是否有效，当检测到delay_resp报文有效时，输出PTPv2同步过程交互完成信号；否则一直在该状态等待，直到超时时间到达后，进入到空闲状态，等待下一轮PTPv2时钟同步交互。

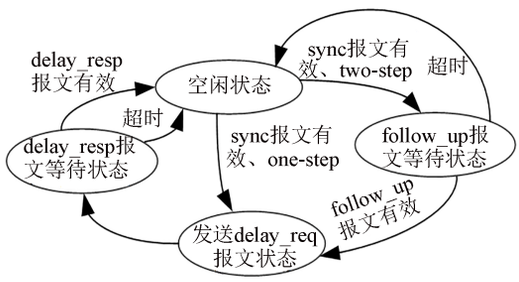


图6 ptp_ctrl_engine_slave模块状态机

Fig. 6 State machine of ptp_ctrl_engine_slave module

3.1.4 时间戳同步计算模块设计

时间戳同步计算模块[17-18]为从时钟同步计算的核心处理模块，在ptp_ctrl_engine_slave模块的控制下，完成以下功能：

1） t_1、t_2、t_3、t_4时间戳记录；

2） 计算从时钟与主时钟之间的时钟偏移S；

3） 从时钟本地走时控制。

时间戳同步计算模块检测到控制引擎输出的t_1～t_4时间戳有效标志时，对t_1～t_4时间戳进行锁存，当检测到同步交互完成标志信号有效时，开始时钟偏移计算。由于FPGA硬件计算的特性，为了提高时钟同步的计算精度，在计算时钟偏移S时将原有“秒+纳秒”形式的时间戳归一为以纳秒为单位的时间戳，进行时钟偏移计算，并将时钟偏移S与从时钟的当前时间进行叠加，得到同步后的时钟结果。其余时间时钟同步计算模块进行正常走时，输出本地时间戳，为本地系统中的日志、事件和控制序列提供时间戳标记。

3.2 主时钟设计与实现

3.2.1 发送模块实现

主时钟发送模块（ptp_send_master模块）用于sync报文、delay_resp报文的发送。在PTPv2时钟同步的过程中，首先，主时钟需要主动发送sync报文给从时钟，接着等待从时钟的delay_req报文，当接收到从时钟的delay_req报文时，记录delay_req报文的接收时间t_4，并将该时间放在delay_resp报文中回复给对应的从时钟，ptp_send_master模块采用状态机实现发送过程的控制，其状态转移过程如图7所示。

状态机开始工作时处于空闲状态，当检测到sync报文发送使能有效时，进入“sync报文信息准备状态”，然后进入“发送sync报文状态”，直到

sync报文发送完毕，回到空闲状态。同理，当delay_res报文发送使能有效时，进行delay_resp报文的发送处理过程，与sync报文发送处理相同。当在空闲状态时，delay_resp报文发送使能与sync报文发送使能同时有效，sync报文的发送优先级高于delay_resp报文。

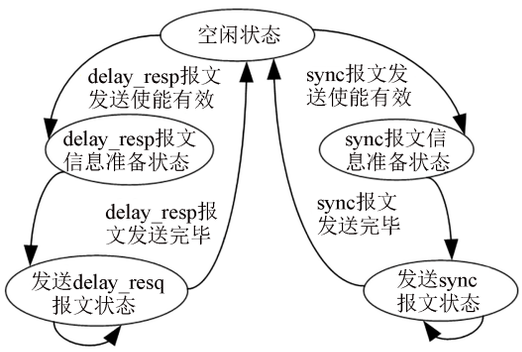


图 7　ptp_send_master 模块状态机

Fig. 7　State machine of ptp_send_master module

3.2.2　接收模块实现

主时钟接收模块（ptp_receive_master模块）实现主时钟对PTPv2报文的接收。主时钟在时钟同步的过程中只需对delay_req报文进行接收解析，忽略丢弃其余类型的PTPv2报文，当报文类型码=1时，说明为delay_req报文，解析出报文的时间戳，输出delay_req报文有效标志，并回到空闲状态，整个接收模块的状态转移过程如图8所示。

