导读：随着人工智能、大数据的迅速发展，FPGA 在人工智能等领域的应用也受到业界的大力关注。可编程性、高能效比等特点使 FPGA 在人工智能等领域的应用中展现出独特优势。为了促进 FPGA 在人工智能、大数据、边缘计算等新兴应用领域的应用研究和技术推广，推动 FPGA 及人工智能领域的发展，《电子技术应用》杂志于 2021 年第 12 期推出“FPGA 及人工智能”主题专栏，内容涉及 FPGA 在设计验证、通信、医疗等领域的应用。期待对 FPGA 的技术应用提供有益的借鉴。

特约主编：韩德强，高级工程师，研究生导师，现担任北京工业大学信息学部计算机学院实验中心主任，长期从事计算机硬件、嵌入式系统、物联网方面的教学、科研、产品开发等工作。从事教学工作前，曾在企业从事过 12 年的 X86 工控机主板开发、控制工程研发，是国内最早从事嵌入式产品研发的工程技术人员。作为项目负责人承担多项教育部-企业共建项目，并组织、实施过数十种嵌入式产品的开发及控制工程项目的研发。与 Intel、Microsoft、TI 和 Xilinx 等国际知名企业有着深入合作关系。主编译著 4 部、参编教材 1 部，发表论文数十篇，获国家专利、软件著作权 30 余项。曾获“北京优秀青年工程师”荣誉称号，多次获微软全球最有价值专家(MVP)称号。

SoPC 安全启动模型与设计实现

苏振宇，徐　峥，刘雁鸣
(浪潮电子信息产业股份有限公司 安全技术部，山东 济南 250101)

摘　要：针对可编程片上系统(SoPC)在启动过程中面临的固件被篡改、植入恶意代码等安全威胁，提出一种安全启动模型。该模型由 Boot ROM 作为信任根，利用公钥算法和对称密码算法对启动过程中各固件镜像进行数字签名，在 SoPC 上电启动后依次对各固件镜像的签名值进行验证，从而建立起完整的信任链。在实现阶段，利用 Intel 现场可编程门阵列(FPGA)开发平台对模型进行了设计和验证，实现了两种模式的安全启动功能。结果表明，该模型能够应用于实际的芯片开发，满足启动过程安全保护的需求。

关键词： SoPC；安全启动；FPGA；数字签名；固件
中图分类号： TP309　　**文献标识码：** A　　DOI：10.16157/j.issn.0258-7998.212073

中文引用格式： 苏振宇，徐峥，刘雁鸣. SoPC 安全启动模型与设计实现[J].电子技术应用，2021，47(12)：22-25，30.
英文引用格式： Su Zhenyu，Xu Zheng，Liu Yanming. Design and implementation of SoPC secure startup model[J]. Application of Electronic Technique，2021，47(12)：22-25，30.

Design and implementation of SoPC secure startup model

Su Zhenyu，Xu Zheng，Liu Yanming
(Security Technology Department，Inspur Electronic Information Industry Company Limited，Jinan 250101，China)

Abstract： Concerning the security threats such as firmware tampering and malicious code implanting in the startup process of system on a programmable chip (SoPC), a secure startup model is proposed. The model takes boot ROM as the trust root, and the public key algorithm and symmetric cipher algorithm are used to sign the firmware in each phase of the startup process. The digital signature value of each firmware image is verified in turn after SoPC is powered on, so as to establish a complete trust chain. In the implementation stage, the model is designed and verified by using Intel field programmable gate array (FPGA) development platform, and two secure boot modes are realized. The results show that the model can be applied to chip development to meet the requirement of secure startup.

