机车电传动 2018 年第 5 期 (2018-09-10)
ELECTRIC DRIVE FOR LOCOMOTIVES №5, 2018(Sep. 10, 2018)

文章编号：1000–128X(2018)05–0056–05

嵌入式控制软件保密性设计研究与应用

戴计生[1]，王成杰[2]，李　益[2]，陈俊波[1]，李　程[2]

（1. 株洲中车时代电气股份有限公司，湖南 株洲　412001；
2. 中车株洲所电气技术与材料工程研究院，湖南 株洲　412001）

摘　要：分析了当前嵌入式控制软件所面临的主要安全威胁：克隆、反向工程、篡改，从安全认证芯片、程序加解密与完整性验证、安全处理器 3 个方面，提出了针对嵌入式控制软件的保密设计策略及其安全性分析，设计了一种基于安全认证芯片、对称和非对称加密算法、安全密钥存储器的嵌入式控制软件保密应用解决方案，可实现对软件的机密性、完整性和抗否认性保护，为嵌入式控制软件保密性设计提供参考。

关键词：加密；安全认证；嵌入式控制软件；保密设计

中图分类号：TP311.5　　**文献标识码：**A

doi：10.13890/j.issn.1000-128x.2018.05.013

Research and Application of Embedded Control Software Security Design

DAI Jisheng[1], WANG Chengjie[2], LI Yi[2], CHEN Junbo[1], LI Cheng[2]

(1. Zhuzhou CRRC Times Electric Co., Ltd., Zhuzhou, Hunan 412001, China;
2. CRRC ZIC Research Institute of Electrical Technology and Material Engineering, Zhuzhou, Hunan 412001, China)

Abstract: Security threats faced by embedded control software were analyzed: cloning, reverse engineering, and tampering. The embedded control software security design strategies and security analysis were presented from three aspects of security authentication chip, program encryption and decryption, integrity verification, and security processor. Finally, based on security authentication chip, symmetric and asymmetric encryption algorithm and embedded control softwarre for secure key memory, a security application solution was designed, which could protect the confidentiality, integrity and non-denied of the software. It has great reference value for the embedded controller design.

Keywords: encryption; security authentication; embedded control software; security design

0　引言

当前嵌入式软件所面临的主要安全威胁有克隆、反向工程、篡改等。克隆是指通过复制获得 CPU 软件的相同副本应用于其他 CPU，反向工程是指通过各种手段侦探软件设计信息以重建目标软件，篡改是对存储在器件中的软件进行修改替换以使系统出现故障或者盗取内部敏感数据。其中，克隆的技术最为成熟，应用也比较广泛，对嵌入式软件的安全造成的威胁也最大。

对嵌入式控制软件攻击的主要目的是窃取系统存储的代码或私密数据，或者简单地让系统不能正常工作，或者获取系统的控制权限等方式进行恶意操作，破坏系统，阻止其提供服务。主要攻击手段有盗取拷贝 Flash 芯片，通过逻辑分析仪探测总线通信，利用 JTAG 调试接口或插入流氓程序读取拷贝处理器内部存储区，插入或替换原有的软件对嵌入式设备进行恶意破坏或盗取系统内部的敏感信息等。

随着当前攻击者所用的技术手段越来越先进，嵌入式控制软件所面临的安全威胁也越来越大[1]。软件被窃取或反向工程将会造成核心知识产权的流失，软件被恶意攻击和篡改会造成整个系统的崩溃，甚至会

收稿日期：2017–08–30；修回日期：2017–11–07

导致重大安全事故。为保护设计人员的知识产权，防止被克隆、反向工程和篡改等方式攻击，嵌入式软件的保密性设计已经显得越来越重要，引起了业界的广泛关注，具有非常重要的意义。

1　嵌入式保密性设计准则

嵌入式控制软件安全保护可分为机密性保护、完整性保护及可鉴别性保护。数据机密功能保护软件或敏感数据不被非法窃听者获取。数据完整性是指数据未经授权不能进行改变的特性，相互认证是确保一个消息的来源的确是他们所宣称的实体，而不是假冒者。密码学是信息安全的基础，信息安全要求的机密性、完整性和抗否认性都是依赖于密码算法。通过加密可以保护信息的机密性，通过消息摘要可以验证信息的完整性，通过数字签名可以保护信息的抗否认性。