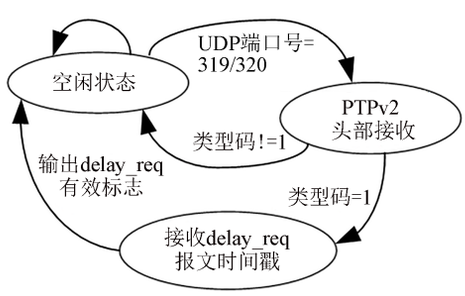


图 8　ptp_receive_master 模块状态机

Fig. 8　State machine of ptp_receive_master module

3.2.3　控制引擎设计

主时钟控制引擎（ptp_ctrl_engine_master模块）用于实现主时钟整个时钟同步过程的控制工作，对时钟同步过程的3次交互过程进行控制。该引擎采用状态机进行设计，为防止状态机卡死，与从时钟控制引擎相同，在设计中采用超时控制机制，对等待交互的状态进行超时控制，防止状态机一直在该状态等待而卡死，其状态转移过程如图9所示。开始工作时，状态机处于空闲状态，随后进入“sync报文使能状态”，输出sync发送使能控制信号，然后跳转到“等待delay_req报文状态”，等待delay_req报文的到来。如果在设定的超时时间范围内，delay_req报文有效，则跳转到“发送间隔延迟状态”，在设定的超时时间后，则delay_req报文无效，跳转到空闲状态，结束本次时钟同步交互过程。为了便于根据实际情况对超时时间与发送间隔时间进行调节，在设计中将超时时间与间隔时间设为可配置，以便于处理器系统对其进行配置。

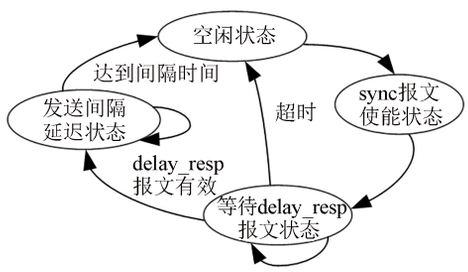


图 9　ptp_ctrl_engine_master 模块状态机

Fig. 9　State machine of ptp_ctrl_engine_master module

4　SM4加、解密逻辑设计

SM4密码算法是一种对称加、解密算法[19-20]，该算法由密钥扩展算法、加解密迭代算法2个基本算法体构成，基本算法体均由32轮的非线性迭代构成。SM4算法的数据分组与密钥分组均为128 bit[21]，加密过程就是将输入的128 bit明文数据经过加密计算后，转换为128 bit的密文输出；同理，解密就是将输入的128 bit密文数据通过解密运算后，转换为128 bit明文输出。加密运算与解密运算结构完全相同，只是轮密钥的使用顺序相反。

4.1　密钥扩展算法实现

加密运算与解密运算均需进行32轮非线性迭代[22]，每一轮非线性迭代均需一个32 bit的轮密钥数据，而原始输入的密钥为128 bit，最多只能支持4次迭代。因此，需要对输入的原始密钥MK进行密钥扩展，生成32个32 bit数据的轮密钥。轮密钥rk扩展算法的计算过程如下。

1）设输入密钥：$MK=(MK_0, MK_1, MK_2, MK_3)$，$MK_i \in Z_{2^{32}}$，$i=0, 1, 2, 3$；

2）计算轮密钥迭代初始值：K_0, K_1, K_2, K_3，且$(K_0, K_1, K_2, K_3)=(MK_0 \oplus FK_0, MK_1 \oplus FK_1, MK_2 \oplus FK_2, MK_3 \oplus FK_3)$；