Key words： system on a programmable chip；secure startup；field programmable gate array；digital signature；firmware

0 引言

可编程片上系统(System on a Programmable Chip,SoPC)是一种特殊的嵌入式系统[1],由单个芯片完成整个系统的主要逻辑功能,通过软硬件在系统可编程的功能使得设计方式具备可裁剪、可扩充、可升级等灵活特性。在SoPC启动过程中存在一定的安全风险,恶意软件有可能会修改引导加载程序等固件,使SoPC受到Rootkit攻击[2]。Rootkit等恶意软件通过修改系统的启动过程,安装到系统内以达到持久驻留系统的目的[3],SoPC一旦受到Rootkit等恶意代码感染,即使重新安装系统也无法清除。因此有必要对SoPC进行安全保护,防止在启动过程中固件被恶意篡改。安全启动对于保护设计的知识产权和防止恶意软件在系统上运行至关重要。相关研究工作存在的问题主要有:(1)采用外接可信平台模块(Trusted Platform Module,TPM)[4]实现可信启动[5],该方式增加了硬件成本且系统集成度低;(2)在构建信任链的过程中仅采用杂凑算法进行度量[6],缺少验证的过程,因此安全性较低;(3)嵌入式系统上电时由最先启动的引导加载程序Boot Loader调用TPM对后续加载的模块进行度量[7-8],Boot Loader默认是安全的,但Boot Loader一旦被攻击篡改,整个信任链就处于非可信的状态。

1 SoPC 安全启动模型

基于现有技术的不足,提出一种SoPC安全启动模型,目的是确保SoPC的引导加载程序等固件是可信的,安全启动模型如图1所示。信任根是创建安全启动最关键的部分,能够确保安全级别配置正确并且安全密钥受到保护。在该模型中信任根是SoPC中受信任的第一阶段的引导固件Boot ROM,作为引导SoPC可信的、固有的安全起点。

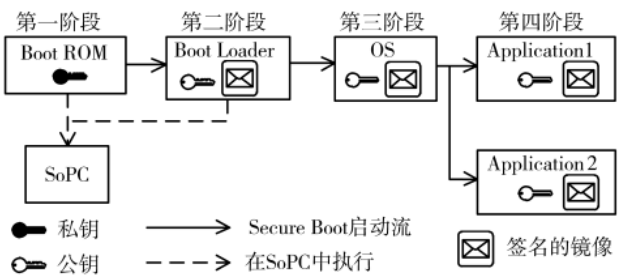

图1 SoPC安全启动模型

为了确保可信性,在安全启动过程中通过建立信任链,使每个固件在加载运行前都经过数字签名的验证,只有当前阶段对后续待启动的模块进行验证通过后,才会加载和执行后续的模块。具体为:依次对第二、三、四阶段的Boot Loader、操作系统和应用程序的镜像文件添加数字签名后存储,在SoPC上电启动建立信任链的过程中,首先由第一阶段的Boot ROM验证第二阶段Boot Loader的签名,确保Boot Loader是受信任的;之后由Boot Loader验证操作系统的签名,依次类推,从而确保各模块是受信任的。为了进一步提高启动过程的安全性,可以对固件镜像加密后再进行数字签名操作,并且将相关的密钥存储于SoPC的安全区域中。

在安全启动之前,对X_i的数字签名SIGi定义如下:

$$SIG_i=D_{SK}[H(X_i)],\quad 1\leqslant i\leqslant n \tag{1}$$

$$SIG_i=D_{SK}[E_K[H(X_i)]],\quad 1\leqslant i\leqslant n \tag{2}$$

其中每阶段待启动的模块为X_i,第一阶段的引导程序X_0为信任根。E代表加密运算,D代表解密运算,H代表杂凑运算,K为对称密码算法的密钥,PK、SK为非对称密码算法的公钥、私钥。SIG_i为数字签名值,$H(X_i)$表示对X_i进行杂凑运算后生成的摘要值。式(1)表示用私钥SK对$H(X_i)$进行数字签名,即对$H(X_i)$做解密运算;为进一步提高安全性,式(2)在进行数字签名之前,利用密钥K对$H(X_i)$进行了对称加密运算。