为保护嵌入式系统知识产权，防止克隆抄板、反向工程或攻击者的恶意破坏，同时具备主动防攻击自毁灭功能，保密设计需考虑如下设计准则：

①软件的鉴别认证。所有的软件开始运行前或运行中，需鉴别其是否运行在合法的处理器单元上，这就需要设计者给每个处理器分配一个可用于认证的 ID，其 ID 要么是唯一的，要么是保密的，其他克隆抄板者不能获得这个 ID，同时需设计一个安全的认证算法，使攻击者无法根据输入条件反推出认证 ID。

②软件或数据的加密及完整性验证。数据加密保证其在总线上传输或存储数据的机密性，减少数据被窃取的可能性，即使被恶意窥探，在机密性保护下也无法得到数据的真实信息，避免攻击者直接读取片外存储来窃取信息。在攻击者恶意篡改了总线或存储数据时，要求处理器必须探测到，这就需要引入完整性验证机制，如果不验证直接处理执行，那些篡改后的数据将被处理器当作合法的数据执行，可能造成运行崩溃，如果使用重放攻击，则可能造成错误的运行结果。

③隔离安全信息和敏感信息的时效性。在处理器内部的安全信息应当与外部普通数据隔离开来，对于安全信息，以及处理安全信息的功能模块，应尽可能地放到信任区域，如集成到处理器芯片或安全存储器上。存放在安全处理器中的敏感信息（如密钥）应当设置一个时效性，避免长期存放导致暴力破解有机可乘。处理器的密钥应定期更换。

④封锁安全“后门”。嵌入式软件的“后门”主要是 JTAG 等调试接口，攻击者很容易通过该接口对程序进行拷贝或反向工程，因此需禁止调试接口或其部分特性以封锁通过它的安全攻击[2]。

以上的这些安全准则用于保护嵌入式系统的知识产权，能有效防止攻击者发动总线窥探和篡改之类的系统级攻击，增强嵌入式系统的安全性。

2　保密性设计策略研究

基于以上原则，总结了几种嵌入式控制软件保密性设计策略并分别进行安全性分析。

2.1　外置安全认证芯片保护嵌入式 IP

2.1.1　设计策略

使用安全认证进行产品身份识别可保护产品免受外部黑客的攻击，同时也要确保产品不会被克隆硬件的方法从内部攻陷，即防抄板设计。认证是指 2 个或多个实体之间建立身份认可的过程，单向认证情况下一方需向另一方证明其身份的合法性，双向认证即双方需要彼此向对方证明自己的身份。基于密钥的认证过程即将密钥和需要认证的数据（信息）作为输入，来计算信息认证码（MAC），为证明 MAC 发送方的合法身份，接收方可产生一个随机数作为挑战码回送给发送方。MAC 发送方必须根据密钥、信息和质询码重新计算新的 MAC[3]，并返回给接收方。如果对应任何挑战码发送方都可产生有效的 MAC，则可以确认发送方是知道密钥的，其身份是合法的。挑战 – 应答认证过程可防止重放攻击，解决信息合法性，克服 MAC 模型的弱点。挑战 – 应答认证过程如图 1 所示。