3）计算轮密钥rk_i：$rk_i = K_{i+4}(K_{i+1} \oplus K_{i+2} \oplus$

$K_{i+3}\oplus CK_i$），进行32轮迭代运算，生成rk_0，rk_1，…，rk_{31}；

4）将生成的32个32 bit轮密钥rk_i存储在轮密钥矩阵中。

说明：（1）Z_{2^n}表示n位的二进制序列，$(Z_{2^n})^4$表示4个n位二进制序列；（2）（FK_0，FK_1，FK_2，FK_3）为SM4算法的固定参数，其值分别为：$FK_0=(\mathrm{A3B1BAC6})_{\mathrm{H}}$，$FK_1=(\mathrm{56AA3350})_{\mathrm{H}}$，$FK_2=(\mathrm{677D9197})_{\mathrm{H}}$，$FK_3=(\mathrm{B27022DC})_{\mathrm{H}}$，$CK_i$为固定常数，其取值参见SM4算法国密标准相关文档，这里不再赘述；（3）$\oplus$表示按位异或运算；（4）$T'(\cdot)$为非线性变换算子，由非线性变换$\tau(\cdot)$与线性变换$L'(\cdot)$2部分组成，即：$T'(\cdot)=L'(\tau(\cdot))$；（5）设线性变换$L'(\cdot)$的输入为$B\in Z_{2^{32}}$，则：$L'(B)=B\oplus(B<<<13)\oplus(B<<<23)$；（6）$B<<<i$，表示将$B$循环左移$i$位；（7）非线性变换$\tau(\cdot)$由4个并行的Sbox盒构成，设非线性变换$\tau(\cdot)$的输入为$A=(a_0,a_1,a_2,a_3)\in(Z_{2^8})^4$，则$\tau(A)=(Sbox(a_0),\ \mathrm{Sbox}(a_1),\ \mathrm{Sbox}(a_2),\ \mathrm{Sbox}(a_3))$，其中，Sbox盒[23]映射表可以通过SM4算法国密标准查看。

根据密钥扩展算法的计算原理，密钥扩展算法模块（round_key_gen模块）的处理逻辑如图10所示。

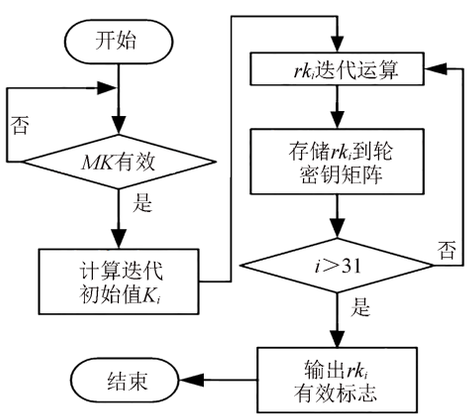


图10　round_key_gen模块处理逻辑

Fig. 10　Processing logic of round_key_gen module

当检测到密钥MK有效时，开始计算轮密钥的迭代初始值K_i，迭代初始值计算完毕后，进入到轮密钥rk_i迭代运算；当本迭代运算计算完毕后，将所得的轮密钥rk_i存储到轮密钥矩阵中，便于后续加密、解密迭代运算时直接使用。每一次迭代计算完毕后，将迭代次数i加1，并对其进行判断，当$i>31$时，说明32轮迭代计算完毕，此时输出轮密钥rk_i有效标志，结束密钥扩展运算。

4.2　加解密轮函数实现

加密运算是将输入的128 bit明文$X=(X_0,X_1,X_2,X_3)\in(Z_{2^{32}})^4$经过32轮迭代运算，变成128 bit的密文$Y=(Y_0,Y_1,Y_2,Y_3)\in(Z_{2^{32}})^4$输出。其计算过程如下。

1）设输入明文：$X=(X_0,X_1,X_2,X_3)$，$X_i\in(Z_{2^{32}})$；

2）进行加密轮函数迭代运算，加密运算的迭代格式为：$X_{i+4}=X_i\oplus \mathrm{T}(X_{i+1}\oplus X_{i+2}\oplus X_{i+3}\oplus rk_i)$，$(i=0,1,\cdots,31)$，直至32轮迭代计算完毕[24]；

3）反序变换：将最后一次迭代计算的结果X_{35}，X_{34}，X_{33}，X_{32}进行反序变换$(Y_0,Y_1,Y_2,Y_3)=(X_{35},X_{34},X_{33},X_{32})$，即得密文$Y=(Y_0,Y_1,Y_2,Y_3)$。

说明：（1）$T(\cdot)$为非线性变换算子，由非线性变换$\tau(\cdot)$与线性变换$L(\cdot)$2部分组成，即：$T(\cdot)=L(\tau(\cdot))$；（2）非线性$\tau(\cdot)$变换与密钥扩展中的$\tau(\cdot)$完全相同；（3）设线性变换$L(\cdot)$的输入为$B\in Z_{2^{32}}$，则$L(B)=B\oplus(B<<<2)\oplus(B<<<10)\oplus(B<<<18)(B<<<24)$，2、10、18、24为轮函数线性变换的参数。

