第 32 卷 第 18 期 计 算 机 工 程 2006 年 9 月
Vol.32 № 18 Computer Engineering September 2006
·安全技术· 文章编号：1000—3428(2006)18—0181—03 文献标识码：A 中图分类号：TP309.07

基于 FPGA 的军用龙芯计算机硬件加密方法

王 勋，毕笃彦
(空军工程大学工程学院航空电子工程系，西安 710038)

摘 要：提出了一种实用的基于 SRAM 编程的 FPGA 加密新方法，它不必选择专用的 FPGA 加密产品，可以降低成本并提供给用户较大的选型空间。通过对不同加密算法的比较，该方法实现了适用于不同需求的加密方案，以便灵活选择，并在某军用龙芯计算机的设计中得到了实际应用。

关键词：军用龙芯计算机；FPGA；加密

Hardware Encryption Method Based on FPGA Applied in Military Godson Computer

WANG Xun, BI Duyan
(Department Aeronautics Electronic Engineering, School of Engineering, Air Force Engineering University, Xi'an 710038)

【Abstract】 Design of hardware encryption of military Godson computer is proposed. This novel method based on SRAM-based FPGA can be easily implemented without specific FPGA products. It can greatly enhance the system security and be convenient for the designer to choose FPGA products without constraints. Different encryption algorithms that can be used in this method are discussed for engineering selection purpose. And this method is finally applied in the design of military PC104 board based on Godson II.

【Key words】 Military GodsonII; FPGA; Encryption

1 概述

某军用龙芯计算机采用通用龙芯 2 号 CPU，第三方北桥和南桥芯片以及扩展外设芯片，在 PC104 工业总线标准基础上，完成了系统板设计，为军用龙芯计算机开发提供了一个硬件支撑环境。在军用龙芯计算机上可以运行通用操作系统和相关的测试程序，并对常用的外部设备进行访问，从而对龙芯 2 号 CPU 设计的正确性可靠性进行验证，并为龙芯 2 号 CPU 进一步运用于军事装备提供相应的技术准备。现阶段军用龙芯计算机主要用于验证龙芯 2 号(Godson II)CPU 的性能和龙芯 IP 核的正确性。在总体设计的时候为了保证军用龙芯计算机设计的正确性和可靠性，必须对北桥芯片作出分阶段的选择。在初期为了保证提供给龙芯 2 号 CPU 一个可靠的板级支持环境，采用了第三方提供的商品化的北桥芯片来进行设计；后期为了降低成本、提高效率和可靠性，将采用我们自行设计的北桥芯片。

在初期设计中，龙芯 2 号 CPU 和第三方北桥芯片之间的复位时序、中断路由、外设访问等时序匹配需要设计接口电路，实际设计中由 FPGA 芯片来实现。在后期设计中的研发或小批量阶段，自行设计的北桥芯片考虑使用 FPGA 芯片来进行设计的验证或在实际应用中取代 ASIC 芯片。

随着半导体工艺的发展和结构的改进，可编程 FPGA 芯片相对于 ASIC 设计，在设计周期和性价比上有很大的优势。因此，越来越多的系统级设计中已经考虑采用经济实用的 FPGA 芯片来作为系统的核心。基于 SRAM 的 FPGA 芯片在容量和速度上比基于 FLASH 和反熔丝工艺的 FPGA 芯片有很大优势，因而成为应用主流。FPGA 芯片应用的推广也促使人们对于现在广泛使用的基于 SRAM 编程的 FPGA 的安全性越来越关心。由于在军用龙芯计算机板级设计中使用了 FPGA 芯片，因此在设计初期对基于 FPGA 系统的安全性研究也是设计的重点。

2 FPGA 攻击原理分析

基于FPGA的系统安全包含整个应用系统的安全、FPGA处理数据的安全、也包括对FPGA内部的设计电路和IP核的安全[1]。本文主要考虑第 3 个方面，怎样保护FPGA内部的设计和IP核，从而达到保护产品设计和知识产权的目的。