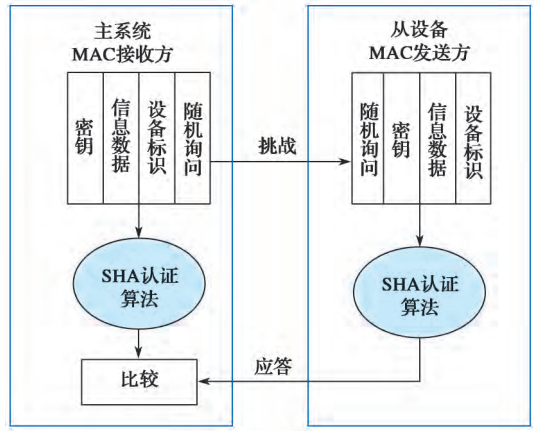

图 1　基于密钥的挑战 – 应答认证

SHA 认证算法具有不可逆、抗冲突、高雪崩效应和易实现等特征，很多芯片制造商如 Maxim、Atmel 都提供了基于 SHA 的挑战 – 应答认证算法的 IC。安全认证芯片通过对密钥的安全存储和 SHA、RSA/ECDSA 等认证算法[4]，可实现软件代码（IP）保护、软件授权和升级管理、配件识别和电子标签、数据传输和文件的加解密等功能。

①通过主机认证板上待认证器件的身份合法性，来决定软件是否要正常执行。由于非法用户无法复制认证器件，以此验证保护软件代码被非法拷贝，从而防止设备被非法抄板。

②通过对认证器件的身份验证和 EEPROM 存储器数据的解析，进行软件授权功能控制。在软件升级时，先进行身份识别，以决定设备的合法性和加载哪种版本的软件，也可应用于版权管理，控制授权客户的出

货量。

③利用 SHA 算法的结果 MAC 码作为数据传输或文件的加 / 解密的密码，进行加解密处理；密码中使用器件唯一的 ID 信息，因此服务器端面向某个终端设备产生的加密数据文件，只能由该终端设备产生的密码解密，从而防止了设备之间相互拷贝数据。

2.1.2 安全性分析

此种方案只需外加一颗带有密钥的小芯片，方案简单实用，成本很低，适用于任何高低性能的嵌入式控制平台场合。但方案只进行了对身份的真伪识别，这种安全措施存在漏洞，攻击者可以对软件进行逆向工程然后将认证部分的代码去掉，很多支持仿真调试的微处理器其内部代码可以很容易读取。

2.2 程序加解密与完整性验证

2.2.1 设计策略

使用安全认证芯片可有效地防克隆抄板，但是由于程序代码保存在外部非易失存储器上，通过外部存储器和总线攻击等手段，黑客可轻易获取程序映像文件并对其进行反向工程。为防止对映像文件的反向工程，可使用安全密钥与 AES 算法对嵌入式程序文件进行加密，在信息的机密性保护方面，流行的加密算法有 RSA、ECC、AES、3DES 等，其中 RSA 和 ECC 是公钥密码算法，AES 和 3DES 是对称加密算法。RSA 加解密使用公钥 / 私钥与仅用私钥相比具有更显著的安全优势，但是 AES 加解密算法效率更高 [5]，因此一般使用 AES 用于信息加密，RSA 用于数字认证 [6]。但是如果仅仅使用加密密文还是有可能受到篡改攻击，除了保护信息的机密性外，为了验证消息的完整性，需使用消息认证码（MAC）和散列函数对消息进行完整性验证，结合 MAC 和 Hash 函数，使用 HMAC 认证验证程序完整性，可有效地抵抗各种密文攻击 [7]。

程序加解密和信息完整性验证设计框图如图 2 所示，方案中安全启动基于 BootLoader+Image 的加载机制，首先使用 HMAC 密钥对 Image 产生认证标识，采用 AES 加解密算法对 Image、HMAC 密钥和认证标识加密完成对 Image 的加密，利用片内的安全存储区或单独的安全存储芯片存放 AES 算法密钥，加密映像存储在片外存储器中，启动时将密文加载入内存，然后在 BootLoader 启动过程中完成认证，认证通过则在该过程中使用 AES 解密算法解密 Image，并使用 HMAC 对解密 Image 进行完整性认证，认证通过则运行 Image。