在启动过程中,由X_{i-1}对X_i进行校验。验证X_i的数字签名SIG_i的定义为:

$$H(X_i)=E_{PK}(SIG_i),\quad 1\leqslant i\leqslant n \tag{3}$$

$$H(X_i)=D_K[E_{PK}(SIG_i)],\quad 1\leqslant i\leqslant n \tag{4}$$

式(3)与式(1)对应,验证签名时利用公钥PK对数字签名值SIG_i进行加密运算后恢复出X_i的标准摘要值$H(X_i)$,之后计算待加载模块X_i镜像的摘要值$H(X_i')$,并与$H(X_i)$进行比对,当$H(X_i')=H(X_i)$时说明X_i未篡改(即$X_i'=X_i$),数字签名验证通过;如果X_i的镜像被篡改为X_i',由于$X_i'\neq X_i$,故$H(X_i')\neq H(X_i)$,校验不通过。式(4)与式(2)对应,先利用公钥PK对SIG_i进行加密运算,再利用K进行解密运算后恢复出标准摘要值$H(X_i)$。

以信任根X_0为信任起点逐级进行验证,如果每个模块X_i的数字签名都验证通过,则启动过程是安全可信的,最终构建一个安全可信的信任链:$X_0\rightarrow X_1\rightarrow X_2\cdots\rightarrow X_i$,如果在启动过程中任意$X_i$的数字签名验证不通过,则信任链的建立终止,系统不会启动。

2 模型设计

2.1 SoPC 硬件架构设计

安全启动模型采用了Intel Arria10现场可编程门阵列(Field Programmable Gate Array,FPGA)开发平台[9]进行设计和验证,EDA工具采用的是Quartus II 19.1[10]、嵌入式开发套件为EDS[11]、操作系统为Linux Ubuntu 14.0。SoPC安全启动硬件架构设计如图2所示,包括硬件处理器系统(Hard Processor System,HPS)和FPGA可编程逻辑区域,其中HPS包括微控制器MPU、安全管理器(Security Manager,SM)、Boot ROM、On-chip RAM等;FPGA区域包括配置逻辑(Configuration Logic,CL)、安全熔丝Fuse、Flash/SD卡等。

主要功能模块的作用如下:

(1)SM:用于系统初始化和引导,在SoPC上电复位后根据熔丝寄存器(Fuse_REG)对系统进行配置。

(2)Boot ROM:管理安全引导过程的第一阶段,并对

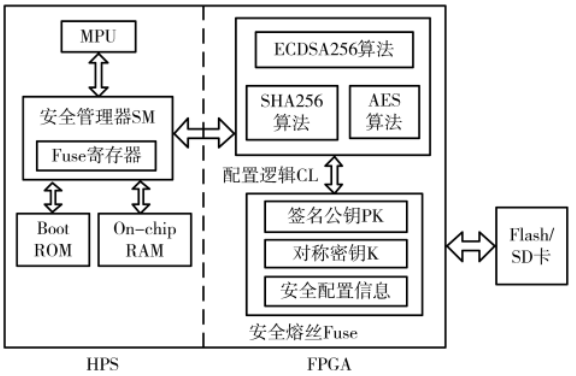

图 2 SoPC 安全启动硬件架构

第二阶段的引导程序 Boot Loader 进行验证。

(3)CL：发送配置信息给 SM，另外包括 256 位椭圆曲线数字签名算法(Elliptic Curve Digital Signature Algorithm，ECDSA-256)[12]、256 位安全散列算法(Secure Hash Algorithm，SHA-256)[13]和高级加密标准(Advanced Encryption Standard，AES)[14]，作为对镜像加密和签名的密码算法模块。

(4)Fuse：一次性可编程区域，作为安全存储区存储公钥 PK、对称密钥 K 以及系统安全配置信息。

(5)Flash/SD 卡：为片外非易失存储器，存储经过加密和签名的 Boot Loader、OS 等镜像文件，启动过程中镜像验证通过后才能加载到 SoPC 片内 RAM 运行。