FPGA的常用攻击方法有克隆与复制、反工程、黑盒攻击等[1]。

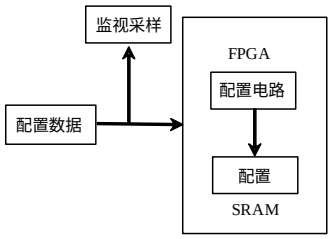


图1 FPGA芯片克隆示意图

芯片克隆是指芯片内容的原样复制，攻击者关心的是最终产品的复制。基于SRAM编程的FPGA在系统加电时，需由配置电路将存放在外部串行EPROM中的配置数据按照一定的配置时序写入到FPGA中。设计是最容易被复制的[2]。如图 1 所示，可以对FPGA的配置数据引脚进行监视来获取配置数

作者简介：王 勋(1979—)，男，博士生，主研方向：信号处理；毕笃彦，教授、博导

收稿日期：2005-10-13 **E-mail：**wangxunmail@163.com

据。利用获取的数据对其他FPGA进行配置，来对整个FPGA内部的设计进行克隆，从而达到复制产品的目的。这种方法比较简单且较容易实现。

反工程是指利用专业的工具对芯片的各金属连接层依次进行照相和剥离以得到芯片结构，从而获取设计的网表和电路图来制作和复制成为新的电路。这种攻击方法需要专业的人力和物力来实现，需要较高的成本和时间。

黑盒攻击指通过对芯片输入所有可能的输入值，获取对应的输出值，得到系统设计的真值表来获取设计。这种方法对小型设计来说比较有效，可以成功地得到设计逻辑，但对于复杂的设计来说，穷举所有的设计输入需要大量的时间甚至不可能，但需要防止局部或核心逻辑被获取。

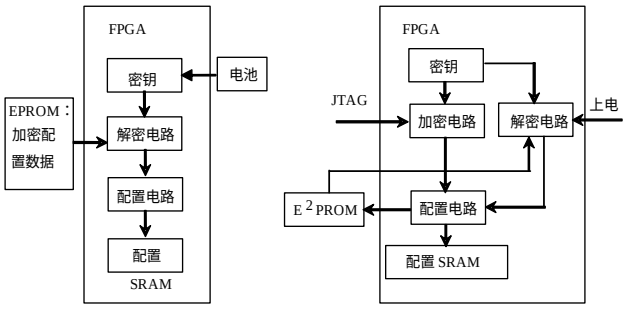
在军用龙芯计算机设计中，由于 FPGA 的设计有一定的复杂性，采用黑盒攻击来获取设计基本不可能，利用反工程来获取设计需要攻击者较高的技能和一定的时间。因此主要考虑对易被获取的配置数据进行保护，达到保护设计的目的。

3 FPGA 加密保护

业界和 FPGA 器件生产厂家也意识到 FPGA 安全的重要性，针对配置数据的保护，推出了一系列的解决方案，其主要思路是针对配置数据进行不可逆加密，在 FPGA 内置加密或解密电路。

如图 2(a)所示，其中一种方案考虑采用将配置数据加密后写入到 EPROM 中，密钥预先写入 FPGA 中特定区域，由外部电池供电保持。上电后由解密电路读入加密配置数据，根据密钥解密后再进行配制。这种方法外部攻击者只能得到加密后的配置数据，只要无法得到密钥，就不能正确复制电路。但它显著的缺点就在于需要用户管理密钥，借助外部电池来记忆密钥容易造成密钥丢失和占用一定的 PCB 空间。

一种针对上述方法的改进方案如图 2(b)所示[2]，每片FPGA芯片在出厂时内部固化一个唯一的密钥，使用时通过JTAG将配置数据传递给加密电路，加密电路根据芯片的密钥将配置数据加密后由内部配置电路写入到外部EPROM或FLASH ROM中。芯片在实际使用中，上电时由内部解密电路进行解密，对SRAM进行配制后，芯片正常工作。这种方案的优点是无需用户管理密钥，使用方便。