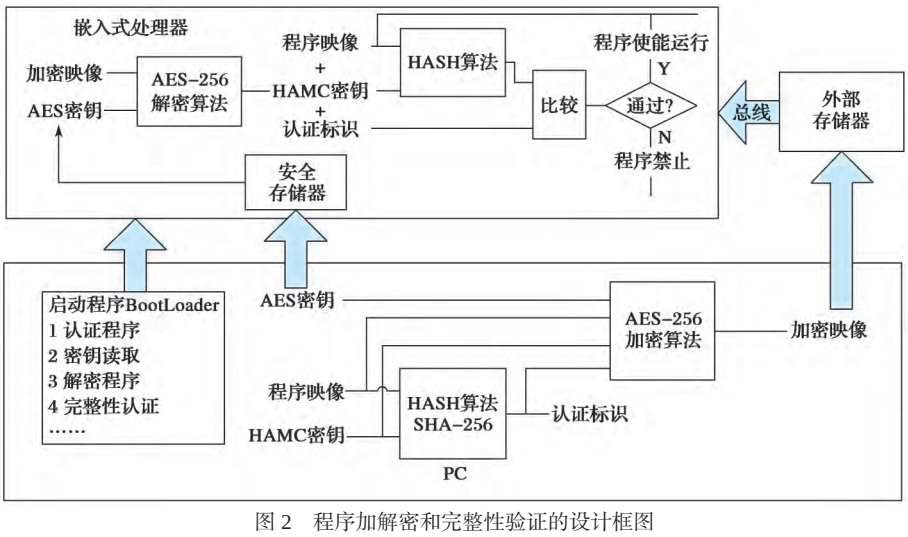

图 2 程序加解密和完整性验证的设计框图

此方案需开发一套加密的上位机软件，嵌入式加解密算法可通过本身软件实现也可外带专用的加解密芯片完成。需注意的是，任何安全方案必须采用标准的加密算法，例如用于对称加密的 AES、Triple DES，用于公钥加密的 RSA、ECDSA。因此在这些加密系统中，最宝贵的资产是密钥。在标准处理器的软件中实施加密时，密钥储存在通用系统存储器中，就很容易通过恶意软件、调试端口（JTAG）或物理攻击而得到。使用安全存储器将大大减少这些弱点，采用安全 IC 集成保护技术，集成了金属屏蔽、环境传感器和外部网状传感器等，防止被物理篡改。

2.2.2 安全性分析

此种方案有效地实现了对程序和数据的机密性、完整性和真伪鉴别保护。但是仍存在以下问题需要解决：①密钥的安全储存。需要考虑专用安全密钥芯片来存储密钥信息。②加密算法的计算时间。因为加密算法尤其是公钥算法（RSA、ECDSA）等都比较复杂需占用较多的计算资源，尤其对于低性能处理器是无法承受的，加密算法的引入会占用一部分处理器的启动时间。③处理器的安全启动，如果处理器不支持安全启动。其启动代码只能以明码的形式加载，就存在攻击者对启动代码的反向工程攻击；④处理器调试接口的安全封锁，如果处理器不支持对其调试接口的安全封锁，攻击者可以通过调试接口“后门”对运行的程序进行篡改和反向攻击。

2.3 安全处理器

2.3.1 设计策略

对嵌入式软件的保密最好的安全措施应该是利用安全处理器 [8]。安全处理器集成了逻辑保护、支持安全启动和存储器管理单元 (MMU) 防止恶意软件注入，可禁用 JTAG 端口，并且处理器的硬件保护措施对最为关键的加密密钥进行保护，这是安全系统的安全关键。由于解密后的指令存放在芯片内部缓冲中，所以代码加密不会降低应用程序的运行速度。目前越来越多的芯片厂家都提供了安全处理器，如 Xilinx 的新一代 Zynq 或 FPGA 都具备了安全保密功能，MAXIM 公

司的安全处理器还包括了自毁输入、顶部蛇形网纹（监测外部微探针攻击）、温度 / 电压传感、硬件加密加速器、无外部时钟、随机加入虚假总线周期、标准加密算法（3DES、RSA）、独特的锁存位和 BootLoader 等[9]，其安全性等级更高。图 3 所示为 MAX32590 安全处理器的设计原理。