SoPC 上电复位后，CL 初始化并将 Fuse 安全配置信息发送到 HPS 中的 SM，配置信息保存在 SM 中的 Fuse_REG 寄存器中。SM 根据 Fuse_REG 中的配置信息执行安全检查并发送系统初始化信号，控制 Fuse 信息自动发送到时钟管理器，内存控制 Fuse 信息自动发送到复位管理器，认证、加密、公钥等配置信息存储在相应的存储器映射位置，由 Boot ROM 代码读取。

之后 HPS 在安全状态下退出复位状态，Boot ROM 开始执行。此时，HPS 处于受信任状态，并且保证 Boot ROM 代码按预期执行。Boot ROM 读取安全报头，如图 3 所示，确定根密钥的存储位置并验证第二阶段 Boot Loader 镜像安全报头中的数字签名值。如果校验通过，Boot ROM 允许 Boot Loader 镜像加载和执行。

2.2 镜像签名

ECDSA-256 算法的私钥用于对镜像进行签名，公钥用于对镜像进行验证。公钥可通过 Fuse_REG 寄存器选择配置为 3 种类型，如表 1 所示。

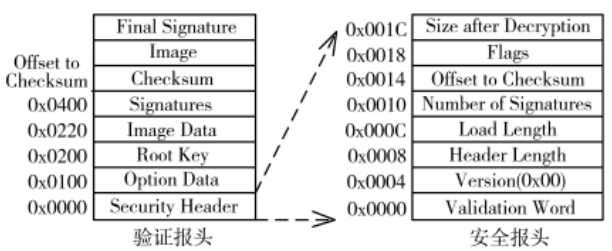

图 3 Boot Loader 镜像的验证报头和安全报头

表 1 公钥类型

类型	存储位置	描述
安全密钥	Fuse	公钥经过 SHA-256 运算后存储于 Fuse
FPGA 密钥	FPGA	公钥为生成的比特流，存储于 FPGA 中
测试密钥	文件	公钥存储于镜像文件，不存储于 SoPC 中

(1)安全密钥：安全性最高，公钥 PK 经过 SHA-256 杂凑运算后的摘要值 $H(PK)$存储于 Fuse 中，使用前先验证公钥完整性，校验通过才允许加载；若公钥被篡改为 PK'，则 $H(PK')\neq H(PK)$，禁止公钥加载。

(2)FPGA 密钥：安全性较高，公钥直接存储于 FPGA 区域的 RAM 中，未经过杂凑运算。

(3)测试密钥：安全性最差，公钥直接存储于镜像文件的报头中，仅在测试阶段使用。

3 模型验证

3.1 安全镜像文件创建

镜像文件有两种创建方式，对应两种安全级别，可通过 Fuse_REG 寄存器进行选择配置。

(1)签名验证方式

该方式首先需要利用 OpenSSL[15]生成签名密钥对，OpenSSL 是一个支持安全套接字协议 SSL 的开源工具包，集成于 EDS 嵌入式命令 shell 中。生成密钥对的具体方式是在 Linux 中启动 Boot Loader 生成器后调用 OpenSSL，运行如下命令使 OpenSSL 生成密钥对：# openssl ecparam -genkey -name prime256v1 -out root_key.pem。

创建的签名密钥对存储于 root_key.pem 文件中，使用安全引导镜像工具 alt-secure-boot 对镜像进行签名的命令为：# alt-secure-boot sign-i Image.bin-o Image_sign.bin-t user。

其中 user 代表的公钥类型是测试密钥，在测试阶段的安全启动过程中，直接读取镜像文件中的公钥，不对公钥进行 SHA-256 杂凑校验比对。Image.bin 和 Image_sign.bin 分别为签名之前和添加数字签名之后镜像文件，将 Image_sign.bin 文件存储于片外 SD 卡中。