解密运算与加密运算采用完全相同的迭代结构进行，只是在进行迭代运算时，使用轮密钥rk_i的顺序与加密运算相反，按照rk_{31}，rk_{30}，…，rk_1，rk_0的顺序参与轮函数的迭代运算。

根据加解密轮函数的计算原理，轮函数计算模块（round_fun模块）的处理逻辑如图11所示。当检测到明文X有效时，则检测轮密钥rk_i是否有效；当明文X与轮密钥rk_i均有效时，进入轮函数迭代计算运算结构；当32轮迭代运算完毕后，将最后一次迭代运算的输出X_{i+1}，X_{i+2}，X_{i+3}，X_{i+4}进行反序变换，得到密文Y输出，同时输出Y有效标志，完成本次加密计算过程。解密计算过程只需将轮函数的原始输入变量变为密文Y，轮密钥的输入顺序变为逆序输入，即由原来的rk_0，rk_1，…，rk_{31}输入，变为rk_{31}，rk_{30}，…，rk_1，rk_0，最后将计算结果进行反序变换，即得解密后的明文[25]。

4.3　SM4_encrypt、SM4_decrypt模块逻辑设计

SM4_encrypt、SM4_decrypt模块的结构如图12所示，由轮密钥生成模块round_key_gen模块与轮函数迭代计算模块round_fun2个子模块组成。当轮密钥使能信号key_en = 1时，round_key_gen模块读取密钥key输入，进行密钥扩展计算，当计

算完毕后，rk_valid信号变为1，表示轮密钥有效；rk为轮密钥输出，rk_cnt为轮密钥矩阵寻址输入信号，round_key_gen模块根据rk_cnt的值输出对应的轮密钥。当data_en = 1时，round_fun模块读取待加密/解密数据，当encrypt_ctrl = 1时，round_fun进行加密运算，从d_out端口输出加密的密文；当encrypt_ctrl = 0时，round_fun进行解密运算，从d_out端口输出解密后的明文；dout_valid信号为加密/解密结果有效信号，dout_valid = 1表示加密、解密结果有效，busy为运算忙标志。

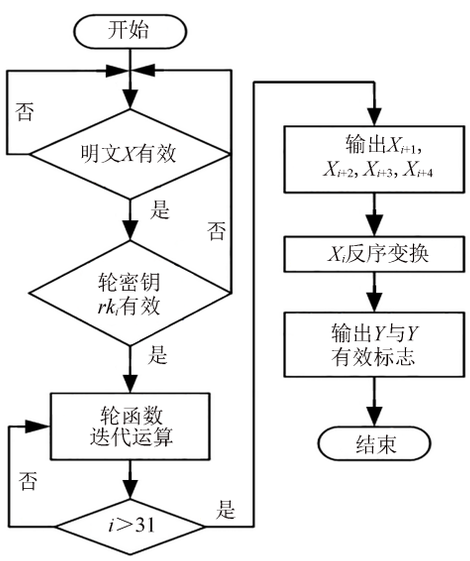


图 11　round_fun 模块处理逻辑

Fig. 11　Processing logic of round_fun module

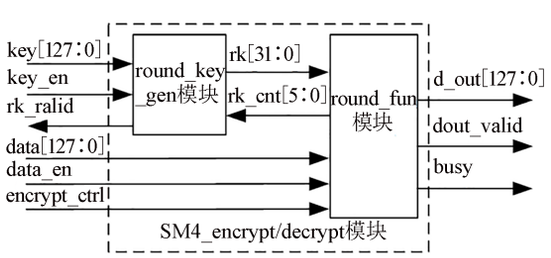


图 12　SM4_encrypt/decrypt 模块结构

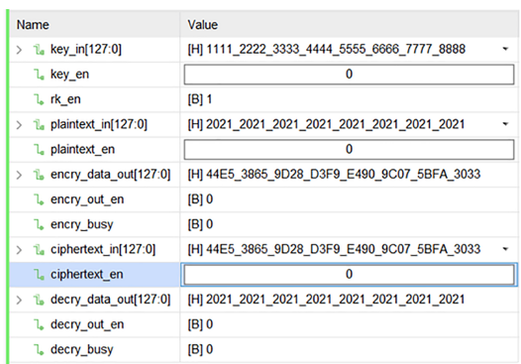
Fig. 12　Structure of SM4_encrypt/decrypt module