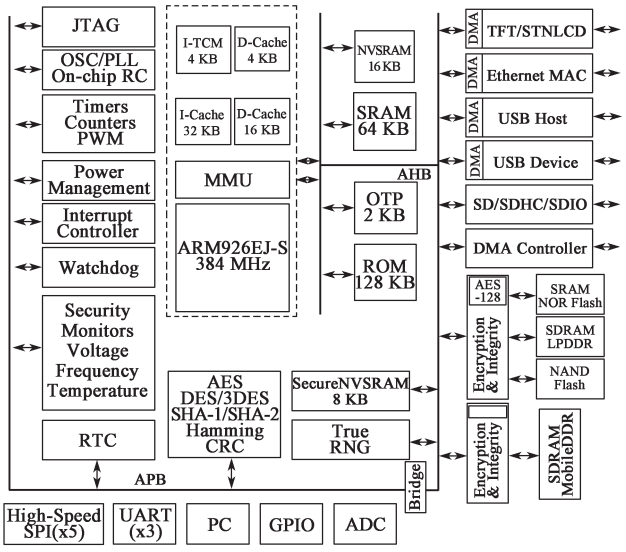

图 3　MAX32590 安全处理器

2.3.2　安全性分析

此方案是实现对嵌入式软件安全保密的最佳方案，但市面上真正的安全处理器可选种类较少，Xilinx 的最新一代的 Zynq 和 FPGA 尽管支持部分的安全功能，但是离 MAXIM 真正的安全处理器还有一定差距，需要使用加密软件等方式来弥补。对于安全性要求特别高并且计算性能适中的应用场合建议选用安全处理器。

3　保密性设计应用

综合应用上述嵌入式控制软件保密性设计策略，由于所用的 CPU 内部没有安全相关的处理模块，要保证其软件安全，需在原有的系统结构上加入可信的安全结构或者加入安全硬件的方法从系统体系结构上进行安全保护。优化设计采用以下方式：

①为防止软件抄板克隆，可使用外部安全认证芯片如 DS28E15[10]，只有通过认证鉴别后软件才可正常运行。

②为保证程序的机密性，防范 Flash 存储拷贝或系统总线的侦测攻击，需要同时结合 PC 机和 CPU 内部软件对程序和参数文件进行 AES 加解密和 HMAC 数据完整性验证。

③为保证密钥等关键数据的存储安全，可使用外部安全管理器如 MAX36025（DS28E15 也可作为密钥安全存储器）存储解密密钥。该安全存储器能主动监测各种物理工具，具备防篡改自毁灭功能，同时内部还集成实时时钟、温度检测和硬件看门狗等。

④为分散程序的泄密分析，使用“BootLoader+ 加密 Image+ 加密参数表”的程序架构，使得一套完整的控制算法分块存储在不同的存储器，任何一个算法和参数文件不能获取或无法解密，都不能获得整个软件。

整体方案框图如图 4 所示。

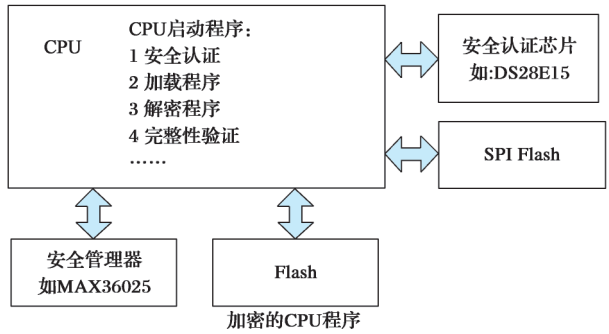

图 4　控制平台保密设计整体框图

图 4 所示的整体设计框图中，通过外置安全认证芯片的 SHA、RSA/ECDSA 等认证算法对嵌入式软件进行授权管理，实现对软件代码（IP）保护。在安全管理器存储关键数据，包括密钥等，利用 HMAC、AES、RSA 算法保护程序文件的机密性和完整性。使用单独的 SPI Flash 存储安全的 DSP BootLoader，BootLoader 需对其自身进行身份认证，如果认证不通过将无法启动，同时它也是解密和验证主算法 Image、安全加载和运行 DSP 算法的引导程序；主算法 Image 和参数表在发布前必须使用专用的 PC 软件加上 HMAC 消息摘要和 AES 加密，安全密钥存储在安全存储器中，保证了主算法 Image 和参数表的机密性；安全密钥存储在具有反篡改自毁灭功能的安全管理器芯片中，攻击者无法拷贝安全管理器中的密钥。