(2)加密+签名验证方式

为了在引导期间提供最高级别的安全性，可以对镜像文件加密后再添加数字签名，以便在校验阶段提供双重保护。在 Boot Loader 生成器中通过文件 encrypt.key 生成加密密钥 key，用于对镜像加密。具体为 EDS 工具调用 Boot Loader 生成器读取 encrypt.key 文件中的二进制位流生成加密密钥 key。对镜像进行加密的命令为：# alt-secure-boot encrypt-i Image.bin-o Image_enc.bin-k key。

其中 key 为 AES 对称算法的密钥，Image.bin 和 Image_enc.bin 分别为加密之前和加密之后镜像文件。

3.2 安全启动功能测试验证

验证过程如图 4 所示，具体说明如下。启动过程的每个阶段利用公钥 PK 和杂凑算法校验下一阶段镜像的数字签名，首先恢复出镜像文件的标准摘要值 $H(X_i)$，如果镜像被加密了，根据式(3)需要利用 PK 和对称密钥 K 恢复出标准摘要值 $H(X_i)$。在镜像加载之前计算其摘要值 $H(X_i')$，并与 $H(X_i)$进行比对，如果 $H(X_i)=H(X_i')$，说明镜像未篡改，即 $X_i=X_i'$，允许镜像加载执行；如果 $H(X_i)\neq H(X_i')$，说明镜像 X_i 已被篡改为X_i'，信任链为非可信状态，启动过程失败。例如在引导过程中，Boot ROM 首先验证 Boot Loader 镜像的数字签名，验证通过后才能加载运行，之后由 Boot Loader 对后续阶段的 Linux 系统镜像进行验证和加载。如果验证数字签名失败，启动过程终止。

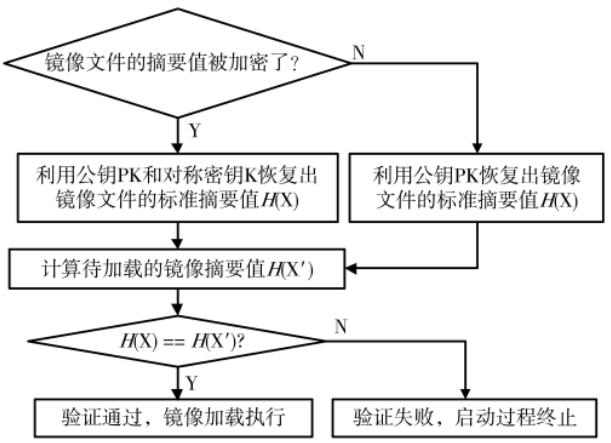

图 4 安全启动验证流程图

表 2 为 Boot Loader 和 Linux 的镜像文件经过杂凑运算后的摘要值。为了验证安全启动的功能，对原始值的前 2 个字节分别篡改为了“0x12,0x34”。

表 3 为两种安全启动方式的测试结果，如果镜像未篡改，SoPC 正常启动；如果对 Boot Loader 或 Linux 系统镜像文件进行了篡改，信任链建立终止，启动失败。

图 5 所示为启动方式的启动时间对比，正常启动时因未对镜像进行签名和验证，所以启动时间最短，但安全性最差；若采用签名验证方式启动，从 SoPC 上电到应用程序加载的时间总共为 15 s；采用签名+加密验证方式的整个启动时间最长，为 25 s，但安全性最高。

4 结论

SoPC 在启动过程中通过校验启动阶段各固件镜像的数字签名方式提升了安全性，保证了启动过程中引导加载程序、操作系统及应用程序的安全可信，从而构建起完整的信任链。此外，测试阶段为了避免把密钥永久写入一次性可编程区域而造成 Fuse 资源浪费，灵活采用了把密钥写入文件和 FPGA RAM 区域的方式进行功能的验证。后续工作可以将该安全启动模型应用于实际 SoPC 的设计开发过程，实现对 SoPC 启动过程的安全保护，并且将密钥固化在安全区域以防止被非法篡改，提高安全性。