5　实验验证

为了对加密算法与PTPv2精确时钟同步协议进行验证，采用AMD公司的XC7K325TFFG676 FPGA作为实验平台对设计进行验证测试。

5.1　SM4加、解密模块测试

为了验证SM4加、解密模块设计的正确性，将SM4加密模块与解密模块封装在一起，采用虚拟输入输出IP核（Virtual Input Output，VIO）手动输入明文与密文，通过VIO观察对应输出的密文与明文，从测试结果中随机抽取2个，如图13所示。其中，key_in为密钥输入，key_en为密钥使能，rk_en为轮密钥有效标识，plaintext_in为明文输入，plaintext_en为明文输入使能，encry_data_out为加密结果输出，encry_out_en为加密结果有效标识，ciphertext_in为密文输入，ciphertext_en为密文输入使能，decry_data_out为解密结果输出，decry_out_en为解密结果有效标识。由测试结果可以看出，SM4加、解密模块功能正确。



Name	Value
key_in[127:0]	[H] 1111_2222_3333_4444_5555_6666_7777_8888
key_en	0
rk_en	[B] 1
plaintext_in[127:0]	[H] 2021_2021_2021_2021_2021_2021_2021_2021
plaintext_en	0
encry_data_out[127:0]	[H] 44E5_3865_9D28_D3F9_E490_9C07_5BFA_3033
encry_out_en	[B] 0
encry_busy	[B] 0
ciphertext_in[127:0]	[H] 44E5_3865_9D28_D3F9_E490_9C07_5BFA_3033
ciphertext_en	0
decry_data_out[127:0]	[H] 2021_2021_2021_2021_2021_2021_2021_2021
decry_out_en	[B] 0
decry_busy	[B] 0

（a）

Name	Value
key_in[127:0]	[H] 1111_2222_3333_4444_5555_6666_7777_8888
key_en	0
rk_en	[B] 1
plaintext_in[127:0]	[H] 1995_2024_1995_2024_1995_2024_1995_2024
plaintext_en	0
encry_data_out[127:0]	[H] A500_FCA6_CE79_984F_461A_FADE_009C_3CE5
encry_out_en	[B] 0
encry_busy	[B] 0
ciphertext_in[127:0]	[H] A500_FCA6_CE79_984F_461A_FADE_009C_3CE5
ciphertext_en	0
decry_data_out[127:0]	[H] 1995_2024_1995_2024_1995_2024_1995_2024
decry_out_en	[B] 0
decry_busy	[B] 0

（b）

图 13　SM4 算法测试结果

Fig. 13　Test results of SM4 algorithm

5.2　PTPv2 协议测试

为了验证回波模拟器PTPv2时钟同步设计的正确性，将程序下载到测试的FPGA板卡中，并通过以太网接口与测试电脑相连，在电脑端运行网络调试助手，分别对主时钟与从时钟模式进行测试。

5.2.1　主时钟模式测试

将测试板卡上的FPGA配置为PTPv2主时钟，并将其IP地址设置为192.168.1.5，在测试电脑端采用网络调试助手作为PTPv2从时钟，并将其IP地址设置为192.168.1.111，端口号设置为319。FPGA以1 s为周期向网络调试助手发送sync报文；网络调试助手收到sync报文后，手动

向FPGA测试板卡发送delay_req报文；FPGA板卡接收后，向网络调试助手发送delay_resp报文，测试结果如图14所示。可以看出，sync报文发送后，发送delay_req报文，收到delay_req报文后立刻回复delay_resp报文，验证了PTPv2主时钟功能正确。