(a)内置解密电路的 FPGA 加密示意图　　(b)内置解密、解密电路的 FPGA 加密示意图

图 2 FPGA 加密保护

上述方案由 FPGA 厂商提供，使用时具有一定的便利性，但现阶段只在一些特定型号的 FPGA 中提供，导致用户选型的局限性。同时内部集成了加解密电路，成本过高。因此在实际应用中，找到一种使用简单、并能应用于现阶段广泛使用的基于 SRAM 编程的普通 FPGA 芯片加密方法十分必要。

4 军用龙芯计算机系统板的 FPGA 加密方法

在主板的设计过程中，作者分析了不同种类设计芯片的安全等级，在此基础上提出了一种实用的基于 SRAM 编程的 FPGA 加密新方法，用于系统板的设计。并对使用不同的加密算法的结果进行了对比分析，以便应用于不同的场合。

如表 1 所示，Actel公司给出了不同种类的设计芯片的安全等级[3]，其中 5 代表安全级别最高，1 代表安全级别最低。

表 1 设计芯片安全等级表

设计芯片	安全等级
普通 SRAM 编程 FPGA	1
基于 EEPROM 的 PLD 和 CPLD	2
ASIC	3
内置加解密电路的 SRAM FPGA	4
基于 FLASH 编程的 FPGA	5
反熔丝(anti-fuse) FPGA	5

由表 1 可知，基于 SRAM 编程的 FPGA 安全性最低，为安全 1 级，代表除了配置数据存在安全性问题外，对其进行专业的反工程可能只需几个小时；而对于安全性最高的 FLASH 或反熔丝 FPGA 芯片来说，首先不存在加电配置的问题，而且进行反工程可能需要几年的时间。

因此，为了保证设计的安全性，可考虑采用借助 FLASH FPGA 的安全性来提高整个设计的安全性。由于 SRAM FPGA 在速度和容量上有优势，可将系统核心的、高速的逻辑在 SRAM FPGA 中实现，而部分低速逻辑可以在 FLASH FPGA 中实现。通过合理分配和芯片选型可以达到充分利用芯片资源的目的。由于攻击者不可能得到设计的全部内容，因此无法进行系统复制，但可能获取部分关键的逻辑。

为了进一步加强系统的安全性，针对 SRAM FPGA 中设计的安全，本文在 SRAM FPGA 和 FLASH FPGA 之间设计握手电路，只有握手通过，才启动 SRAM FPGA 内部逻辑的实现，这样攻击者即使获取 SRAM FPGA 的配置数据，内部电路也不会工作，达到保护 SRAM FPGA 内部设计的目的。具体设计如图 3 所示。

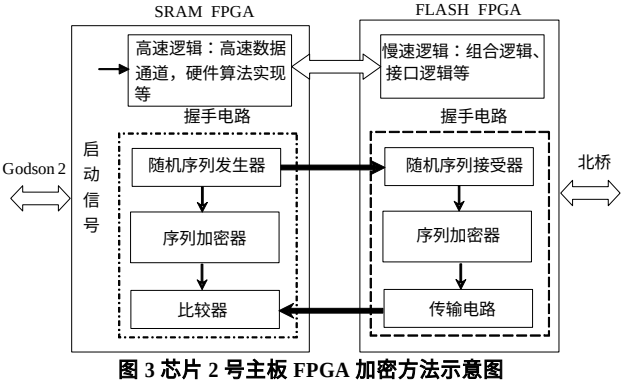


图 3 芯片 2 号主板 FPGA 加密方法示意图

图 3 所示的方案中，将系统逻辑分为高速逻辑和低速逻辑，分别配置在基于 SRAM 和 FLASH 的 FPGA 中。FLASH FPGA 的高安全性保证了整个系统逻辑的安全性，为了进一步保证 SRAM FPGA 的安全性，在二者之间设置了握手电路。