表 3 安全启动测试结果

启动方式	镜像未篡改	篡改 Boot Loader	篡改 Linux 系统
签名验证	启动成功	启动失败	启动失败
加密+签名	启动成功	启动失败	启动失败

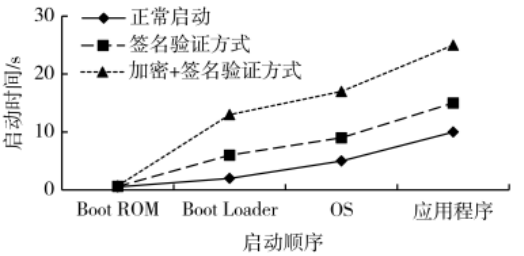

图 5 安全启动模式的启动时间对比

参考文献

[1] 李海生，张凡.基于 SOPC 的脉冲信号参数测量系统设计[J].电子技术应用，2020，46(7)：84-87.

[2] 张瑜.Rootkit 隐遁攻击技术及其防范[M].北京：电子工业出版社，2017.

[3] 杨章象，代祖华，王博.基于 Kprobe 的 Rootkit 检测机制[J].计算机工程与应用，2016，52(7)：127-137.

[4] 冯登国.可信计算-理论与实践[M].北京：清华大学出版社，2013.

[5] 王希冀，张功萱，郭子恒.基于可信密码模块的 SoC 可信启动框架模型[J].计算机工程与科学，2019，41(4)：606-611.

[6] 罗钧，蒋敬旗，闵志盛，等.基于 SHA-1 模块的可信嵌入式系统安全启动方法[J].山东大学学报(理学版)，2012，47(9)：1-6.

[7] 徐明迪，张帆.可信计算技术在嵌入式操作系统中的应用[J].武汉大学学报(理学版)，2014，60(3)：237-241.

[8] 苏培培，刘宝明.基于国产处理器的可信系统研究与实现[J].电子技术应用，2012，38(1)：136-138.

表 2 镜像文件摘要值篡改

镜像类型		镜像文件摘要值及篡改数据(Hex)
Boot Loader 镜像文件 (u-boot-dtb.bin)	原始数值	74,D2,48,94,7C,3C,CA,45,B0,20,01,6B,91,2A,F3,2E,BA,B8,48,61,C1,BE,55,BE,11,44,E4,B7,69,2B,50,2B
	篡改数值	12,34,48,94,7C,3C,CA,45,B0,20,01,6B,91,2A,F3,2E,BA,B8,48,61,C1,BE,55,BE,11,44,E4,B7,69,2B,50,2B
Linux 镜像文件 (zImage.bin)	原始数值	D6,93,6C,C0,C6,A6,52,0D,8B,D5,D6,49,0E,28,58,6A,87,00,6B,B7,6A,F6,FE,D6,71,DB,A3,32,3B,CB,07,11
	篡改数值	12,34,6C,C0,C6,A6,52,0D,8B,D5,D6,49,0E,28,58,6A,87,00,6B,B7,6A,F6,FE,D6,71,DB,A3,32,3B,CB,07,11

(下转第 30 页)

采用传统的下变频滤波的方式，每一路数字混频滤波大概需要 67 个 DSP，则高速跳频信号的 51 个频点总共需要 3 417 个 DSP，此时一片 XC7Z045 的 DSP 资源无法完成此工作。

7 结论

针对高速跳频信号的侦察由于其信号带宽宽、跳速快、编码体制复杂，对接收机的设计及后端数字信号处理都是极大的考验[7,12-13]。本文探讨了针对高速跳频信号接收的宽带接收设计，结合当前高性能的射频频率捷变芯片，两片 ADRV9009 的四个接收通道可实现高速跳频信号带宽的全部覆盖。该方法不仅简化了前端射频部分的功耗、体积，而且前端四路接收通道只需设置对应频点即可。后端采用多相滤波的数字信道化技术，降低了数字信号处理的速度，节省了硬件资源，增强了系统的灵活性。该处理技术适应于不同带宽的高速跳频信号，而且针对其他非协作通信的宽带信号依然有较大的优势。

参考文献

[1] 蒋鸿宇，叶江峰，肖仕伟，等.一种超宽带高速跳频信号实时非合作接收机[J].信息与电子工程，2012，10(4)：390-395.