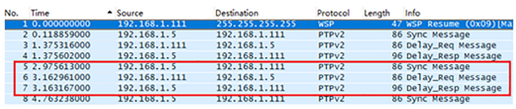


图14　主时钟wireshark抓包结果

Fig. 15　Wireshark packet capture results of the main clock

5.2.2　从时钟模式测试

与主时钟模式测试相反，从时钟模式测试时，网络调试助手作为PTPv2主时钟，FPGA测试板卡作为从时钟。将测试电脑的IP地址设置为192.168.1.100，打开网络调试助手，将FPGA侧的IP地址设为192.168.1.111，本地端口与远程端口均设置为319。在网络调试助手中发送sync同步报文，可以看到在网络调试助手的接收框中收到FPGA测试板卡发送的delay_req请求报文，用wireshark对该过程进行抓取，结果如图15所示。可以看出，测试电脑发送sync同步报文后，FPGA测试板卡立刻发送delay_req请求报文，验证了PTPv2从时钟功能正确。

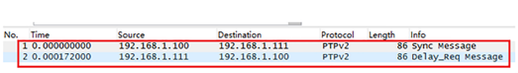


图15　从时钟wireshark抓包结果

Fig. 15　Wireshark packet caplare results of the slave clock

5.3　同步结果与时钟偏差测量

对直连模式下的同步结果与同步精度进行测试，流程如图16所示。在一片FPGA上同时装载2个PTPv2时钟同步模块，将其中一个配置为主时钟，与FPGA测试板卡上的以太网口1绑定；另一个配置为从时钟，与FPGA测试板卡上的以太网接口2绑定。将网口1与网口2通过网线进行连接，主从时钟模块的系统时钟频率为125 MHz。在测试程序中，采用逻辑分析仪IP核（Integrated Logic Analyzer，ILA）对主、从时钟的当前时间戳进行抓取，并将主、从时钟的当前时间戳送入时间戳偏差计算模块，计算主时钟与从时钟的时间戳之差，时间戳偏差计算模块输出的结果为主、从时钟时间戳差的绝对值。采用在一片FPGA上进行实时计算的方式，可以准确地计算出同一时刻主、从时间戳同步偏差的数值结果，相对于用示波器测量秒脉冲的方式[2]，该测量方式拥有更高的精度。

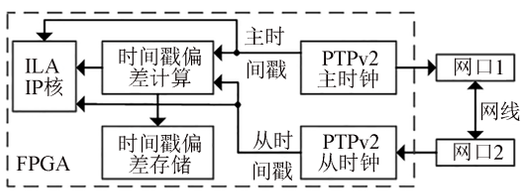


图16　时钟同步偏差测量流程

Fig. 16　Flowchart of deviation measurement of clock synchronization

随机抽取一次ILA抓取结果，如图17所示。其中，master_clk_value表示当前主时钟的时间戳，单位为ns；slave_slave_timestamp_s为从时钟的时间戳，单位为ns；timestamp_subtract为时间戳偏差值，当前采样偏差为220 ns。

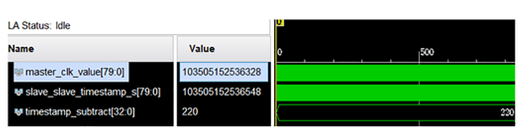


图17　时钟同步结果与偏差

Fig. 17　Results and deviations of clock synchronization

为了测量直连模式下时钟偏差情况，随机抽取某时间段，用逻辑分析仪连续观测1 h，时间戳偏差存储模块每1 s对时钟偏差进行采样，并对同步过程中时钟偏差的采样结果进行分析。在观测时间范围内，时钟同步偏差的观测结果分别为：356、352、348、228、224、220 ns，由结果可知，所测偏差均小于500 ns，满足设计指标要求。

6　结论

本文基于PTPv2协议与SM4密码算法设计了一种应用于回波模拟器的加密时钟同步系统，并在FPGA上以纯硬件的方式对系统进行设计实现。通过实验验证，主要得到以下结论。

1）该时钟同步系统可以实现回波模拟器设备之间的时钟同步，通过实验测试，在直连情况下最大同步误差为356 ns，满足系统指标要求；

2）PTPv2从时钟计算模块，采用归一到纳秒的方式进行时钟偏差计算，可以降低计算的复杂度与误差，提高时钟同步的精度；

3）采用时间戳信息加密传输的方式，相较于直接明文传输，具有更高的安全性，可以在信息融合时，增强回波模拟器对虚假数据的筛选能力；

4）相对于PTPv2主、从时钟独立设计的方式，本设计采用的主、从时钟冗余设计方式，可以增强时钟同步节点的适应性，从而根据实际需要，通过软件配置实现主、从时钟的切换。

参考文献

[1] 张巍然，侯艳红．自适应卡尔曼滤波的精密时钟同步方法[J]．单片机与嵌入式系统应用，2023，23(5)：59-64.
ZHANG W R，HOU Y H. Precision clock synchronization method based on adaptive Kalman filter [J]. Microcontroller & Embedded System，2023，23(5)：59-64.