握手电路通过主方(SRAM FPGA)的随机系列发生器，上电时产生随机数序列传送到从方(FLASH FPGA)。主从两方采用相同的的序列加密器，从方将加密后的序列传送到主方，通过比较器比较一致后，产生启动信号，启动高速逻辑部分。各部分具体设计如下：

(1)随机系列发生器采用m 级线性反馈移位寄存器(LFSR)，m是产生随机数序列的位数，LFSR的优势在于随着m 的增大，LFSR对FPGA 内部逻辑资源的消耗量是线性增加的，而LFSR 输出的最长序列周期却呈指数增长，这一点对实际应用来说是非常有利的。最长序列周期(2^m-1)决定了攻击者获取完备序列的难度。实际应用中m

可取 32，64 甚至 128。这取决于用户在加密深度和FPGA 内部逻辑资源消耗量的权衡上。实际上当m=32 时，最大长度序列的周期可达 $2.147\ 48\times10^9$。

(2)主从两方采用的相同的序列加密器，可根据实际应用选择。第 1 种选择可以考虑采用加密算法公开，借助私钥来进行加密的算法，如 3DES、AES 等。这些算法都经过实践的检验，破解难度极大。用户可根据自己的需要选择不同级别的加密算法，在文献[4]中对这些算法的原理和实现都有很详尽的介绍。缺点就是由于实现相对复杂，难免占用一定的硬件资源。因此另一种选择可以针对现有算法进行简化，由于攻击者不知道改造后的加密算法，从而导致解密的难度增大，起到保护的作用。设计时参考经典的 RC5 算法(由 Rivest 设计的一种序列密码算法)进行简化，只用几种最简单的运算(XOR，ADD，SUB 等)。但具有很好的统计特性，在加密多轮后，可以有效抗击穷举攻击。经过简化后的算法占用较少硬件资源，但可获得一定程度的安全性能，工程实现也相对便利。

本文对以上算法进行了设计实现，设计平台为 Synplifer / Xilinx Foundation，目标器件为 Virtex 系列，一个 Virtex slice 包括两个查找表(LUTs)。表 2 给出了各种实现算法占用 FPGA 资源的对比，实际应用中可在安全性能要求与硬件资源需求之间进行权衡。表 2 中的数据由于在设计实现时可能采用不同的结构或优化方法而不同。

表 2 加密算法占用资源比较

设计 算法	设计说明	占用资源 (CLB SLICES)
标准 RC5	Word length=32 Rounds=12	3 223
3DES	16 rounds	3 977
AES	Rijndael	4 309
简化 RC5	Word length=16 Rounds=12	778

(3)随机序列接收和传输电路采用串行方式进行收发，可减少占用 FPGA 引脚资源。

在军用龙芯计算机的设计过程中，考虑到 FPGA 资源的充分利用，本文采用了针对龙芯 2 号主板特点的序列加密算法，该加密算法易于工程实现，具有较高的安全性。在安全等级要求更高的实际应用中，可采用加密性能更好的 3DES，AES 等算法。

5 小结

本文提出了一种军用龙芯计算机设计中易于工程应用的 FPGA 加密方法，它避免了选择专用的加密 FPGA 产品，可以减少成本并提供给用户较大的选型空间。同时给出了方法中适用于不同需求的加密算法选择方案，便于灵活选择。

随着龙芯系列 CPU 在军事上的应用推广，具有自主知识产权的 CPU 将在国防、航空等关键领域发挥重大的作用，军用龙芯系列计算机的 FPGA 加密技术将有更大应用价值。

参考文献

1 Bossuet L, Gogniat G, Burleson W. Dynamically Configurable Security for SRAM FPGA Bitstreams[C]. Proceedings of Parallel and Distributed Processing Symposium, 2004.