[2] 李宏彦.Link-16 数据链波形分析与 FPGA 工程实现[D].成都：国防科技大学，2016：17-20.

[3] 刘军，简义全.基于 LTC5586 的 1 GHz 实时带宽接收机设计[J].电子世界，2018(11)：145-146.

[4] 江岩，詹建，钱时祥.监测接收机高速扫描速度检测方法探讨[J].国外电子测量技术，2016，35(3)：85-88.

[5] 刘昕卓，米胜男，曲志昱，等.宽带数字信道化接收机算法研究与硬件实现[J].航空兵器，2017(1)：68-73.

[6] 鲁艳.一种宽带数字信道化接收机的设计及实现[J]电子技术与软件工程，2018(4)：87.

[7] 唐济远，刘渝，袁春珊.多相滤波结构的信道化接收机设计[J].军事通信技术，2012(33)：57-62.

[8] 杨小牛，楼才义，徐建良.软件无线电原理与应用[M].北京：电子工业出版社，2001：21- 87.

[9] 侯聪.多相滤波数字信道化的 FPGA 实现[J].电讯技术，2012(8)：1345-1348.

[10] 叶金伟，刘渝.基于多相滤波的跨信道宽带信号处理技术[J].航天电子对抗，2011(8)：52-55.

[11] 胡广书.数字信号处理(理论、算法与实现)[M].北京：清华大学出版社，2004：179-191.

[12] 黄卫英.一种 Link16 信号的检测识别算法[J].电讯技术，2021，61(2)：186-190.

[13] 齐小辉，卢丹.基于多相滤波器的信道化接收机设计[J].军事通信技术，2012(3)：57-62.

[14] 宗孔德.多抽样率信号处理[M].北京：清华大学出版社，1996：150-164.

[15] 陈涛，岳玮，刘颜琼，等.宽带数字信道化接收机部分信道重构技术[J].哈尔滨工程大学学报，2011(12)：1610-1616.

(收稿日期：2021-09-24)

作者简介：

王杰(1984-)，男，硕士，工程师，主要研究方向：数字信号处理。

扫码下载电子文档

(上接第 25 页)

[9] Intel Corporation.Intel Arria 10 hard processor system technical reference manual[EB/OL].(2021-07-xx)[2021-08-19].https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/arria-10/a10_5v4.pdf.

[10] Intel Corporation.Intel quartus prime standard edition user guide[EB/OL].(2019-12-xx)[2021-08-19].https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug-qps-getting-started.pdf.

[11] Intel Corporation.Intel SoC FPGA embedded development suite user guide[EB/OL].(2021-03-xx)[2021-08-19].https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_soc_eds.pdf.

[12] National Institute of Standards and Technology.FIPS PUB 186-5(Draft)：digital signature standard(DSS)[EB/OL].(2019-10-xx)[2021-08-19].https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5-draft.pdf.

[13] National Institute of Standards and Technology.FIPS PUB 180-4：secure hash standard(SHS)[EB/OL].(2015-08-10)[2021-08-19].https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

[14] National Institute of Standards and Technology.FIPS PUB 197：announcing the advanced encryption standard(AES)[EB/OL].(2001-11-26)[2021-08-19].https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf.

[15] VIEGA J，MESSIER M，CHANDRA P.Network security with OpenSSL[M].United States of America：O'Reilly & Associates，Inc.，2002.

(收稿日期：2021-08-19)

作者简介：

苏振宇(1983-)，通信作者，男，硕士，高级工程师，主要研究方向：信息安全、嵌入式系统，E-mail：suzhy@inspur.com。

徐峥(1982-)，男，系统架构师，主要研究方向：信息安全、计算机系统架构。

刘雁鸣(1989-)，男，硕士，工程师，主要研究方向：安全测评、渗透测试。

扫码下载电子文档

版权声明