[2] 孙攀，陈贻骜，李培宜，等．北斗、GPS卫星授时在电力系统时间同步装置中的应用[J]．电工技术，2024(15)：218-220.
SUN P，CHEN Y A，LI P Y，et al. The application of BeiDou and GPS satellite timing in power system time synchronizers[J]. Electric Engineering，2024(15)：218-220.

[3] 于合理，孙晓东，贾赞杰，等．限制环境下的GNSS精密授时方法研究综述[J]．海洋测绘，2024，44(2)：46-50.
YU H L，SUN X D，JIA Z J，et al. A review of GNSS precision timing method in restricted environments[J]. Hydrographic Surveying and Charting，2024，44(2)：46-50.

[4] 徐卓汀，商艳娟，王成群．基于FPGA的IEEE 1588时钟同步系统的设计与实现[J]．智能计算机与应用，2023，13(5)：64-69.
XU Z T，SHANG Y J，WANG C Q. Design and implementation of IEEE 1588 protocol based on FPGA[J]. Intelligent Computer and Applications，2023，13 (5)：64-69.

[5] 单飞桥，王照伟，沈跃．基于精确时间协议的工业无线传感器网络时间同步方法[J]．计算机应用，2023，43(7)：2255-2260.
SHAN F Q，WANG Z W，SHEN Y. Time synchronization method based on precision time protocol in industrial wireless sensor networks[J]. Journal of Computer Applications，2023，43 (7)：2255-2260.

[6] 朱炎平，陆俊，徐志强，等．智能变电站IEEE 1588同步时延优化方法[J]．电力系统自动化，2018，42(12)：148-153.
ZHU Y P，LU J，XU Z Q，et al. Optimization method of IEEE 1588 synchronization time delay for smart substation[J]. Automation of Electric Power Systems，2018，42(12)：148-153.

[7] 冀崇傑，赵刚，王立珂，等．无人战车中时钟同步技术的研究[J]．火力与指挥控制，2024，49(4)：95-100.
JI C J，ZHAO G，WANG L K，et al. Research on clock synchronization technology in unmanned combat vehicles[J]. Fire Control & Command Control，2024，49(4)：95-100.

[8] 景金荣，范正吉，洪应平．基于IEEE 1588精密时钟同步系统的设计[J]．电子器件，2022，45(1)：27-32.
JING J R，FAN Z J，HONG Y P. The design of a precision clock synchronization system based on IEEE 1588[J]. Chinese Journal of Electron Devices，2022，45(1)：27-32.

[9] 赵永成，陈健，张俊杰，等．基于IEEE 1588的多路可调同步时钟板设计与实现[J]．工业控制计算机，2024，37(2)：4-7.
ZHAO Y C，CHEN J，ZHANG J J，et al. High precision adjustable synchronous clock board based on IEEE 1588 protocol[J]. Industrial Control Computer，2024，37(2)：4-7.

[10] 许万，余磊涛，夏瑞东．基于无迹卡尔曼滤波的精密时钟同步算法[J]．湖北工业大学学报，2024，39(1)：8-11，27.
XU W，YU L T，XIA R D. A precision clock synchronization algorithm based on unscented Kalman filtering[J]. Journal of Hubei University of Technology，2024，39(1)：8-11，27.

[11] 杨媛媛，王晓华，武健．一种提高FC网络时间同步精度的方法[J]．中国新通信，2021，23(8)：61-62.

[12] 卢灏，余修武，刘永．基于节点自补偿的IEEE 1588时钟同步算法[J]．传感技术学报，2023，36(1)：53-59.
LU H，YU X W，LIU Y. IEEE 1588 clock synchronization algorithm based on node self-compensation[J]. Chinese Journal of Sensors and Actuators，2023，36(1)：53-59.