3.1　PC 端软件加密步骤

PC 端加密软件的加密基本步骤如图 5 所示。在进行加密前，随机生成一个密钥，利用 HMAC 算法对程序明文生成 HMAC 消息认证码，然后使用 AES 算法对文件进行加密，再用 RSA 算法对 AES 加密后的密文进行数字签名。具体步骤如下：

①加密软件生成随机 HMAC 密钥，并采用 HMAC 算法对明文（即被加密程序）生成消息摘要，并将 HMAC 密钥及消息摘要写入到文件中；其中随机 HMAC 的密钥格式为时间 + 随机数。

②采用 AES 加密算法对明文和 HMAC 密钥、摘要进行整体加密，生成程序密文。

③利用 RSA 算法对程序密文进行数字签名，并将签名写入到文件中；在文件起始位置写入文件头。文件头主要包含以下内容：a. 加密类型，如是否采用 HMAC、AES、RSA 算法中其中一种或几种；b.HMAC

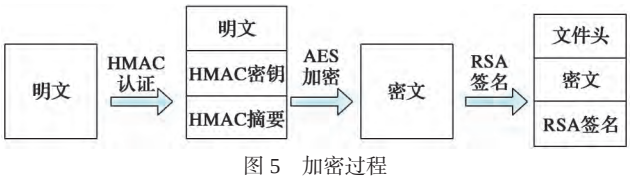

图 5　加密过程

密钥在文件中位置偏移及字节长度；c.HMAC 密钥在文件中位置偏移及字节长度；d.RSA 签名在文件中位置偏移及字节长度。

加密后的文件主要包含 3 部分：①文件头；②采用 AES 对目标码以及 HMAC 密钥、摘要进行整体加密后的密文；③采用 RSA 对 AES 加密的密文进行签名。文件加密格式如图 6 所示。

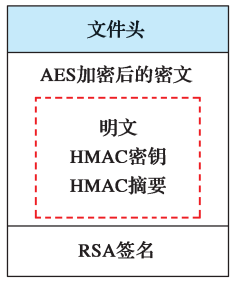

图 6　加密文件格式

3.2　嵌入式系统认证解密过程

首先，利用外部安全芯片进行认证对嵌入式软件进行授权管理，授权认证通过后对加密 Image 进行解密认证。Image 解密过程如图 7 所示。其次从文件头中获取文件加密类型，如果进行了 RSA 签名，则对密文进行 RSA 签名认证，然后获取 AES 密钥进行解密，最后获取 HMAC 密钥利用 HMAC 算法对解密后的明文进行认证，一次完整的解密步骤如下：

①利用外部安全芯片进行授权认证，授权认证通过后才进行后续步骤。认证器件 DS28E15 装载了身份信息，通过对身份信息的认证来决定是否执行后续解密步骤，外部安全芯片的认证有效防止抄板克隆。

②从文件头中获取加密类型。获取加密类型时为了支持上位机加密方式多样性，通过对加密类型的识别，来选择解密算法。不同解密算法的执行时间不一样，可以根据 CPU 运算能力及系统对解密时间的要求选择不同的加解密算法。

③判断加密类型是否有 RSA 签名。如果是，利用 RSA 进行密文签名认证。RSA 认证能够有效鉴别程序的身份。

④判断加密类型是否有 AES 加密。如果是，从密钥安全管理器获取 AES 加密密钥，对 AES 密文进行解密。密钥安全管理器 DS28E15 装载有 AES 解密使用的密钥，解密后将以程序明文的方式存在于系统内存以便所有认证通过时加载和引导程序。