2 Kean T. Secure Configuration of a Field Programmable Gate Array[C]. Proceedings of IEEE Symposium on Field Programmable Custom Computing Machines, Rohnert Park, CA , 2001.

3 Coporation A. Security Resource Center[Z]. http://www.actel.com, 2005.

4 Weaver N. A Comparison of the AES Candidates Amenability to FPGA Implementation[C]. Proceedings of the 3rd Advanced Encryption Standard Candidate Conference, 2000: 12-14.

(上接第 180 页)

ACSPI 建立在 CSPI 的基础之上，并对 CSPI 提供的函数进行了简化。用户只需要提供数据，就可以通过 ACSPI 完成加密、解密、签名、验证等工作。ACSPI 对 CSPI 提供的函数作了如下简化：

(1)对称加密算法指定为 CBC(加密块链接)模式的 IDEA 算法。

(2)非对称加密算法指定为密钥长度为 1 024 比特的 RSA 算法。

(3)用于签名和校验的消息摘要算法指定为 MD5。

ACSPI 函数库的包含如下两个模块：

(1)用户接口模块：将用户的请求转化为 CSSM 体系内部的消息，将从请求/响应队列返回的消息转化为对用户的响应。

(2)队列接口模块：提供对请求/响应队列的消息的收发功能。

其中的队列接口模块与 CSPI 中的队列接口模块相同。用户接口模块分为加密/解密模块、签名/校验模块和密钥管理模块等 3 个功能模块。

(1)加密/解密模块：对输入的数据用 IDEA 算法加密/解密；对指定文件中的数据用 IDEA 算法加密/解密，并输出到指定输出文件中。

(2)签名/校验模块：对输入数据用 RSA 算法进行签名；对输入数据及签名用 RSA 算法进行校验；对指定文件用 RSA 算法签名，并把签名输出到指定文件；对指定文件及签名文件用 RSA 算法进行校验。

(3)密钥管理模块：实现 IDEA 密钥和 RSA 公钥、私钥的生成和管理。其主要功能有：生成对称密钥/非对称密钥句柄；将对称密钥/非对称密钥保存到指定的文件；从指定的文件读取对称密钥/非对称密钥。

4 小结

CSPI 是和 CSSM 紧密相关的软件模块，直接建立在 CSSM 的服务上。用户可以通过 CSPI 访问 CSSM 提供的服务，完成安全中间件的基本功能。

CSSM 通过向上提供统一的编程接口 CSPI，实现底层复杂的安全系统对应用软件的透明化，应用软件只需使用安全中间件提供的接口，即可方便地使用各种网络信息安全服务，从而缩短了应用软件的开发时间，降低了软件的开发成本，提高了软件的开发效率。

参考文献

1 The Open Group. Common Security: CDSA and CSSM[Z]. http://www.opengroup.org/pubs/catalog/c914.htm.

2 Rajan A, Wood M, Bowler D. Mechanics of the Common Security Services Manager[Z]. http://www.pentium.fr/cd/ids/developer/asmona/eng/20289.htm.

3 佘　堃, 周明天. 安全中间件核心——公共安全服务[J]. 小型微型计算机系统, 2003, 24(7): 1190.

4 Krawczyk H, Bellare M, Canetti R. Keyed-Hashing for Message Authentication[S]. RFC 2104, http://www.ietf.org/rfc/rfc2104.txt.

5 Diffie W, Hellman M E. New Direction in Cryptography[J]. IEEE Transaction on Information Theory, 1976, 22(6): 644-654.

6 Rivest R L, Shamir A, Adleman L. A Method for Obtaining Digital Signature and Public Cryptosystem[J]. Communication of the ACM, 1978, 21(2): 120-126.

7 Quisguater J J, Couvrevur C. Fast Decipherment Algorithm for RSA Public-key Cryptosystem[J]. Electronics Letters, 1982, 18(21): 905-907.