[13] 黄广舟．基于FPGA的高精度时钟同步技术的研究与实现[D]．武汉：华中科技大学，2018：8-12.
HUANG G Z. Research and implementation of high-precision clock synchronization technology based on FPGA[D]. Wuhan：Huazhong University of Science and Technology，2018：8-12.

[14] 陶征亮．基于IEEE 1588协议的高精度时钟同步技术研究[D]．赣州：江西理工大学，2022：10-15.
TAO Z L. Research on high-precision clock syn-

chronization technology based on IEEE 1588 protocol[D]. Ganzhou: Jiangxi University of Science and Technology, 2022: 10-15.

[15] 韩一德. 基于IEEE 1588协议的高精度时钟同步系统研究与实现[D]. 太原：中北大学，2021：15-20.
HAN Y D. Research and implementation of high precision clock synchronization system based on IEEE 1588 protocol[D]. Taiyuan: North University of China, 2021: 15-20.

[16] 魏小寒. 基于IEEE 1588协议的时钟同步技术研究[D]. 自贡：四川轻化工大学，2021：16-17.
WEI X H. Research on clock synchronization technology based on IEEE 1588 protocol[D]. Zigong: Sichuan University of Science & Engineering, 2021: 16-17.

[17] 刘畅. IEEE 1588透明时钟原理研究及其应用设计[D]. 哈尔滨：哈尔滨工业大学，2021：19-22.
LIU C. Research on the principles of IEEE 1588 transparent clock and its application design[D]. Harbin: Harbin Institute of Technology, 2021: 19-22.

[18] 王存粮. 分布式系统高精度时钟同步的设计与实现[D]. 太原：中北大学，2023：7-10.
WANG C L. Design and implementation of high precision clock synchronization in distributed systems [D]. Taiyuan: North University of China, 2023: 7-10.

[19] 王泽芳，唐中剑. SM4算法CTR模式的高吞吐率ASIC实现[J]. 电子器件，2019，42(1)：173-177.
WANG Z F, TANG Z J. A high-throughput ASIC implementation of SM4 algorithm CTR mode [J]. Chinese Journal of Electron Devices, 2019, 42(1): 173-177.

[20] 袁天柱. 基于SM4算法的RFID高频数字基带控制器的设计[D]. 武汉：华中科技大学，2017：7-11.
YUAN T Z. Design of the HF RFID digital baseband controller based on SM4 algorithm[D]. Wuhan: Huazhong University of Science and Technology, 2017: 7-11.

[21] 李建立，莫燕南，粟涛，等. 基于国密算法SM2、SM3、SM4的高速混合加密系统硬件设计[J]. 计算机应用研究，2022，39(9)：2818-2825，2831.
LI J L, MO Y N, SU T, et al. Hardware design of high-speed hybrid encryption system based on SM2, SM3 and SM4 algorithm[J]. Application Research of Computers, 2022, 39(9): 2818-2825, 2831.

[22] 窦玉超. SM4算法优化及其密钥扩展算法的设计与实现[D]. 哈尔滨：哈尔滨工业大学，2021：7-11.
DOU Y C. Design and implementation of SM4 algorithm optimization and key extension algorithm [D]. Harbin: Harbin Institute of Technology, 2021: 7-11.

[23] 何诗洋，李晖，李凤华. SM4算法的FPGA优化实现方法[J]. 西安电子科技大学学报，2021，48(3)：155-162.
HE S Y, LI H, LI F H. Optimization and implementation of the SM4 on FPGA[J]. Journal of Xidian University, 2021, 48(3): 155-162.

[24] 王文静，陈青华，张龙云. 国产SM4密码算法的蓝绿激光通信系统设计[J]. 单片机与嵌入式系统应用，2019，19(11)：44-46，52.
WANG W J, CHEN Q H, ZHANG L Y. Design of blue-green laser communication system based on domestic SM4 encryption algorithm[J]. Integrated Circuits and Embedded Systems, 2019, 19(11): 44-46, 52.

[25] 李森. 基于PSOC的测控计算机加密存储卡PL端设计[J]. 火控雷达技术，2024，53(1)：53-60.
LI S. Design of PL side of encrypted memory card for measurement and control computer based on PSOC[J]. Fire Control Radar Technology, 2024, 53(1): 53-60.

（编辑：于福兴）