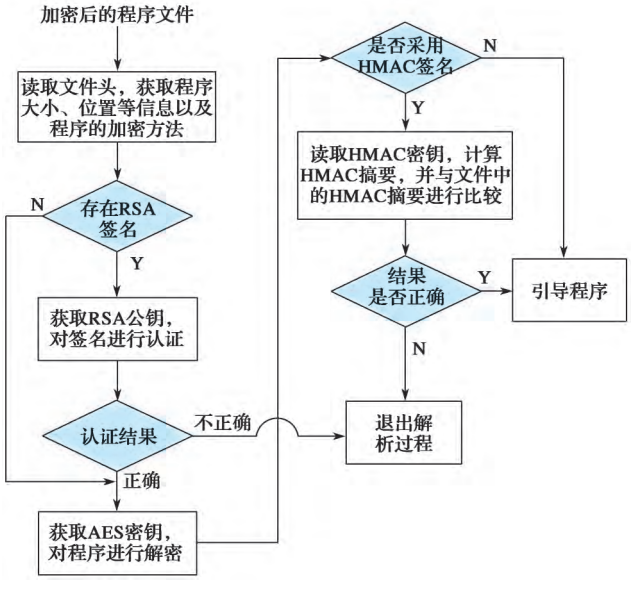

图 7　解密过程

⑤判断加密类型是否有 HMAC 消息摘要。如果是，获取 HMAC 密钥，利用 HMAC 算法对解密后的明文进行认证。HMAC 消息摘要认证能够确保程序的完整性。

⑥认证通过则执行和引导程序，认证不通过则退出执行。以上算法确保了器件身份合法性，程序完整性、加密性及抗否认性。如果其中一项未通过，则退出执行。

4　结语

在未来将会有更多的嵌入式设备出现，产品设计的知识产权保护与不受侵犯的需求也日益受到关注，良好的系统构架以及设计方法是非常有必要的。本文阐述了嵌入式系统设计者所面临的安全问题，并根据嵌入式系统特性，提出了完整的系统安全保密解决方案，对嵌入式系统设计有一定的参考价值。

参考文献：

[1] GIORDANO P. 嵌入式系统“多大程度的安全才算安全”[J]. 电子产品世界，2009，16(8)：52–54.

[2] 王沁 . FPGA 设计安全性综述 [J]. 小型微型计算机系统，2010，7(7)：1333–1337.

[3] MCEVOY R, TUNSTALL M, MURPHY C, et al. Differential power analysis of HMAC based on SHA-2, and countermeasures [C]// Information Security Applications, International Workshop, Wisa 2007. Jeju Island, Korea: DBLP, 2007:317–332.

[4] ZHOU X, TANG X. Research and implementation of RSA algorithm for encryption and decryption [C] // Proceedings of 2011 6th international forum on strategic technology. Harbin, Heilongjiang:IEEE, 2011: 1118–1121.

[5] DAEMEN J, RIJMEN V. The design of Rijndael:AES - the advanced encryption standard [M]. Berlin:Springer-Verlag Berlin and Heidelberg GmbH & Co, 2013.

[6] SOMANI U, LAKHANI K, MUNDRA M. Implementing digital signature with RSA encryption algorithm to enhance the Data security of cloud in cloud computing [C] //Parallel distributed and grid computing(PDGC), 2010 1st International conference on parallel distributed and grid computing. Solan, India:IEEE, 2010：211–216.

[7] 韩煜 . 嵌入式系统安全的密码算法及实现技术研究 [D]. 武汉：华中科技大学，2008.

[8] CHRISTOPHE, TREMLET. 工业系统需要使用安全 IC 增强保护 [J]. 中国电子商情：基础电子，2013(3)：22–26.

[9] Maxim Intergrated, MAX32590-DeepCover Secure Microcontroller with ARM926EJ-S Processor Core [EB/OL]. (2016-10) [2017–08–22]. https://datasheets.maximintegrated.com/en/ds/MAX32590.pdf.

[10] Maxim Intergrated, DE28E15-DeepCover Secure Authenticator with 1-Wire SHA-256 and 512-Bit User EEPROM [EB/OL]. (2015-03) [2017–08–22]. https://datasheets.maximintegrated.com/en/ds/DS28E15.pdf.

作者简介：戴计生（1983–），男，高级工程师，从事嵌入式控制软件和 PHM 技术的研发工作。